

**операционная
система**

ОС ЕС

В. П. ДАНИЛОЧКИН,
В. В. МИТРОФАНОВ,
Б. В. ОДИНЦОВ,
Г. В. ПЕЛЕДОВ

Операционная система ОС ЕС

СПРАВОЧНОЕ ПОСОБИЕ

Под редакцией Л. Д. Райкова



МОСКВА «СТАТИСТИКА» 1980

ББК 32.973
О-60

О-60 **Операционная система ОС ЕС: Справочное пособие/**
В. П. Данилочкин, В. В. Митрофанов, Б. В. Одинцов,
Г. В. Пеледов; Под ред. Л. Д. Райкова.—М.: Статистика,
1980.— 480 с.

В пер.: 1 р. 80 к.

В справочное пособие вошел ключевой материал, обобщающий сведения по операционной системе ОС ЕС (издания 4.1 и 6.1) и ее применению на программно-совместимых моделях ЕС ЭВМ. Пособие содержит краткие сведения о языке Ассемблера, командах оператора, процедуре начальной загрузки, форматах языка управления заданиями и системных макрокоманд, использовании Редактора связей, Загрузчика, Сортировки-объединения, программном обеспечении машинной графики и телеобработки данных, кодах аварийного завершения и ожидания.

Для прикладных программистов и операторов ЭВМ.

О 30502—133
008(01)—80 90—80 2405000000

ББК 32.973
6Ф7.3

Предисловие

Операционная система ОС ЕС в настоящее время получила значительное распространение в нашей стране и за рубежом на установках ЕС ЭВМ. Это привело к расширению круга пользователей этой системы. В связи с этим возникла проблема обеспечения пользователей операционной системы ОС ЕС справочными материалами в качестве вспомогательного средства в повседневной практической деятельности. Необходимость в издании справочных материалов объясняется также тем фактом, что операционная система ОС ЕС имеет много разнообразных средств и возможностей, которые требуют помощи для быстрой ориентации в них.

Настоящее справочное пособие содержит информацию, необходимую для эффективного использования операционной системы ОС ЕС изданий 4.1 и 6.1. Эти издания сейчас наиболее распространенные. Большая часть сведений является общей для обоих указанных изданий ОС ЕС и не содержит упоминаний об издании. Те сведения, которые относятся только к одному из изданий, содержат специальные указания об этом.

Пособие адресовано в основном программистам и операторам, использующим операционную систему ОС ЕС ЭВМ, однако оно может быть полезным также системным программистам, управленческому и инженерному персоналу вычислительных центров. Материал, включенный в справочное пособие, охватывает управляющую программу операционной системы ОС ЕС (включая компоненты, обеспечивающие использование средств телеобработки данных и машинной графики), системные обрабатывающие программы и компоненты системы программирования ОС ЕС: Ассемблер, Редактор связей и Загрузчик. Главы 2, 3 и 17 предназначены в основном для операторов операционной системы ОС ЕС, главы 12 и 16 могут быть использованы как операторами, так и программистами. Остальные главы предназначены в основном для программистов.

Основным критерием подготовки данного пособия была полнота охвата материала в сочетании с краткостью и наглядностью изложения. Справочный материал сгруппирован по главам. В связи с этим при расположении внутри глав описаний команд опера-

тора, макрокоманд, управляющих операторов алфавитный порядок не соблюдается и для поиска информации необходимо пользоваться оглавлением. Такая структура справочного пособия позволяет, во-первых, использовать его как справочник для быстрого и частого поиска необходимой информации при использовании ОС ЕС и, во-вторых, для ознакомления с материалом по определенным темам при изучении ОС ЕС. В связи с этим пособие будет полезным не только для пользователей ОС ЕС, но и для студентов, аспирантов и лиц, изучающих операционную систему.

В описании форматов команд оператора, управляющих операторов и макрокоманд приняты условные обозначения, приведенные в конце книги.

Ассемблер

1.1. Основные элементы языка Ассемблера

1.1.1. Операторы

Программа на языке Ассемблера есть последовательность операторов, которые представляют собой команды и комментарии. Команды делятся на три группы: машинные команды, команды Ассемблера и макрокоманды.

Для записи операторов языка Ассемблера используется набор символов:

прописные буквы латинского алфавита: A B C D E F G H I J K L M N O P Q R S T U V W X Y Z;

цифры: 0 1 2 3 4 5 6 7 8 9;

специальные символы: + - = * . () / ' &, пробел \square # @ (последние три символа рассматриваются Ассемблером как буквы).

В комментариях, а также между апострофами в операндах могут использоваться все символы двоичного кода обмена информацией (ДКОИ).

Операторы Ассемблера записываются на стандартных бланках. Позиции на бланке соответствуют позициям на перфокарте или позициям в исходном операторе, вводимом, например, с клавиатуры абонентского пункта. Каждая строка бланка делится на три части: поле оператора (позиции 1—71), позиция указателя продолжения (позиция 72) и поле идентификации последовательности (позиции 73—80). Продолжение оператора в следующей строке начинается с позиции 16. Указателем продолжения является символ, отличный от пробела в позиции 72. Для записи одного оператора Ассемблера можно использовать только две строки продолжения. Исключение составляют макрокоманды и оператор прототипа в макроопределении, для которых допускается необходимое число строк продолжения. От рассмотренного стандартного формата записи оператора с помощью команды Ассемблера `ICTL` можно перейти к произвольному формату записи операторов, т. е. изменить стандартные позиции начала и конца оператора, позиции продолжения.

Любой оператор Ассемблера, за исключением операторов комментариев, может включать следующие поля: поле названия, поле операции, поле операндов, поле комментариев, которые отделяются друг от друга пробелами.

В стандартной форме записи оператора каждое поле записывается в определенной графе бланка. Поле названия начинается с позиции 1, поле операции — с позиции 10, поле операндов — с позиции 16.

Поле названия может содержать символическое имя, идентифицирующее оператор, или может быть пустым. Поле названия не должно содержать более восьми элементарных символов.

Поле операции предназначено для записи мнемонического кода операции, который определяет функцию оператора. Допустимые коды операций могут содержать не более пяти символов для кодов операций машинных команд или команд Ассемблера и не более восьми символов для кодов операций макрокоманд.

Поле операндов предназначено для записи операндов, идентифицирующих и описывающих данные, с которыми будут производиться действия по данной команде. Операнды друг от друга отделяются запятой.

Поле комментариев используется для пояснения и описания оператора. Заполнять поле комментариев не обязательно. В операторах, в которых запись операндов не обязательна, но желательна запись комментариев, отсутствие операндов должно быть указано запятой, перед которой и после которой должен быть по крайней мере один пробел. В операторах прототипа макрокоманд и в макрокомандах без операндов комментарии не допускаются.

Операторы комментариев дают возможность программисту записывать пояснения к программе, используя все поле оператора, а не только поле комментариев. Операторы комментариев записываются по следующим правилам:

в позиции начала должен быть записан символ *;

в макроопределениях в позиции начала записывается точка и за ней символ *;

комментарии должны находиться в поле оператора (допускается строка продолжения);

операторы комментариев не должны включаться между несколькими строками одного оператора.

Поле идентификации в строке стандартного бланка используется для идентификации программы и (или) нумерации операторов.

1.1.2. Термы

Операнды операторов языка Ассемблера строятся из выражений, а выражения — из термов. Терм представляет собой некоторое значение. Допускается пять типов термов: имя, самоопределенный терм, значение счетчика адреса, характеристика длины имени, литерал. Значение может присваиваться Ассемблером (имя,

характеристика длины имени, значения счетчика адреса) или может быть определено самим термом (самоопределенный терм, литерал). Значение термина не должно превышать $2^{24}-1$.

Термы классифицируются на абсолютные и перемещаемые. Терм является абсолютным, если его значение не изменяется при перемещении программы. Терм является перемещаемым, если его значение изменяется при перемещении программы.

Имя — это символ или комбинация символов, используемые для обозначения адресов или произвольных величин. Предусмотрены три типа имен: обычные имена, символы переменных, метки следования. Обычные имена записываются в виде последовательности букв и цифр. Имя не должно состоять более чем из восьми символов и должно начинаться с буквы.

Символы переменных должны начинаться специальным символом **&**, за которым следуют от одной до семи букв и (или) цифр, первой из которых должна быть буква.

Метки следования состоят из точки, за которой следуют от одной до семи букв и (или) цифр, первой из которых должна быть буква.

Метки следования и символы переменных применяются в макросредствах Ассемблера.

Считается, что имя определено, если оно появляется в поле названия какого-либо оператора. Возможно определение имени с помощью команды Ассемблера **EQU**. Определение имени означает присвоение этому имени значения, а также присвоение признака перемещаемости и характеристики длины. Имена, называющие области памяти, являются перемещаемыми, остальные имена — абсолютными. Характеристика длины имени представляет собой длину в байтах той области памяти, адрес которой представлен этим именем.

Самоопределенный терм — это терм, значение которого выражено в нем самом. Существуют четыре типа самоопределенных термов: десятичный, шестнадцатеричный, двоичный и символьный. Самоопределенные термы являются абсолютными.

Десятичный самоопределенный терм — это десятичное целое число без знака, записанное как последовательность десятичных цифр. Десятичный терм не может состоять больше чем из десяти цифр или превышать значение $2^{31}-1$. Шестнадцатеричный самоопределенный терм — это шестнадцатеричное число без знака, записанное в виде последовательности шестнадцатеричных цифр (от одной до восьми). Цифры должны быть заключены в апострофы, первому апострофу должна предшествовать буква **X**. Двоичный самоопределенный терм — это последовательность единиц и нулей без знака, заключенная в апострофы. Первому апострофу должна предшествовать буква **B**. Двоичный терм может иметь до 32 двоичных цифр. Символьный самоопределенный терм — это последовательность от одного до четырех элементарных символов, заключенная в апострофы. Первому апострофу должна предшествовать буква **C**. В символьном терме может быть использован любой

символ кода ДКОИ. Для каждого апострофа или символа &, которые нужно использовать в символьном самоопределенном терме, необходимо записать два апострофа или два символа &.

Счетчик адреса используется для присвоения адресов операторам программы. Ассемблер формирует счетчик адреса для каждой программной секции. На значение счетчика адреса можно повлиять командами Ассемблера START, ORG. Воздействию какой-либо из этих команд всегда подвергается счетчик адреса той программной секции, в которой эти команды появились. Максимальное значение, которое может принимать счетчик адреса, равно $2^{24}-1$. Текущее значение счетчика адреса может использоваться в операторах исходной программы в качестве терма. Для этого необходимо указать в поле операндов оператора звездочку *. Значение счетчика адреса является перемещаемым термом.

Характеристика длины имени может использоваться как абсолютный терм. Для этого необходимо записать L' и далее имя.

Литерал — это константа, записанная в машинной команде на месте адреса. Этой константе предшествует знак =.

Литералы дают возможность вводить в программу константы (например, числа для вычислений, адреса, слова или текст для печати сообщений) путем определения константы прямо в поле операндов машинной команды, в которой она используется. Литералы следует отличать от непосредственных данных в машинных командах. Непосредственные данные содержатся в команде, тогда как вместо литералов в команду транслируются их адреса. Метод описания и определения константы как литерала почти идентичен методу определения константы в операнде команды Ассемблера DC.

1.1.3. Выражения

Выражения состояются из отдельных термов или из арифметической комбинации термов. Выражение является абсолютным или перемещаемым в зависимости от соответствующих свойств терма (или термов, его составляющих). В выражении допускаются одноместные (оперирующие с одним значением) и двуместные (оперирующие с двумя значениями) операции. Знаки одноместных операций: + (положительный), — (отрицательный). Знаки двуместных операций: + (сложение), — (вычитание), × (умножение), / (деление). В начале выражения или перед любым термом выражения может находиться один или несколько знаков одноместных операций.

Самый левый знак операции среди нескольких знаков операций между двумя термами считается знаком двуместной операции, остальные должны быть знаками одноместных операций. Выражение не должно начинаться со знака двуместной операции × или /. Выражение не должно содержать два терма или два знака двуместных операций подряд. Выражение может содержать не более 19 знаков одноместных и двуместных операций и не более 20

термов. В выражении допускается не более шести уровней скобок. Выражение, состоящее из нескольких термов, не может содержать литерал. В операциях умножения и деления не должны быть компонентами перемещаемые термы. Значение выражения должно находиться в пределах от минус 2^{31} до плюс $2^{31}-1$.

1.2. Машинные команды

В табл. 1.1 — 1.10 приведен состав машинных команд ЕС ЭВМ с распределением команд по функциональным группам.

В таблицах используются следующие обозначения:

RR — формат команды RR; S — формат команды S;

RS — формат команды RS; SI — формат команды SI;

RX — формат команды RX; SS — формат команды SS.

R1, R2, R3 — абсолютные выражения, определяющие номера общих регистров 0—15 или номера регистров с плавающей точкой 0, 2, 4 и 6 соответственно в операндах 1, 2, 3;

X2 — абсолютное выражение, определяющее номер одного из общих регистров 0—15, используемого в качестве регистра индекса во втором операнде;

B1 и B2 — абсолютные выражения, определяющие номера регистров базы 0—15 для операндов 1 и 2 соответственно;

D1 и D2 — абсолютные выражения, определяющие смещения для операндов 1 и 2 соответственно (значение выражения должно находиться в пределах от 0 до 4095);

L, L1, L2 — абсолютные выражения, определяющие длины операндов 1 и 2 (значение L должно быть в пределах от 1 до 256, значения L1 и L2 должны быть в пределах от 1 до 16);

I, I2, I3 — абсолютное выражение, определяющее непосредственные данные машинной команды, являющиеся операндом 1, 2 или 3 (значения выражений I, I2 должны быть в пределах от 0 до 255, выражения I3 — в пределах от 0 до 9);

M1, M3 — абсолютные выражения, определяющие 4-разрядную маску машинной команды (значения выражений должны быть в пределах от 0 до 15);

ПР — устанавливается признак результата (ПР1 — для команды LPSW признак результата определяется содержимым соответствующего поля нового PSW; ПР2 — для команды SPM признак результата устанавливается в соответствии со значением второго и третьего бит регистра, заданного полем R1);

ПК — привилегированная команда;

P2 — машинная команда входит в состав команд ЭВМ, работающих по принципам работы Ряд-2 (ЕС ЭВМ-2)¹.

¹ Некоторые отличия имеются в составе команд ЭВМ ЕС-1035.

Таблица 1.1

Стандартный набор команд

Название	Мнемоника	Код	Формат	Операнды	Примечания		
1	2	3	4	5	6	7	8
Сложение	AR	1A	RR	R1,R2	ПР		
Сложение	A	5A	RX	R1,D2(X2,B2)	ПР		
Сложение полуслова	AH	4A	RX	R1,D2(X2,B2)	ПР		
Сложение кодов	ALR	1E	RR	R1,R2	ПР		
Сложение кодов	AL	5E	RX	R1,D2(X2,B2)	ПР		
И	NR	14	RR	R1,R2	ПР		
И	N	54	RX	R1,D2(X2,B2)	ПР		
И	NC	D4	SS	D1(L,B1),D2(B2)	ПР		
И непосредственное	NI	94	SI	D1(B1),I2	ПР		
Переход с возвратом	BALR	05	RR	R1,R2			
Переход с возвратом	BAL	45	RX	R1,D2(X2,B2)			
Условный переход	BCR	07	RR	M1,R2			
Условный переход	BC	47	RX	M1,D2(X2,B2)			
Переход по счетчику	BCTR	06	RR	R1,R2			
Переход по счетчику	BCT	46	RX	R1,D2(X2,B2)			
Переход по индексу больше	BXH	86	RS	R1,R3,D2(B2)			
Переход по индексу меньше или равно	BXLE	87	RS	R1,R3,D2(B2)			
Освободить ввод-вывод	CLRIO	9D01	S	D2(B2)	ПР	ПК	P2
Сравнение	CR	19	RR	R1,R2	ПР		
Сравнение	C	59	RX	R1,D2(X2,B2)	ПР		
Сравнение полуслова	CH	49	RX	R1,D2(X2,B2)	ПР		
Сравнение кодов	CLR	15	RR	R1,R2	ПР		
Сравнение кодов	CL	15	RX	R1,D2(X2,B2)	ПР		
Сравнение кодов	CLC	D5	SS	D1(L,B1),D2(B2)	ПР		
Сравнение кодов	CLI	95	SI	D1(B1),I2	ПР		
Сравнение символов по маске	CLM	BD	RS	R1,M3,D2(B2)	ПР		
Сравнение кодов длинное	CLCL	OF	RR	R1,R2	ПР		P2
Преобразование в двоичную	CVB	4F	RX	R1,D2(X2,B2)			
Преобразование в десятичную	CVD	4E	RX	R1,D2(X2,B2)			
Диагностика		83				ПК	
Деление	DR	1D	RR	R1,R2			
Деление	D	5D	RX	R1,D2(X2,B2)			
Исключающее ИЛИ	XR	17	RR	R1,R2	ПР		
Исключающее ИЛИ	X	57	RX	R1,D2(X2,B2)	ПР		
Исключающее ИЛИ	XC	D7	SS	D1(L,B1),D2(B2)	ПР		
Исключающее ИЛИ непосредственное	XI	97	SI	D1(B1),I2	ПР		
Выполнить	EX	44	RX	R1,D2(X2,B2)			
Остановить устройство	HDV	9E01	S	D2(B2)	ПР	ПК	P2
Остановить ввод-вывод	HI0	9E00	S	D2(B2)	ПР	ПК	
Прочитать символ	IC	43	RX	R1,D2(X2,B2)			
Прочитать символы по маске	ICM	BF	RS	R1,M3,D2(B2)	ПР		P2

Название	Мнемо-ника	Код	Формат	Операнды	Примечания		
					6	7	8
1	2	3	4	5	6	7	8
Прочитать ключ памяти	ISK	09	RR	R1,R2		ПК	
Загрузка	LR	18	RR	R1,R2			
Загрузка	L	58	RX	R1,D2(X2,B2)			
Загрузка адреса	LA	41	RX	R1,D2(X2,B2)			
Загрузка и проверка	LTR	12	RR	R1,R2	ПР		
Загрузка дополнения	LCR	13	RR	R1,R2	ПР		
Загрузка управляющих регистров	LCTL	B7	RS	R1,R3,D2(B2)		ПК	P2
Загрузка полуслова	LH	48	RX	R1,D2(X2,B2)			
Загрузка групповая	LM	98	RS	R1,R3,D2(B2)			
Загрузка отрицательная	LNR	11	RR	R1,R2	ПР		
Загрузка положительная	LPR	10	RR	R1,R2	ПР		
Загрузка PSW	LPSW	82	S	D2(B2)	ПР1	ПК	
Обращение к монитору	MC	AF	SI	D1(B1),I2			P2
Пересылка	MVC	D2	SS	D1(L,B1),D2(B2)			
Пересылка непосредственная	MVI	92	SI	D1(B1),I2			
Пересылка длинная	MVCL	0E	RR	R1,R2	ПР		P2
Пересылка цифр	MVN	D1	SS	D1(L,B1),D2(B2)			
Пересылка со сдвигом	MVO	F1	SS	D1(L1,B1),D2(L2,B2)			
Пересылка зон	MVZ	D3	SS	D1(L,B1),D2(B2)			
Умножение	MR	1C	RR	R1,R2			
Умножение	M	5C	RX	R1,D2(X2,B2)			
Умножение	MH	4C	RX	R1,D2(X2,B2)			
ИЛИ	OR	16	RR	R1,R2	ПР		
ИЛИ	O	56	RX	R1,D2(X2,B2)	ПР		
ИЛИ	OC	D6	SS	D1(L,B1),D2(B2)	ПР		
ИЛИ непосредственное	OI	96	SI	D1(B1),I2	ПР		
Упаковать	PACK	F2	SS	D1(L1,B1),D2(L2,B2)			
Поставить часы	SCK	B204	S	D2(B2)	ПР	ПК	P2
Установить маску программы	SPM	04	RR	R1	ПР2		
Установить ключ памяти	SSK	08	RR	R1,R2		ПК	
Установить маску системы	SSM	80	S	D2(B2)		ПК	
Сдвиг влево двойной арифметический	SLDA	8F	RS	R1,D2(B2)	ПР		
Сдвиг влево двойной кода	SLDL	8D	RS	R1,D2(B2)			
Сдвиг влево арифметический	SLA	8B	RS	R1,D2(B2)	ПР		
Сдвиг влево кода	SLL	89	RS	R1,D2(B2)			
Сдвиг вправо двойной арифметический	SRDA	8E	RS	R1,D2(B2)	ПР		
Сдвиг вправо двойной кода	SRDL	8C	RS	R1,D2(B2)			

Название	Мнемо-ника	Код	Формат	Операнды	Примечания		
					6	7	8
1	2	3	4	5			
Сдвиг вправо арифметический	SRA	8A	RS	R1,D2(B2)	ПР		
Сдвиг вправо кода	SRL	88	RS	R1,D2(B2)			
Начать ввод-вывод	SIO	9C00	S	D2(B2)	ПР	ПК	
Начать ввод-вывод с быстрым отключением	SIOF	9C01	S	D2(B2)	ПР	ПК	P2
Запись в память	ST	50	RX	R1,D2(X2,B2)			
Запись в память идентификатора канала	STIDC	B203	S	D2(B2)	ПР	ПК	P2
Запись в память символа	STC	42	RX	R1,D2(X2,B2)			
Запись в память символов по маске	STCM	BE	RS	R1, M3,D2(B2)			P2
Запись в память показания часов	STCK	B205	S	D2(B2)	ПР		P2
Запись в память управляющих регистров	STCTL	B6	RS	R1,R3,D2(B2)		ПК	P2
Запись в память идентификатора процессора	STIDP	B202	S	D2(B2)		ПК	P2
Запись в память полуслова	STH	40	RX	R1,D2(X2,B2)			
Запись в память групповая	STM	90	RS	R1,R3,D2(B2)			
Вычитание	SR	1B	RR	R1,R2	ПР		
Вычитание	S	5B	RX	R1,D2(X2,B2)	ПР		
Вычитание полуслова	SH	4B	RX	R1,D2(X2,B2)	ПР		
Вычитание кодов	SLR	1F	RR	R1,R2	ПР		
Вычитание кодов	SL	5F	RX	R1,D2(X2,B2)	ПР		
Обращение к супервизору	SVC	0A	RR	I			
Проверить и установить	TS	93	S	D2(B2)	ПР		
Проверить канал	TCH	9F00	S	D2(B2)	ПР	ПК	
Проверить ввод-вывод	TIO	9D00	S	D2(B2)	ПР	ПК	
Проверить по маске	TM	91	SI	D1(B1),I2	ПР		
Перекодировать	TR	DC	SS	D1(L,B1),D2(B2)			
Перекодировать и проверить	TRT	DD	SS	D1(L,B1),D2(B2)	ПР		
Распаковать	UNPK	F3	SS	D1(L1,B1),D2(L2,B2)			

Таблица 1.2

Команды десятичной арифметики

Название	Мнемоника	Код	Формат	Операнды	Примечание
Сложение десятичное	AP	FA	SS	D1 (L1,B1), D2 (L2,B2)	ПР
Сравнение десятичное	CP	F9	SS	D1 (L1,B1), D2 (L2,B2)	ПР
Деление десятичное	DP	FD	SS	D1 (L1,B1), D2 (L2,B2)	
Отредактировать	ED	DE	SS	D1 (L1,B1), D2 (L2,B2)	ПР
Отредактировать и отметить	EDMK	DF	SS	D1 (L1,B1), D2 (L2,B2)	ПР
Умножение десятичное	MP	FC	SS	D1 (L1,B1), D2 (L2,B2)	
Сдвиг с округлением десятичный	SRP	F0	SS	D1 (L1,B1), D2 (L2,B2), I3	ПР P2
Вычитание десятичное	SP	FB	SS	D1 (L1,B1), D2 (L2,B2)	ПР
Сложение с очисткой	ZAP	F8	SS	D1 (L1,B1), D2 (L2,B2)	ПР

Таблица 1.3

Команды арифметики с плавающей точкой

Название	Мнемоника	Код	Формат	Операнды	Примечания
Сложение с нормализацией (длинные операнды)	ADR	2A	RR	R1,R2	ПР
Сложение с нормализацией (длинные операнды)	AD	6A	RX	R1,D2(X2,B2)	ПР
Сложение с нормализацией (короткие операнды)	AER	3A	RR	R1,R2	ПР
Сложение с нормализацией (короткие операнды)	AE	7A	RX	R1,D2(X2,B2)	ПР
Сложение без нормализации (длинные операнды)	AWR	2E	RR	R1,R2	ПР
Сложение без нормализации (длинные операнды)	AW	6E	RX	R1,D2(X2,B2)	ПР
Сложение без нормализации (короткие операнды)	AUR	3E	RR	R1,R2	ПР
Сложение без нормализации (короткие операнды)	AU	7E	RX	R1,D2(X2,B2)	ПР
Сравнение (длинные операнды)	CDR	29	RR	R1,R2	ПР
Сравнение (длинные операнды)	CD	69	RX	R1,D2(X2,B2)	ПР
Сравнение (короткие операнды)	CER	39	RR	R1,R2	ПР
Сравнение (короткие операнды)	CE	79	RX	R1,D2(X2,B2)	ПР
Деление (длинные операнды)	DDR	2D	RR	R1,R2	

Название	Мнемо-ника	Код	Фор-мат	Операнды	Примечания
Деление (длинные операнды)	DD	6D	RX	R1,D2(X2,B2)	
Деление (короткие операнды)	DER	3D	RR	R1,R2	
Деление (короткие операнды)	DE	7D	RX	R1,D2(X2,B2)	
Пополам (длинные операнды)	HDR	24	RR	R1,R2	
Пополам (короткие операнды)	HER	34	RR	R1,R2	
Загрузка (длинные операнды)	LDR	28	RR	R1,R2	
Загрузка (длинные операнды)	LD	68	RX	R1,D2(X2,B2)	
Загрузка (короткие операнды)	LER	38	RR	R1,R2	
Загрузка (короткие операнды)	LE	78	RX	R1,D2(X2,B2)	
Загрузка и проверка (длинные операнды)	LTDR	22	RR	R1,R2	ПР
Загрузка и проверка (короткие операнды)	LTER	32	RR	R1,R2	ПР
Загрузка дополнения (длинные операнды)	LCDR	23	RR	R1,R2	ПР
Загрузка дополнения (короткие операнды)	LCER	33	RR	R1,R2	ПР
Загрузка отрицательная (длинные операнды)	LNDR	21	RR	R1,R2	ПР
Загрузка отрицательная (короткие операнды)	LNER	31	RR	R1,R2	ПР
Загрузка положительная (длинные операнды)	LPDR	20	RR	R1,R2	ПР
Загрузка положительная (короткие операнды)	LPER	30	RR	R1,R2	ПР
Умножение (длинные операнды)	MDR	2C	RR	R1,R2	
Умножение (длинные операнды)	MD	6C	RX	R1,D2(X2,B2)	
Умножение (короткие операнды, длинный результат)	MER	3C	RR	R1,R2	
Умножение (короткие операнды, длинный результат)	ME	7C	RX	R1,D2(X2,B2)	
Запись в память (длинные операнды)	STD	60	RX	R1,D2(X2,B2)	
Запись в память (короткие операнды)	STE	70	RX	R1,D2(X2,B2)	
Вычитание с нормализацией (длинные операнды)	SDR	2B	RR	R1,R2	ПР
Вычитание с нормализацией (длинные операнды)	SD	6B	RX	R1,D2(X2,B2)	ПР

Продолжение табл. 1.3

Название	Мнемоника	Код	Формат	Операнды	Примечания
Вычитание с нормализацией (короткие операнды)	SER	3B	RR	R1,R2	ПР
Вычитание с нормализацией (короткие операнды)	SE	7B	RX	R1,D2(X2,B2)	ПР
Вычитание без нормализации (длинные операнды)	SWR	2F	RR	R1,R2	ПР
Вычитание без нормализации (длинные операнды)	SW	6F	RX	R1,D2(X2,B2)	ПР
Вычитание без нормализации (короткие операнды)	SUR	3F	RR	R1,R2	ПР
Вычитание без нормализации (короткие операнды)	SU	7F	RX	R1,D2(X2,B2)	ПР

Таблица 1.4

Команды арифметики с плавающей точкой повышенной точности

Название	Мнемоника	Код	Формат	Операнды	Примечания	
Сложение с нормализацией (расширенные операнды)	AXR	36	RR	R1,R2	ПР	P2
Загрузка с округлением (преобразование расширенного операнда в длинный)	LRDR	25	RR	R1,R2		P2
Загрузка с округлением (преобразование длинного операнда в короткий)	LRR	35	RR	R1,R2		P2
Умножение (расширенные операнды)	MXR	26	RR	R1,R2		P2
Умножение (длинные операнды, расширенный результат)	MXDR	27	RR	R1,R2		P2
Умножение (длинные операнды, расширенный результат)	MXD	67	RX	R1,D2(X2,B2)		P2
Вычитание с нормализацией (расширенные операнды)	SXR	37	RR	R1,R2	ПР	P2

Таблица 1.5

Команды, связанные с внутренним таймером и компаратором

Название	Мнемоника	Код	Формат	Операнды	Примечания		
Установить компаратор	SCKC	B206	S	D2(B2)	ПК	P2	
Установить внутренний таймер	SPT	B208	S	D2(B2)	ПК	P2	
Запись в память значения компаратора	STCKC	B207	S	D2(B2)	ПК	P2	
Запись в память значения внутреннего таймера	STPT	B209	S	D2(B2)	ПК	P2	

Таблица 1.6

Команды прямого управления

Название	Мнемоника	Код	Формат	Сперанды	Примечания		
Прямое чтение	RDD	85	SI	D1(B1),I2	ПК		
Прямая запись	WDR	84	SI	D1(B1),I2	ПК		

Таблица 1.7

Команды динамической переадресации

Название	Мнемоника	Код	Формат	Операнды	Примечания		
Загрузка реального адреса	LRA	B1	RX	R1,D2(X2,B2)	ПР	ПК	P2
Очистка TLB	PTLB	B20D	S			ПК	P2
Сбросить бит обращения	RRB	B213	S	D2(B2)	ПР	ПК	P2
Запись в память и модификация маски системы с логическим умножением	STNSM	AC	SI	D1(B1),I2		ПК	P2
Запись в память и модификация маски системы с логическим сложением	STOSM	AD	SI	D1(B1),I2		ПК	P2

Таблица 1.8

Команды мультипроцессирования

Название	Мнемоника	Код	Формат	Операнды	Примечания		
Установить префикс	SPX	B210	S	D2(B2)	ПР	ПК	P2
Сигнал процессору	SIGP	AE	RS	R1,R3,D2(B2)		ПК	P2
Запись в память адреса процессора	STAP	B212	S	D2(B2)		ПК	P2
Запись в память префикса	STPX	B211	S	D2(B2)		ПК	P2

Таблица 1.9

Команды условного обмена

Название	Мнемоника	Код	Формат	Операнды	Примечания	
Сравнение с обменом	CS	BA	RS	R1,R3,D2(B2)	ПР	P2
Сравнение двойное обменом	CDS	BB	RS	R1,R3,D2(B2)	ПР	P2

Таблица 1.10

Команды изменения ключа PSW

Название	Мнемоника	Код	Формат	Операнды	Примечания	
Прочитать ключ PSW	IPK	B20B	S		ПК	P2
Установить ключ PSW	SPKA	B20A	S	D2(B2)	ПК	P2

1.3. Команды Ассемблера

Команды Ассемблера предназначены для указания действий, которые должен выполнить транслятор во время трансляции программы.

Команды Ассемблера делятся на следующие группы: команды регистра базы (USING, DROP), команды определения (EQU, OPSYN, DC, DS, CCW), команды секционирования и соединения программ (START, CSECT, DSECT, DXD, COM, CXD, ENTRY, EXTRN, WXTRN), команды управления выводом распечатки (TITLE, EJECT, SPACE, PRINT), команды управления трансляцией (ICTL, ISEQ, ORG, LTORG, CNOP, COPY, PUNCH, REPRO, PUSH, POP, END), команды макросредств (ACTR, AGO, AIF, ANOP, GBLA, GBLB, GBLC, LCLA, LCLB, LCLC, MACRO, MEND, MEXIT, MNOTE, SETA, SETB, SETC).

Формат команд Ассемблера:

Название Операция Операнды

Состав команд Ассемблера в алфавитном порядке

Мнемоника команд в поле операции

Назначение, содержание поля названия и поля операндов

ACTR

Используется для установки значения счетчика максимального количества переходов, выполняемых внутри макроопределения или внутри основной части программы по операторам AGO и AIF. Без указания ACTR значение счетчика принимается равным 4096.

Название: пробел.

Операнды: арифметическое выражение, указывающее количество переходов

AGO

Используется для безусловного изменения последовательности обработки операторов исходного модуля или макроопределения.

Название: метка следования или пробел.

Операнды: метка следования

AIF	Используется для изменения последовательности обработки операторов в зависимости от результатов проверки некоторого условия. Название: метка следования или пробел. Операнды: логическое выражение в скобках, за которым следует метка следования
ANOP	Позволяет выполнять условный и безусловный переход на обработку оператора, в поле названия которого находится имя или символ переменной и невозможно поместить метку следования. Название: метка следования. Операнды: не используются
CCW	Используется для создания восьмибайтовой команды канала и размещения ее на границу двойного слова. Название: имя или пробел. Операнды: четыре операнда, разделенные запятой, определяющие содержимое команды канала: первый операнд — абсолютное выражение, определяющее код команды; второй операнд — выражение (абсолютное или перемещаемое), определяющее адрес данных и рассматриваемое как трехбайтовая константа типа A; третий операнд — абсолютное выражение, определяющее флажки в команде канала; четвертый операнд — абсолютное выражение, определяющее длину данных
CNOP	Предназначена для выравнивания значения счетчика на границу полуслова, слова или двойного слова. Название: метка следования или пробел. Операнды: два абсолютных выражения, разделенные запятой: первый операнд указывает номер байта в слове или двойном слове, на который должен быть установлен счетчик адреса (значение операнда 0, 2, 4 или 6); второй операнд указывает, находится ли байт в слове (значение операнда — 4) или в двойном слове (значение операнда — 8)
COM	Предназначена для идентификации начала или продолжения общей области. Название: имя или пробел. Операнды: не используются
COPY	Используется для вызова из библиотеки ранее составленного набора операторов языка Ассемблера (раздела) и включения его в исходный модуль, обрабатываемый транслятором. Название: пробел. Операнды: имя включаемого раздела
CSECT	Используется для идентификации начала или продолжения программной секции. Название: имя или пробел. Операнды: не используются
CXD	Выделяет выровненную на границу слова четырехбайтовую область памяти, которая будет содержать сумму длин всех внешних фиктивных секций (общая длина подсчитывается Редактором связей или Загрузчиком). Название: имя или пробел. Операнды: не используются
DC	Используется для определения констант различных типов: с фиксированной точкой, с плавающей точкой, десятичных, шестнадцатеричных, символьных, адресных. Одна команда DC позволяет определить одну или несколько констант. Название: имя или пробел. Операнды: один или несколько (до 32) операндов, разделенных запятыми

DROP	Указывает общие регистры, определенные ранее командой USING в качестве регистров базы, которые не должны в дальнейшем использоваться как регистры базы. Название: от одного до шестнадцати абсолютных выражений, разделенных запятыми. Каждое выражение определяет общий регистр
DS	Используется для резервирования областей основной памяти и присвоения имен этим областям. Название: имя или пробел. Операнды: один или несколько (32) операндов, разделенных запятыми
DSECT	Используется для идентификации начала или продолжения фиктивной секции. Название: имя или пробел. Операнды: не используются
DXD	Используется для определения внешней фиктивной секции. Название: имя. Операнды: один или несколько операндов, разделенных запятыми
EJECT	Предназначена для указания вывода на печать следующего за этой командой текста на новую страницу распечатки. Название: метка следования или пробел. Операнды: не используются
END	Используется для указания конца исходного модуля. Название: метка следования или пробел. Операнды: перемещаемое выражение или пробел
ENTRY	Предназначена для определения имен связи, которые определяются в одном исходном модуле, но могут использоваться в других исходных модулях. Название: метка следования или пробел. Операнды: одно или несколько перемещаемых имен, разделенных запятыми
EQU	Используется для определения имени путем присваивания ему значения, характеристики длины, характеристики типа и признака перемещаемости. Название: обычное имя или символ переменной. Операнды: от одного до трех абсолютных или перемещаемых выражений, разделенных запятыми
EXTRN	Предназначена для идентификации имен связи, которые используются данным исходным модулем, но определены в другом исходном модуле. Название: метка следования или пробел. Операнды: одно или несколько перемещаемых имен, разделенных запятыми
GBLA GBLB GBLC	Предназначены для определения глобальных SETA-, SETB-, SETC-переменных и присвоения им первоначальных значений. Название: пробел. Операнды: один или несколько символов переменных, разделенных запятыми. Команды GBLA, GBLB, GBLC могут находиться в любом месте макроопределения или основной части программы

ICTU	Используется для изменения стандартного формата операторов исходного модуля. Название: пробел. Операнды: от одного до трех десятичных чисел, разделенных запятыми (первый операнд определяет номер колонки начала исходного оператора, второй операнд определяет номер колонки конца исходного оператора, третий операнд указывает номер колонки продолжения исходного оператора)
ISEQ	Используется для проверки последовательности вводимых перфокарт. Название: пробел. Операнды: два десятичных числа, разделенных запятой, или пробел (пробел используется для указания прекращения проверки последовательности)
LCLA LCLB LCLC	Предназначены для определения локальных SETA-, SETB-, SETC-переменных и присвоения им первоначальных значений. Название: пробел. Операнды: один или несколько символов переменных, разделенных запятыми. Команды LCLA, LCLB, LCLC могут находиться в любом месте макроопределения или в основной части программы
LTORG	Определяет начало области основной памяти, в которую помещаются литералы, встретившиеся в исходном модуле после предыдущего оператора LTORG или после начала исходного модуля до данного оператора LTORG. Началом области является адрес первого двойного слова, следующего за оператором LTORG. Название: имя или пробел. Операнды: не используются
MACRO	Используется для указания начала макроопределения. Всегда первая команда в макроопределении. Название: пробел. Операнды: не используются (можно записать любой комментарий)
MEND	Используется для указания конца макроопределения. Всегда последняя команда в макроопределении. Название: метка следования или пробел. Операнды: не используются (можно записать любой комментарий)
MEXIT	Вызывает прекращение обработки макроопределения и переход к обработке следующего оператора после макрокоманды. Название: метка следования или пробел. Операнды: не используются
MNOTE	Порождает сообщение об ошибке, отражающее степень серьезности ошибки. Команда MNOTE может использоваться внутри макроопределения и в основной части программы. Название: метка следования или пробел. Операнды: два операнда, разделенные запятой. Первый операнд: код серьезности — число в пределах от 0 до 255 или *. Этот операнд может быть опущен. Второй операнд определяет текст сообщения и записывается как любая комбинация символов длиной не более 256 символов, заключенная в апострофы

Прототип макрокоманды (обычное имя)	Используется для определения мнемонического кода макрокоманды и названий символических параметров, которые используются в данном макроопределении. Название: символический параметр или пробел. Операнды: один или несколько символических параметров, разделенных запятыми, или пробел
OPSYN	Используется для определения кода операции машинной команды, расширенного мнемонического кода или кода операции команды Ассемблера как эквивалента другого мнемонического кода операции. Эта команда используется также и в том случае, когда необходимо, чтобы Ассемблер не воспринимал какой-либо код операции. Используются два формата команды OPSYN: 1) название: имя или код операции. Операнды: код операции; 2) название: код операции. Операнды: пробел
ORG	Используется для изменения значения счетчика адреса текущей программной секции. Название: имя или пробел. Операнды: перемещаемое выражение или пробел. Все имена, используемые в выражении, записанном в поле операндов, должны быть предварительно определены. Если операнд опущен, счетчику адреса присваивается значение на единицу больше максимального адреса, отведенного для данной секции к моменту обработки данного оператора ORG.
POP	Позволяет восстанавливать состояние команд Ассемблера PRINT и (или) USING, сохраненное самым последним оператором PUSH. Название: метка следования или пробел. Операнды: один из следующих четырех: PRINT; USING; PRINT, USING; USING, PRINT. Команда POP используется в сочетании с командой PUSH
PRINT	Используется для определения содержания распечатки исходного и объектного модулей. Название: метка следования или пробел. Операнды: от одного до трех операндов, разделенных запятыми: ON (распечатка выводится) или OFF (распечатка не выводится); GEN (печатаются все операторы, порождаемые макрокомандами) или NOGEN (не печатаются операторы, порождаемые макрокомандами); DATA (константы печатаются полностью) или NODATA (печатаются только восемь левых байт константы)
PUNCH	Предназначена для указания вывода на перфокарту данных, которые определяются в поле операндов этой команды. Перфокарты, выведенные по команде PUNCH, не являются логической частью объектного модуля. Название: метка следования или пробел. Операнды: от одного до восьмидесяти символов, заключенных в апострофы

PUSH	<p>Позволяет сохранить текущее состояние команд Ассемблера PRINT и (или) USING в специальной (стековой) области памяти, с которой Ассемблер работает по принципу: последний введенный элемент извлекается первым.</p> <p>Название: метка следования или пробел.</p> <p>Операнды: один из следующих четырех: PRINT; USING; PRINT; USING; PRINT.</p> <p>Команда PUSH используется в сочетании с командой POP</p>
REPRO	<p>Позволяет выводить на перфокарту данные, которые определяются следующим за командой REPRO оператором. Перфокарты, выведенные по командам REPRO, не являются логической частью объектного модуля.</p> <p>Название: метка следования или пробел.</p> <p>Операнды: не используются</p>
SETA	<p>Позволяет присваивать арифметическое значение SETA-переменным.</p> <p>Название: SETA- переменная.</p> <p>Операнды: арифметическое выражение</p>
SETB	<p>Позволяет присвоить логическое значение SETB- переменной.</p> <p>Название: SETB- переменная.</p> <p>Операнды: 0 или 1, или логическое выражение, заключенное в скобки</p>
SETC	<p>Позволяет присвоить символьные значения SETC-переменным.</p> <p>Название: SETC-переменная.</p> <p>Операнды: символьное выражение</p>
SPACE	<p>Используется для указания пустых строк в распечатке.</p> <p>Название: метка следования или пробел.</p> <p>Операнды: десятичный самоопределенный терм или пробел (десятичное число указывает количество пустых строк; пробел в поле операндов означает пропуск на одну строку; если число строк превышает число возможных строк, оставшихся на данной странице распечатки, то команда выполняется подобно команде EJECT)</p>
START	<p>Используется для присваивания имени первой (или единственной) программной секции исходного модуля. Может использоваться также и для установки начального значения счетчика адреса первой программной секции исходного модуля.</p> <p>Название: имя или пробел.</p> <p>Операнды: самоопределенный терм или пробел (значение используется Ассемблером в качестве начального значения счетчика адреса для исходного модуля; если операнд опущен, начальное значение счетчика адреса устанавливается в нуль)</p>
TITLE	<p>Предназначена для идентификации распечатки с результатами трансляции и перфокарт объектного модуля.</p> <p>Название: имя или пробел.</p> <p>Операнды: от одного до ста символов, заключенных в апострофы.</p> <p>Имя в поле названия команды TITLE может содержать от одной до восьми букв и (или) цифр в любой последовательности. Имя перфорируется в позициях 73—80 всех выводимых при трансляции перфокарт, за исключением тех, которые выводятся по командам PUNCH и REPRO. Содержимое поля названия и поля операндов оператора TITLE печатается в виде заголовка на каждой странице распечатки с результатами трансляции</p>

USING

Предназначена для указания номеров общих регистров, которые можно использовать в качестве регистров базы. Также указывает Ассемблеру значения базовых адресов, которые будут находиться в этих регистрах в момент выполнения программы.

Название: метка следования или пробел.

Операнды: абсолютное или перемещаемое выражение, за которым следует от двух до семнадцати абсолютных выражений, разделенных запятыми. Первое выражение определяет значение, которое должно использоваться Ассемблером в качестве базового адреса. Каждое следующее выражение определяет общий регистр

WXTRN

Используется для идентификации слабых внешних имен (для имен, идентифицированных командой WXTRN, не используется механизм автоматического вызова библиотеки Редактора связей или Загрузчика).

Название: метка следования или пробел.

Операнды: одно или несколько перемещаемых имен, разделенных запятыми

1.3.1. Общие сведения о константах

Команда Ассемблера DC позволяет определить константы следующих типов: символьные, шестнадцатеричные, двоичные, с фиксированной точкой, с плавающей точкой, десятичные, адресные.

Для определения константы используются четыре поля, расположенные в операнде в следующей последовательности: поле 1 — коэффициент кратности, поле 2 — тип константы, поле 3 — модификаторы, поле 4 — константа или константы. Первое или третье поля могут быть опущены. Для большинства типов констант в четвертом поле может быть указано более одной константы. В этом случае каждая константа определяется одним и тем же типом, указанным в поле 2. Константы, определяемые в одном операнде, имеют, следовательно, одни и те же характеристики, но каждый операнд команды DC может определять различные типы констант.

Обобщенные сведения о константах приведены в табл. 1.11. Диапазон значений модификатора длины для констант всех типов указывается в байтах независимо от того, задается модификатор длины в байтах или битах. Все сведения в равной мере относятся и к команде Ассемблера DS.

1.4. Макросредства Ассемблера

Макросредства, входящие в язык Ассемблера, позволяют упростить программирование путем однократного описания повторяющихся последовательностей операторов языка Ассемблера и последующего вызова этих последовательностей с помощью операторов, представляющих собой макрокоманды.

Набор операторов языка Ассемблера, вызываемый по макрокоманде, представляет собой *макроопределение*.

Таблица 1.11

Константы	Тип	Невная длина, байт	Выравни- вание	Диапазон модифи- катора длины	Представление константы	Число констант в операнде	Диапазон порядка	Диапазон масштаба	Усече- ние или добав- ление
1	2	3	4	5	6	7	8	9	10
Символьная	С	По требо- ванию	Байт	От 1 до 256	Символы ДКОИ	Одна	—	—	Справа
Шестнадцатеричная	Х	По требо- ванию	Байт	От 1 до 256	Шестнадца- теричные цифры	Несколько	—	—	Слева
Двоичная	В	По требо- ванию	Байт	От 1 до 256	Двоичные цифры	Несколько	—	—	Слева
С фиксированной точкой (число с фиксированной точкой)	F	4	Слово	От 1 до 8	Десятичные цифры	Несколько	От минус 85 до плюс 75	От минус 187 до плюс 346	Слева
С фиксированной точкой (число с фиксированной точкой)	H	2	Полу- слово	От 1 до 8	Десятичные цифры	Несколько	От минус 85 до плюс 75	От минус 187 до плюс 346	Слева
С плавающей точкой (ко- роткое число с плаваю- щей точкой)	E	4	Слово	От 1 до 8	Десятичные цифры	Несколько	От минус 85 до плюс 75	От 0 до 14	Справа
С плавающей точкой (длинное число с плаваю- щей точкой)	D	8	Двойное слово	От 1 до 8	Десятичные цифры	Несколько	От минус 85 до плюс 75	От 0 до 14	Справа

Продолжение табл. 1.11

Константы	Тип	Независимая длина, байт	Выравни- вание	Диапазон модифи- катора длины	Представление константы	Число констант в операнде	Диапазон порядка	Диапазон масштаба	Усече- ние или добав- ление
1	2	3	4	5	6	7	8	9	10
С плавающей точкой (длинное число с плаваю- щей точкой)	L	16	Двойное слово	От 1 до 16	Десятичные цифры	Несколько	От минус 85 до плюс 75	От 0 до 28	Справа
Десятичная (упакован- ный десятичный формат)	P	По требо- ванию	Байт	От 1 до 16	Десятичные цифры	Несколько	—	—	Слева
Десятичная (десятичный формат с зоной)	Z	По требо- ванию	Байт	От 1 до 16	Десятичные цифры	Несколько	—	—	Слева
Адресная (значение адре- са, обычно слово)	A	4	Слово	От 1 до 4 ³	Абсолютное выражение	Несколько	—	—	Слева
				3 или 4	Перемещаемое или составное перемещаемое выражение	Несколько	—	—	Слева
Адресная (значение адре- са, обычно полуслово)	Y	2	Полу- слово	1 или 2 ³ Только 2	Абсолютное выражение Перемещаемое или составное перемещаемое выражение	Несколько	—	—	Слева
						Несколько	—	—	Слева

Константы	Тип	Неявная длина, байт	Выравнивание	Диапазон модификатора длины	Представление константы	Число констант в операнде	Диапазон порядка	Диапазон масштаба	Усечение или добавление
1	2	3	4	5	6	7	8	9	10
Адресная (область, резервируемая для внешнего адреса)	V	4	Слово	3 или 4	Перемещаемое имя	Несколько	—	—	Слева
Адресная (значение адреса в виде регистра базы и смещения)	S	2	Полуслово	Только 2	Одно абсолютное или перемещаемое выражение или два абсолютных выражения в виде: выражение (выражение)	Несколько	—	—	—
Адресная (область, резервируется для значения смещения внешней фиктивной секции)	Q	4	Слово	От 1 до 4	Имя, называемое оператор DXD или DSECT	Несколько	—	—	Слева

- Примечания: 1. Для констант типа C и X в команде Ассемблера DS может указываться длина до 65 535 байт.
 2. Если усекаются значащие биты или если указанная константа не помещается в область, отводимую согласно явной или неявной длине, то будут отмечены ошибки.
 3. Модификатор длины в битах применим только к абсолютному выражению.

Операторы, которые включаются в программу в результате обработки макроопределения по макрокоманде, называются *макрорасширением*.

1.4.1. Макроопределение

Набор операторов языка Ассемблера, составляющих макроопределение, включает: начальный оператор макроопределения (оператор MACRO); оператор прототипа; последовательность операторов, составляющих собственно макроопределение (модельные операторы, операторы COPY, MEXIT, MNOTE, команды условного ассемблирования); конечный оператор макроопределения (оператор MEND). Операторы MACRO, COPY, MEXIT, MNOTE, MEND, оператор прототипа и другие операторы — команды Ассемблера — описаны в 1.3. В зависимости от формата записи операндов прототипа макроопределения разделяют на позиционные, ключевые, смешанные.

Модельные операторы предназначены для создания из них операторов языка Ассемблера во время макрогенерации и условного ассемблирования. Указывая символами переменных точки подстановки в модельных операторах, можно изменять содержимое операторов, создаваемых из модельных операторов. В качестве модельных операторов могут использоваться машинные команды, команды Ассемблера (за исключением команд условного ассемблирования ICTL, OPSYN, MACRO, MEND), макрокоманды и операторы комментариев. Модельные операторы в макроопределении следуют за оператором прототипа.

Внутри макроопределений в модельных операторах могут использоваться все три типа символов переменных, допускаемых в макросредствах языка Ассемблера: символические параметры, системные символы переменных (&SYSDATE — дата трансляции, &SYSTIME — время трансляции, &SYSPARM — системная строка символов, &SYSNDX — порядковый номер макрокоманды, &SYSECT — название текущей секции, &SYSLIST — операнд макрокоманды), SET-переменные.

1.4.2. Макрокоманда

Макрокоманда используется для вызова макроопределения. Результат обработки макроопределения Ассемблером существенно зависит от операндов макрокоманды, так как значения операндов макрокоманды замещают используемые в макроопределении символические параметры. Размещение и порядок операндов в макрокоманде определяются размещением и порядком символических параметров в поле операндов оператора прототипа. В связи с этим различаются позиционные, ключевые и смешанные макрокоманды.

В качестве операнда макрокоманды можно использовать любую последовательность символов, количество которых не долж-

Таблица 1.12

Информация о выражении	Выражение		
	арифметическое	символьное	логическое
Операнды	Самоопределенный терм. Характеристика длины, масштабирования, целой части, количества символов, количества операндов. SETA-, SETB-переменные, SETC-переменная ¹ . Символический параметр ² : &SYSLIST(n) ² &SYSLIST(n,m) ² &SYSNDX &SYSPARM ¹	Любая комбинация символов, заключенная в апострофы. Любой символ переменной, заключенный в апострофы. Соединение символов переменных и других символов, заключенное в апострофы. Подстрока символов. Характеристика типа	0 или 1 SETB-переменная. Арифметическое отношение ³ . Символьное отношение ⁴ . Самоопределенный терм
Операции	Сложение (+), вычитание (—), умножение (×), деление (/), положительный (+), отрицательный (—), допускаются скобки	Соединение	AND (И), OR (ИЛИ), NOT (НЕТ), допускаются скобки
Область значений	От -2^{31} до $+2^{31}-1$	От 0 до 255 символов	0 (ложь) или 1 (истина)
Использование	Операнд оператора SETA. Арифметическое отношение ³ . Обозначение индекса SET-переменных. Обозначение индексов &SYSLIST. Обозначение подстроки. Обозначение списка. Операнд оператора ACTR	Операнд оператора SETC. Символьное отношение ⁴	Операнд оператора SETB. Операнд оператора AIF

¹ Значение символа переменной должно представляться не более чем десятью десятичными цифрами (от 0 до 2147483647).

² Значение символа переменной должно быть самоопределенным термом.

³ Арифметическое отношение состоит из двух арифметических выражений, связанных операндами GT, LT, EQ, NE, GE или LE.

⁴ Символьное отношение состоит из двух символьных отношений, связанных операциями GT, LT, EQ, NE, GE или LE. Обозначение характеристики типа и обозначение подстроки могут использоваться в символьных отношениях. Максимальный размер символьных выражений, которые могут сравниваться, — 255 символов. Если два символьных выражения имеют неодинаковый размер, то при сравнении более короткое считается меньшим по сравнению с более длинным.

Таблица 113

Характеристика	Обозначение	Использование		
		операнд характеристики ¹	характеристика типа операнда	место использования
Тип	T'	Имя вне макроопределений; символический параметр, &SYSLIST(n) или &SYSLIST(n,m) внутри макроопределений; SET-переменные, &SYSTIME, &SYSDATE, &SYSECT, &SYSNDX, &SYSPARM	Любое значение	Поле операндов оператора. Символьное отношение
Длина	L'	Имя вне макроопределений; символический параметр, &SYSLIST(n) или &SYSLIST(n,m) в макроопределениях	Любая буква за исключением N, O, T, U и Q	Арифметическое выражение
Масштабирование	S'	То же	Буквы H, F, G, D, E, L, K, P, Z	То же
Целая часть	I'	»	Буквы H, F, G, D, E, L, K, P, Z	»
Количество символов	K'	Символический параметр, &SYSLIST(n) или &SYSLIST(n,m) внутри макроопределений; SET-переменная, все системные символы переменных	Любая буква	Арифметическое выражение
Количество операндов	N'	Символический параметр, &SYSLIST(n) или &SYSLIST(n,m) внутри макроопределений	Любая буква	То же

¹ Операнд характеристики — элемент, характеристика которого используется.

Таблица 1.14

Символ переменной	Определение	Первоначальное значение	Измененные значения	Использование
Символический параметр ¹	В операторе прототипа	Соответствующий операнд макрокоманды	Постоянно для данного макроопределения	В арифметических ² , символических и логических выражениях
SETA-переменная	В операторах LCLA, GBLA	0	Оператором SETA	В арифметических, символических и логических выражениях
SETB-переменная	В операторах LCLB, GBLB	0 (ложь)	Оператором SETB	В арифметических, символических и логических выражениях
SETC-переменная	В операторах LCLC, GBLC	Нулевое символическое значение	Оператором SETC	В арифметических ³ , символических и логических выражениях
&SYSNDX ¹	Ассемблером	Номер макрокоманды	Постоянно для одного макроопределения; уникально для каждой макрокоманды	В арифметических, символических и логических выражениях
&SYSECT ¹	Ассемблером	Имя программной секции, в которой появляется макрокоманда	Постоянно для одного макроопределения; устанавливается посредством операторов CSECT, DSECT, START, COM	В символических и логических выражениях
&SYSLIST ¹	Ассемблером	—	—	N/SYSLIST в арифметических и логических выражениях
&SYSLIST(n) ¹ &SYSLIST(n,m) ¹	Ассемблером	Соответствующий операнд макрокоманды	Постоянно в одном макроопределении	В арифметических ² , символических и логических выражениях
&SYSPARM	Поле PARM	Определяется пользователями ЭВМ или нулевое символическое значение	Постоянно на протяжении всей трансляции	В арифметических ³ , символических или логических выражениях
&SYSTIME	Ассемблером	Системное время	Постоянно на протяжении всей трансляции	В символическом выражении
&SYSDATE	Ассемблером	Системная дата	Постоянно на протяжении всей трансляции	В символическом выражении

¹ Может использоваться только в макроопределениях.² Только если значением является самоопределенный терм.³ Только если значение представляется не более чем десятью десятичными цифрами (от 0 до 2147483647).

но быть больше 255. При этом должны быть выполнены правила использования в операнде апострофов, круглых скобок, знака равно, символов &, запятых и пробелов.

Операнд макрокоманды может быть списком. Списки представляют программисту удобный способ обращения к группе операндов макрокоманды как к одному операнду или к каждому операнду в группе операндов. Список состоит из одного или нескольких операндов, разделенных запятыми и заключенных в парные скобки.

1.4.3. Условное ассемблирование

Возможности условного ассемблирования можно использовать как в макроопределении, так и в основной части программы. В процессе условного ассемблирования могут использоваться SET-переменные, характеристики, выражения условного ассемблирования, метки следования. В табл. 1.12 описаны выражения, которые могут использоваться в командах условного ассемблирования. В табл. 1.13 описаны характеристики, которые могут использоваться в этих выражениях. В табл. 1.14 приведены символы переменных, используемые в каждом выражении.

Команды оператора ОС ЕС ЭВМ

2.1. Общие сведения

Эффективность работы компонентов операционной системы ОС ЕС ЭВМ определяется правильным выбором режима ее работы в зависимости от конфигурации технических средств и предлагаемых к выполнению задач. Решения, определяющие режим работы операционной системы, принимает оператор ОС ЕС ЭВМ. Он осуществляет начальную загрузку операционной системы, запуск программ системного ввода и вывода, следит за состоянием технических средств, управляет программными средствами аппаратного контроля, предпринимает необходимые действия в случае сбоев программного обеспечения. Оператор ОС ЕС ЭВМ вмешивается в работу операционной системы, перераспределяет периферийные устройства, отменяет выполнение одних заданий и изменяет приоритеты других.

Оператор ОС ЕС ЭВМ взаимодействует с операционной системой с помощью специально выделенных для этого устройств ввода-вывода, называемых консолями. В качестве консолей используются пишущие машинки ЕС-7070, ЕС-7073, ЕС-7077, алфавитно-цифровые дисплеи ЕС-7061, ЕС-7063, ЕС-7066, ЕС-7927-01, графические дисплеи ЕС-7064, ЕС-7065 и совокупности устройств ввода с перфокарт и вывода на печать (например, ЕС-6012 и ЕС-7032 или ЕС-6016 и ЕС-7035).

Оператор ОС ЕС ЭВМ вводит с консоли команды, предписывающие операционной системе определенные действия. Операционная система выводит на консоль сообщения. Некоторые из них требуют от оператора ОС ЕС ЭВМ действий по установке томов или включению устройств, другие — ответа о принятых решениях, третьи — информируют о ходе выполнения заданий, состоянии периферийных устройств и используемых ресурсах.

Команды оператора ОС ЕС ЭВМ подразделяются на четыре группы: информационные команды (группа 0), команды управления операционной системой (группа 1), команды управления периферийными устройствами (группа 2), команды управления консолями (группа 3). Каждая команда состоит из поля операции, поля операндов и поля комментариев. Все три поля должны отделяться друг от друга не менее чем одним пробелом. В случае

отсутствия информации в поле операндов комментарии должны следовать через один пробел и запятую после поля операции. Операнды должны отделяться друг от друга запятыми. Операнды, записанные прописными буквами или цифрами, следует кодировать как указано. Выражения в поле операндов, записанные строчными буквами, указывают местоположения конкретных значений параметров, которые они определяют. Квадратные скобки ограничивают необязательную информацию. Из возможных значений операндов, заключенных в фигурные скобки, должно быть выбрано одно.

2.2. Информационные команды

Информационные команды позволяют оператору ОС ЕС ЭВМ следить за ходом работы компонентов операционной системы ОС ЕС и заданий пользователей. С помощью информационных команд оператор ОС ЕС ЭВМ получает сведения о занятости периферийных устройств, заносит необходимую информацию в системный журнал и отвечает на сообщения операционной системы.

К группе информационных команд относятся BRDCST, CONTROL, DISPLAY, LOG, MONITOR, MSG, MSGRT, REPLY, SHOW и STOPMN. Команды BRDCST, MSG и SHOW в свою очередь являются командами управления работой системы диалогового удаленного ввода заданий (ДУВЗ) и описываются в разделе 2.11.

2.2.1. Команда CONTROL

Команда CONTROL описывается в этом разделе полностью. Если эта команда употребляется с операндом L, то она относится к группе команд управления консолями. В зависимости от формата и используемых операндов команда CONTROL обеспечивает удаление сообщений с экрана дисплей-консоли, дает возможность определить назначение функциональной клавиатуры для дисплеев ЕС-7064 и ЕС-7927-01, позволяет изменять режим работы дисплей-консоли.

Удаление сообщений с экрана дисплей-консоли. Команда CONTROL для обеспечения удаления сообщений с экрана дисплей-консоли имеет следующий формат:

$$\left\{ \begin{array}{c} \text{CONTROL} \\ K \end{array} \right\} \left\{ \begin{array}{l} E \left[\begin{array}{l} , nn [, mm] \\ , \frac{SEG}{F} \end{array} \right] \\ S \left[\begin{array}{l} , DEL=x \\ , SEG=nn \\ , CON=x \\ , RNUM=nn \\ , RTME=nnn \\ , REF \end{array} \right] \end{array} \right\}$$

где операнды определяют:

Е — применение команды CONTROL для удаления сообщений;
пп, [mm] — номер строки удаляемого сообщения или группу строк удаляемых сообщений от пп до mm;

SEG — удаление сегмента сообщений;

F — удаление сообщений, идентифицированных вертикальной или горизонтальной чертой;

S — применение команды CONTROL для изменения режима удаления сообщений;

DEL=x — метод удаления сообщений; x может принимать любое из следующих значений:

N — метод удаления сообщений только с помощью команды CONTROL, светового пера или курсора,

Y — метод автоматического удаления сообщений,

R — метод свертки всех сообщений,

RD — метод свертки сообщений, не требующих действия;

SEG=пп — размер сегмента в пп строк;

CON=x — режим удаления сообщений; x может принимать одно из следующих значений:

Y — режим удаления с подтверждением,

N — режим удаления без подтверждения.

RNUM=пп — размер сегмента свертываемых сообщений;

пп — число строк. Размер сегмента не должен превышать размера области сообщений.

RTME=ппп — интервал времени, по истечении которого будет происходить свертывание сообщений; ппп — число секунд.

REF — вывод на экран дисплей-консоли текущих значений параметров DEL, SEG, RTME, RNUM и CON.

При автоматическом удалении сообщений, операционная система удаляет сообщения, если область сообщений заполнена и требуется вывод на дисплей-консоль новых сообщений. Автоматически удаляются только сообщения, помеченные вертикальной чертой.

При свертке (DEL=R или DEL=RD) операционная система автоматически удаляет сообщения с экрана дисплей-консоли целыми сегментами. Сегмент сообщений удаляется по истечении заданного интервала времени в том случае, если требуется вывод на дисплей-консоль новых сообщений. Размер сегмента для удаления сообщений устанавливается равным числу сообщений, ожидающих вывода, но не больше определенного операндом RNUM.

Если команда CONTROL выдается без операндов, то подразумевается K E, SEG. Если команда CONTROL выдается в формате K S, то на экран дисплей-консоли выводится команда CONTROL со всеми текущими значениями параметров.

Нумерация сообщений. Для удобства удаления сообщений с помощью команды CONTROL можно задать нумерацию сообщений. Для этого следует воспользоваться командой CONTROL в следующем формате:

$$\left\{ \begin{array}{c} \text{CONTROL} \\ K \end{array} \right\} \left\{ \begin{array}{c} \{D, N[, \text{HOLD}]\} \\ \{E, N\} \end{array} \right\}$$

где операнды определяют:

D, N — нумерацию строк на экране дисплей-консоли. Если дополнительно задан операнд HOLD, то номера строк будут находиться на экране независимо от сообщений. Если операнд HOLD опущен, то номера строк будут пропадать после удаления сообщений;

E, N — отмену нумерации строк на экране дисплей-консоли.

Определение назначения функциональной клавиатуры. Команда CONTROL для определения назначения функциональной клавиатуры (ФК) имеет следующий формат:

$$\left\{ \begin{array}{c} \text{CONTROL} \\ \text{K} \end{array} \right\} \left\{ \begin{array}{l} \text{N, PFK} = (\text{nn} \left\{ \begin{array}{l} \text{, CMD='команда [; команда]'} \\ \text{, KEY=ll [, mm ...]} \end{array} \right\}) [\text{, CON=x}] \\ \text{D, PFK} \\ \text{E, PFK} \end{array} \right\}$$

где операнды определяют:

N, PFK — применение команды CONTROL для изменения состава команд, закрепленных за клавишей ФК;

nn — номер клавиши ФК;

CMD='команда [; команда]' — закрепление за клавишей nn одной или группы команд, заданных в параметрах. Количество символов в параметрах не должно превышать 101;

KEY=ll, [, mm ...] — закрепление за клавишей nn команд, ранее закрепленных за клавишами ll, mm ... Всего может быть задано до 52 номеров клавиш;

CON=x — режим ввода команд, где x может принимать одно из следующих значений:

Y — режим ввода с подтверждением,

N — режим ввода без подтверждения;

D, PFK — применение команды CONTROL для вывода на экран дисплей-консоли номеров клавиш ФК;

E, PFK — применение команды CONTROL для удаления номеров ФК с экрана дисплей-консоли.

Определение характеристик областей информации состояния. Команда CONTROL для изменения или вывода на экран характеристик областей информации состояния имеет следующий формат:

$$\left\{ \begin{array}{c} \text{CONTROL} \\ \text{K} \end{array} \right\} \left\{ \begin{array}{l} \text{A} \left[\begin{array}{l} \text{, nn [, mm ...] [, L=cc]} \\ \text{, NONE [, L=cc]} \\ \text{, REF} \end{array} \right] \\ \text{D, F} \left[\begin{array}{l} \text{, L=x]} \\ \text{id [, L=cc]} \end{array} \right] \\ \text{C, D} \\ \text{E, D} \left[\begin{array}{l} \text{, L=x]} \\ \text{, L=x]} \end{array} \right] \\ \text{D, H} \\ \text{D, U} \left[\begin{array}{l} \text{, L=x]} \\ \text{, UTME=nnn} \end{array} \right] \\ \text{M} \left[\begin{array}{l} \text{, REF} \end{array} \right] \end{array} \right\}$$

где операнды определяют:

A — применение команды CONTROL для изменения или вывода на экран дисплей-консоли характеристик областей информации

состояния. Области информации состояния предназначены для вывода информации, получаемой в ответ на команды DISPLAY или MONITOR;

nn [mm] — количество областей информации состояния и размер каждой из них в строках. Первый параметр — размер самой нижней (на экране) области, второй — следующей области (снизу вверх) и т. д. Количество строк всех областей информации состояния не может превышать количества строк области сообщений;

L=cc — номер дисплей-консоли, работающей в режиме вывода, характеристики областей информации состояния которой необходимо изменить. Если этот операнд опущен, то команда CONTROL относится к дисплей-консоли, с которой вводится;

NONE — отмену областей информации состояния для дисплей-консоли, номер которой задан следующим операндом;

REF — вывод на экран дисплей-консоли характеристик областей информации состояния в формате команды CONTROL;

D, F — применение команды CONTROL для вывода на экран дисплей-консоли следующего кадра информации состояния;

L=x — номер дисплей-консоли и (или) идентификатор области информации состояния;

C, D — применение команды CONTROL для отмены вывода на экран дисплея информации состояния;

id — идентификатор в заглавной строке информации состояния, которую необходимо удалить;

E, D — применение команды CONTROL для удаления информации состояния;

D, H — применение команды CONTROL для приостановки изменения информации состояния о выполнении заданий (динамической информации состояния);

D, U — применение команды CONTROL для возобновления изменения информации состояния о выполнении заданий;

M — применение команды CONTROL для определения интервала времени, по истечении которого должна выводиться информация состояния о выполнении заданий:

UTME=ppp — интервал времени в секундах; ppp — число секунд;

REF — вывод на экран дисплей-консоли текущего значения интервала времени.

Управление режимом работы дисплей-консоли. Команда CONTROL для изменения режима работы дисплей-консоли имеет следующий формат:

$$\left\{ \begin{array}{c} \text{CONTROL} \\ K \end{array} \right\} V, USE = \left\{ \begin{array}{c} SD \\ MS \\ FC \end{array} \right\} [L=cc]$$

где операнды определяют:

V — применение команды CONTROL для изменения режима работы дисплей-консоли;

USE — новый режим работы, где

SD — режим вывода информации состояния,

MS — режим вывода сообщений,

FC — режим ввода-вывода.

L=сс — номер дисплей-консоли, режим работы которой необходимо изменить.

С помощью этой команды можно изменить режим работы любой дисплей-консоли, кроме основной.

2.2.2. Команда DISPLAY

Команда DISPLAY позволяет оператору ОС ЕС ЭВМ получить информацию о количестве и состоянии периферийных устройств, входящих в конфигурацию операционной системы, выводить на консоль информацию о содержимом системных очередей и группах команд, доступных для ввода с консолями операционной системы с мультиконсольным обеспечением.

Формат команды:

{DISPLAY}	{D}	{	T	{	[,TP ,GRAPHIC ,TAPE ,DASD ,UR]	{	[,ONLINE]	{	[,адрес-устройства] [,число устройств]	}	}
			A								
			U								
			CONSOLES								
			R								
			Q [=список]								
			N [=список]								
			задание [,T]								
			SQA								
			C,K								
			PFK								

где операнды определяют:

T — вывод на консоль сообщения о времени дня;

A — вывод на консоль сообщения о выполняемых заданиях;

U — вывод на консоль сообщения о состоянии периферийных устройств:

TP — устройства телеобработки,

GRAPHIC — графические устройства,

TAPE — накопители на магнитной ленте,

DASD — устройства прямого доступа,

UR — устройства ввода-вывода символьной информации;

ONLINE — вывод на консоль информации о тех периферийных устройствах, которые находятся в оперативном состоянии;

OFFLINE — вывод на консоль информации о тех периферийных устройствах, которые находятся в автономном состоянии;

адрес-устройства — вывод на консоль информации о состоянии тех периферийных устройств, адреса которых не меньше значения этого параметра. Если этот параметр опущен, то на консоль будет выведена информация о состоянии периферийных устройств начиная с адреса 000;

число-устройств — число периферийных устройств, информация о которых должна быть выведена на консоль. Если этот параметр опущен, то на консоль будет выведена информация не более чем о 100 периферийных устройствах. Пропуск промежуточных параметров операнда U следует отмечать запятыми;

CONSOLES — вывод на консоль информации о группах команд, которые можно вводить с консолей операционной системы с мультиконсольным обеспечением;

R — вывод на консоль информации о сообщениях, оставшихся без ответа оператора ОС ЕС ЭВМ;

Q — вывод на консоль информации о содержимом системных очередей;

N — вывод на консоль всех имен заданий, находящихся в системных очередях:

параметр «список» принимает следующие значения:

имя класса входной системной очереди от A до O. Указание этого параметра вызывает вывод на консоль информации о содержимом заданной входной системной очереди;

SOUT — вывод на консоль информации о содержимом выходных системных очередей;

HOLD — вывод на консоль информации о содержимом очереди задержанных заданий;

BRDR — вывод на консоль информации о содержимом очереди заданий, введенных через систему разделения времени.

Все перечисленные значения параметра «список» или некоторые из них можно задать одновременно. В этом случае они должны быть разделены между собой запятыми и весь параметр заключен в круглые скобки;

задание — определяет имя задания, о котором необходимо получить информацию. Если имя задания — JOBNames, PFK, STATUS A, R, Q, N, T, U, SPACE, DSNAME, CONSOLES, USER или SESS, то оно должно быть заключено в круглые скобки;

T — включение времени суток в сообщение о задании;

SQA — вывод на консоль сообщения о верхней и нижней границах области системных очередей и размер свободного места в ней.

C, K — вывод операндов команды CONTROL;

PFK — вывод на дисплей-консоль номеров клавиш ФК, закрепленных за ними команд и информации о режиме их ввода.

2.2.3. Команда LOG

Команда LOG позволяет оператору ОС ЕС ЭВМ занести информацию в системный журнал. Формат команды

$$\left. \begin{array}{l} \text{LOG} \\ \text{L} \end{array} \right\} \text{'текст'}$$

где текст — информация (до 128 символов) для занесения в системный журнал.

2.2.4. Команда MONITOR

Команда MONITOR обеспечивает вывод на консоль информации о времени начала выполнения и завершения каждого задания и обрабатываемых им наборах данных. Для дисплей-консоли с помощью этой команды можно задать вывод на экран динамической информации о выполняющихся заданиях. Формат команды:

$$\left\{ \begin{array}{l} \text{MONITOR} \\ \text{MN} \end{array} \right\} \left\{ \begin{array}{l} \text{JOBNAMES [T]} \\ \text{DSNAME} \\ \text{SPACE} \\ \text{STATUS} \\ \text{A[L=x]} \end{array} \right\}$$

где операнды определяют:

JOBNAMES — вывод на консоль оператора ЭВМ сообщений о времени начала выполнения и завершения заданий;

T — включение в сообщения, определяемые параметром JOB-
NAMES, времени суток;

DSNAME — включение имени постоянного набора данных, используемого в пункте задания и расположенного на томе прямого доступа, в сообщение о снятии или установке тома;

SPACE — включение информации о свободном месте на томе прямого доступа в сообщение о снятии тома;

STATUS — вызывает при завершении пункта задания вывод на консоль имен наборов данных и регистрационных номеров томов, на которых эти наборы данных расположены. Обеспечивается вывод на консоль имен наборов данных, которые использовались в пункте задания и имели диспозицию KEEP, CATLG, UNCTLG;

A — вывод на заданную дисплей-консоль динамической информации о выполняющихся заданиях.

L=x — номер дисплей-консоли, для которой задан вывод динамической информации о выполняющихся заданиях, и (или) идентификатор ее области информации состояния.

2.2.5. Команда MSGRT

Команда MSGRT позволяет направить сообщения, выводимые в ответ на команды DISPLAY, MONITOR и CONTROL на заданную консоль и (или) в указанную область информации состояния, если это дисплей-консоль. Формат команды:

$$\left\{ \begin{array}{l} \text{MSGRT} \\ \text{MR} \end{array} \right\} \left\{ \left(\left\{ \begin{array}{l} \text{D= (операнд, ...) [MN=A] [K]} \\ \text{MN=A[,D=операнд [K]]} \\ \text{K[,D=операнд [MN=A]]} \\ \text{REF} \end{array} \right\} \right) \left[\text{L=x} \right] \right\}$$

где операнды определяют:

D = (операнд, ...) — сообщения, которые должны выводиться на заданную консоль в ответ на команду DISPLAY. В скобках указываются операнды команды DISPLAY;

MN = A — сообщения, которые должны выводиться на заданную консоль, в ответ на команду MONITOR.

K — действие команды CONTROL для заданной консоли;

REF — вывод на заданную консоль информации о распределении сообщений по консолям операционной системы с мультиконсольным обеспечением;

L = x — номер консоли, на которую направляются сообщения, и (или) идентификатор ее области информации состояния, если это дисплей-консоль.

2.2.6. Команда REPLY

Команда REPLY предназначена для ввода ответа на сообщения, которые его требуют. Формат команды:

$$\left\{ \begin{array}{c} \text{REPLY} \\ R \end{array} \right\} \text{ идентификатор, 'текст'}$$

где операнды определяют:

идентификатор — десятичное число от 0 до 99, представляющее собой идентификатор сообщения;

текст — текст ответа.

2.2.7. Команда STOPMN

Команда STOPMN отменяет действие команды MONITOR. Формат команды:

$$\left\{ \begin{array}{c} \text{STOPMN} \\ \text{PM} \end{array} \right\} \left\{ \begin{array}{l} \text{JOBNAME} \\ \text{STATUS} \\ \text{DSNAME} \\ \text{SPACE} \\ \text{A}[L=x] \end{array} \right\}$$

где операнды определяют:

JOBNAME — отмену вывода на консоль сообщений о запуске и завершении заданий;

STATUS — отмену вывода на консоль имен наборов данных с диспозицией KEEP, CATLG или UNCTLG;

DSNAME — отмену вывода имени постоянного набора данных в сообщениях о снятии или установке томов прямого доступа;

SPACE — отмену вывода информации о свободном месте на томе прямого доступа в сообщении о снятии тома;

A — отмену вывода на заданную дисплей-консоль динамической информации состояния о выполнении заданий;

L = x — номер дисплей-консоли, для которой отменяется вывод динамической информации состояния о выполнении заданий и (или) идентификатор ее области информации состояния.

2.3. Команды управления операционной системой

Команды управления операционной системой предоставляют оператору ОС ЕС ЭВМ возможность управлять распределением основной памяти (MFT), выполнением системных программ и заданий пользователей. Оператор ОС ЕС ЭВМ с помощью команд управления операционной системой осуществляет запуск и останов системных программ, изменяет параметры этих программ в процессе работы или отменяет их выполнение. Эти команды позволяют задерживать выполнение всех или определенных заданий во входных системных очередях, изменять классы и приоритеты заданий и отменять их выполнение.

К группе команд управления операционной системой относятся команды: CANCEL, CENOUT, DEFINE, HALT, HOLD, MODIFY, RELEASE, RESET, SET, START, STOP, SWITCH, USERID, WRITELOG.

Команды CENOUT и USERID являются в свою очередь командами управления работой системы ДУВЗ и описываются в разделе 2.11.

2.3.1. Команда CANCEL

Команда CANCEL вызывает прекращение выполнения задания или отменяет ранее выданную команду MOUNT. Формат команды:

$$\left\{ \begin{array}{l} \text{CANCEL} \\ \text{C} \end{array} \right\} \left\{ \begin{array}{l} \text{задание} \left[\begin{array}{l} \text{,DUMP} \\ \text{,ALL} \\ \text{,IN[=класс]} \\ \text{,OUT[=класс]} \end{array} \right] \\ \text{идентификатор} \\ \text{типовое-имя} \\ \text{адрес-устройства} \end{array} \right\}$$

где операнды определяют:

задание — имя задания, выполнение которого необходимо отменить;

DUMP — возможность получить дамп памяти в момент отмены задания. Если в задании не определен ни оператор DD с именем SYSABEND, ни оператор DD с именем SYSUDUMP, то на устройство программы системного вывода будет выдан индикативный дамп;

ALL — уничтожение всех наборов данных задания во входных и выходных системных очередях;

IN=класс — поиск и уничтожение всех наборов данных задания во входных системных очередях, определенных параметром «класс»;

OUT=класс — поиск и уничтожение всех наборов данных задания в выходных системных очередях, определенных параметром «класс». Если параметр «класс» опущен, то уничтожаются все наборы данных, связанные с заданием;

идентификатор — идентификатор процедуры, используемый в соответствующей команде START;

типовое-имя — типовое имя периферийного устройства, зарезервированного с помощью команды MOUNT или выделенного программе системного вывода;

адрес-устройства — адрес периферийного устройства, зарезервированного с помощью команды MOUNT или выделенного программе системного вывода.

2.3.2. Команда DEFINE

Команда DEFINE позволяет изменить разделы основной памяти, выделяемые для выполнения программ, в операционной системе ОС ЕС, работающей в режиме MFT. Формат команды:

$$\left\{ \begin{array}{l} \text{DEFINE} \\ \text{N} \end{array} \right\} [\text{LIST}]$$

где LIST определяет вывод на консоль текущего состояния разделов основной памяти.

В ответ на команду DEFINE на консоль выдается сообщение "IEE802A ENTER DEFINITION", после чего оператор ОС ЕС ЭВМ может заново определить разделы основной памяти аналогично тому, как это делается при выполнении загрузки операционной системы (см. главу 3).

2.3.3. Команда HALT

Команда HALT вводится по завершении работы всех системных программ и заданий пользователей для регистрации состояния наборов данных системного журнала и занесения туда статистических данных о работе периферийных устройств. Формат команды:

$$\left\{ \begin{array}{l} \text{HALT} \\ \text{Z} \end{array} \right\} \text{EOD}$$

2.3.4. Команда HOLD

Команда HOLD позволяет задержать выполнение определенного задания или указанной системной очереди заданий. Формат команды:

$$\left\{ \begin{array}{l} \text{HOLD} \\ \text{H} \end{array} \right\} \left\{ \begin{array}{l} \text{задание} \\ \text{Q}^{\text{г-список}} \end{array} \right\}$$

где операнды определяют:

задание — имя задания, выполнение которого должно быть задержано. Если имя задания — Q, то его следует заключить в круглые скобки;

Q [=список] — задержка выполнения всех заданий, находящихся во входных системных очередях, определенных параметром «список». В качестве значения этого параметра можно указать до четырех классов системных очередей одновременно, например H Q=ABCD. Если параметр «список» опущен, то задерживается выполнение заданий во всех системных очередях.

2.3.5. Команда MODIFY

Команда MODIFY позволяет изменять параметры выполняемой системной программы или задания пользователя. Формат команды:

$$\left\{ \begin{array}{l} \text{MODIFY} \\ \text{F} \end{array} \right\} \left\{ \begin{array}{l} \text{задание} \\ \text{[процедура.] идентификатор} \end{array} \right\} \text{параметры}$$

где операнды определяют:

задание — имя задания, параметры которого следует изменить;
 процедура — имя процедуры, определенное в команде START;
 идентификатор — идентификатор процедуры;
 параметры — новые значения параметров системной программы или задания пользователя.

2.3.6. Команда RELEASE

Команда RELEASE отменяет команду HOLD, разрешая таким образом выполнение задержанных заданий. Формат команды:

$$\left\{ \begin{array}{l} \text{RELEASE} \\ \text{A} \end{array} \right\} \left\{ \begin{array}{l} \text{задание} \\ \text{Q[=список]} \end{array} \right\}$$

где операнды определяют:

задание — имя задания, выполнение которого разрешается; (см. 2.3.4);

Q[=список] — входные системные очереди, для которых отменяется команда HOLD (см. 2.3.4).

2.3.7. Команда RESET

Команда RESET позволяет изменить приоритеты и классы заданий, находящихся в системных очередях. Формат команды:

$$\left\{ \begin{array}{l} \text{RESET} \\ \text{E} \end{array} \right\} \text{задание [PARTY=n] [CLASS=k] [OUT=k]}$$

где операнды определяют:

задание — имя задания, параметры которого изменяются;
 PARTY=n — новый приоритет задания, где n может принимать значения, выражаемые двузначным числом от 00 до 13;

CLASS=k — класс системной очереди, в которую следует поместить задание. Буквы от A до O определяют классы входных

системных очередей, а буквы от А до Z и цифры от 0 до 9 — выходных;

OUT=k — класс выходной системной очереди, в которую следует поместить задание.

2.3.8. Команда SET

Команда SET завершает подготовку операционной системы к работе в процессе выполнения процедуры начальной загрузки. С помощью этой команды устанавливается дата и время суток, адреса наборов данных SYS1.SYSJOBQE и SYS1.PROCLIB, а также отменяется автоматический запуск системных программ. Формат команды:

{SET } DATE=гг.ддд[,CLOCK=чч.мм.сс] [,Q=([адрес-устройства] [,F])] [T] [,PROC=адрес-устройства] [,AUTO=xxx]

где операнды определяют:

DATE=гг.ддд — дату, где гг — две последние цифры года, ддд — трехзначное число от 001 до 366, обозначающее порядковый день года;

CLOCK=чч.мм.сс — время суток, где чч.мм.сс — двузначные числа, обозначающие часы (от 00 до 23), минуты и секунды (от 00 до 59);

Q=([адрес-устройства],[F]) — адрес устройства прямого доступа, на котором установлен том с набором данных SYS1.SYSJOBQE и указание ([,F]), следует ли этот набор данных заново размечать;

PROC=адрес-устройства — адрес устройства прямого доступа, на котором установлен том с набором данных SYS1.PROCLIB;

AUTO=xxx — подтверждает или отменяет автоматический запуск системных программ, определенных при генерации операционной системы; xxx — набор символов Y и N, где Y соответствует разрешению на автоматический запуск, а N — отмене. Если отменяется автоматический запуск всех системных программ, то в качестве параметра AUTO можно указать NONE (AUTO=NONE).

2.3.9. Команда START

Команда START инициирует выполнение процедуры, хранящейся в библиотеке SYS1.PROCLIB. Формат команды:

{START }процедура[.идентификатор], параметры
S

где операнды определяют:

процедура — имя процедуры;

идентификатор — в режиме MFT номер зоны выполнения в форме Pxx, где xx — номер зоны от 0 до 15. Для режимов MVT и SVS в качестве значения этого параметра можно указывать любую буквенно-цифровую последовательность от 1-го до 8-и символов;

параметры — любые значения переменных параметров, определенных в процедуре.

2.3.10. Команда STOP

Команда STOP останавливает выполнение системной программы или задания пользователя. Формат команды:

$$\left\{ \begin{array}{l} \text{STOP} \\ P \end{array} \right\} \left\{ \begin{array}{l} \text{процедура [идентификатор]} \\ \text{задание} \end{array} \right\}$$

где операнды определяют:

процедура — имя процедуры, указанное в команде START;
идентификатор — идентификатор, указанный в команде START;
задание — имя задания.

2.3.11. Команда WRITELOG

Команда WRITELOG позволяет распечатать системный журнал через выходную системную очередь заданного класса или отменить его ведение. Формат команды:

$$\left\{ \begin{array}{l} \text{WRITELOG} \\ W \end{array} \right\} \left\{ \begin{array}{l} \text{класс} \\ \text{CLOSE} \end{array} \right\}$$

где операнды определяют:

класс — имя класса выходной системной очереди, которую следует использовать для распечатки системного журнала;

CLOSE — прекращение ведения системного журнала до следующей загрузки операционной системы.

2.3.12. Команда SWITCH

Команда SWITCH позволяет заменить набор данных для записи информации, которую формирует системная мониторинговая программа (СМП). После выдачи этой команды СМП продолжает запись в пустой набор данных на томе прямого доступа (SYS1.MANX или SYS1.MANY), или на новый том, если для записи информации определена магнитная лента. Формат команды:

$$\left\{ \begin{array}{l} \text{SWITCH} \\ I \end{array} \right\} \text{ SMF}$$

2.4. Команды управления периферийными устройствами

Периферийные устройства распределяются заданию операционной системой во время запуска на основе информации, содержащейся в операторах DD. Оператор ОС ЕС ЭВМ, используя команды управления периферийными устройствами, может заранее подготовить устройства для выполнения задания или изменить адреса уже выделенных заданию устройств, если это необходимо.

В эту группу входят команды MOUNT, UNLOAD, VARY и SWAP.

2.4.1. Команда MOUNT

Команда MOUNT позволяет зарезервировать периферийное устройство для требуемого тома и задать параметры, определяющие его использование другими заданиями. До того как операционная система потребует установить требуемый том, команду MOUNT отменяет команда CANCEL (см. п. 2.3.1). После запроса операционной системы установить указанный в команде MOUNT том освободить зарезервированное устройство можно только с помощью команды UNLOAD. Формат команды:

$$\left\{ \begin{array}{l} \text{MOUNT} \\ \text{M} \end{array} \right\} \left\{ \begin{array}{l} \text{адрес-устройства} \\ \text{имя-устройства} \end{array} \right\} \left\{ \begin{array}{l} \text{,VOL=(NL, том)} \\ \text{,VOL=(SL, том)} \end{array} \right\} \\ \left[\text{,USE} = \left\{ \begin{array}{l} \text{STORAGE} \\ \text{PUBLIC} \\ \text{PRIVATE} \end{array} \right\} \right]$$

где операнды определяют:

адрес-устройства — адрес резервируемого устройства. В режиме MFT это должно быть устройство, предварительно разгруженное с помощью команды UNLOAD;

имя-устройства — резервирование одного из периферийных устройств указанного типа;

VOL=(NL, том) — резервирование накопителя на магнитной ленте для тома без стандартной метки;

том — регистрационный номер тома;

VOL=(SL, том) — резервирование периферийного устройства для тома, имеющего стандартную метку и заданный регистрационный номер;

USE — резервирование устройства прямого доступа для распределения в зависимости от значения параметра:

STORAGE — том для всех типов наборов данных;

PUBLIC — том для временных наборов данных и тех, для которых в операторах DD указан регистрационный номер тома;

PRIVATE — том только для тех наборов данных, для которых в операторах DD указан регистрационный номер тома.

2.4.2. Команда UNLOAD

Команда UNLOAD запрещает распределение указанного в команде периферийного устройства вновь запускаемым заданиям, а после завершения заданий, использующих это устройство, вызывает вывод на консоль сообщения IEF282I об освобождении устройства. После появления этого сообщения том может быть снят с данного периферийного устройства. Формат команды:

$$\left\{ \begin{array}{l} \text{UNLOAD} \\ \text{U} \end{array} \right\} \text{адрес-устройства}$$

где адрес-устройства — адрес разгружаемого периферийного устройства.

2.4.3. Команда VARY

Команда VARY в приведенном ниже формате предназначена для перевода заданных в команде устройств в оперативное или автономное состояние. Формат команды:

$$\left\{ \begin{array}{l} \text{VARY} \\ \text{V} \end{array} \right\} (\text{устройство 1} [\text{устройство2}] \dots) \left\{ \begin{array}{l} \text{,ONLINE} \\ \text{,OFFLINE} \\ \text{,PATH, устройство,} \end{array} \right\} \left\{ \begin{array}{l} \text{ONLINE} \\ \text{OFFLINE} \end{array} \right\}$$

где операнды определяют:

устройство 1 [устройство 2]...—адреса периферийных устройств, состояние которых устанавливается следующим операндом команды VARY;

ONLINE — перевод в оперативное состояние периферийных устройств, определенных первым операндом;

OFFLINE — перевод в автономное состояние устройств, определенных первым операндом;

PATH — применение команды VARY для изменения состояния пути к периферийному устройству, основной путь к которому определен первым операндом;

устройство — адрес периферийного устройства на том пути, состояние которого необходимо изменить;

ONLINE — разрешает использовать путь, определенный предыдущим операндом;

OFFLINE — запрещает использовать путь, определенный предыдущим операндом.

2.4.4. Команда SWAP

Команда SWAP предназначена для управления динамической реконфигурацией периферийных устройств. Формат команды:

$$\left\{ \begin{array}{l} \text{SWAP} \\ \text{G} \end{array} \right\} \left\{ \begin{array}{l} \text{OFF} \\ \text{ON} \end{array} \right\} \left\{ \begin{array}{l} \text{устройство 1, устройство 2} \end{array} \right\}$$

где операнды определяют:

OFF — перевод в неактивное состояние средств динамической реконфигурации периферийных устройств;

ON — перевод в активное состояние средств динамической реконфигурации периферийных устройств;

устройство 1 — адрес периферийного устройства (по основному пути), которое необходимо перевести в автономное состояние;

устройство 2 — новый адрес периферийного устройства (по основному пути).

2.5. Команды управления консолями

Консоли операционной системы ОС ЕС ЭВМ с мультиконсольным обеспечением отличаются друг от друга составом команд, которые можно с них вводить, составом сообщений, которые опе-

рационная система может на них посылать, и адресами альтернативных консолей, выполняющих роль резервных. Эти характеристики консолей, а также основная консоль определяются при генерации операционной системы.

Команды управления консолями предоставляют возможность изменять перечисленные характеристики консолей. В эту группу команд входят команды VARY, MSGRT и CONTROL с операндом L. Команды MSGRT и CONTROL описаны в 2.2. Периферийные устройства, определенные при генерации операционной системы в качестве консолей, могут находиться в одном из трех состояний: автономном, оперативном и в состоянии активной консоли.

2.5.1. Команда VARY CONSOLE

Команда VARY в приведенном ниже формате позволяет изменять состояние и характеристики всех консолей, исключая основную консоль и консоль, используемую в качестве устройства сбора протокола. Формат команды:

$$\left\{ \begin{array}{l} \text{VARY} \\ \text{V} \end{array} \right\} (\text{адрес 1} [\text{адрес 2}] \dots) \left\{ \begin{array}{l} \text{ONLINE} \\ \text{,OFFLINE} \\ \text{,CONSOLE} [\text{AUTH}=\text{x}] [\text{,ROUT}=\text{y}] \\ \quad [\text{ALTCONS}=\text{адрес}] \end{array} \right\}$$

где операнды определяют:

адрес1,[адрес2] — адреса периферийных устройств, используемых в качестве консолей, состояние или характеристики которых необходимо изменить. Если в качестве консоли используется устройство, предназначенное только для вывода информации, то этот операнд записывается в форме О-адрес, где «адрес» — адрес этого устройства. Для указания адресов устройств, образующих составную консоль, этот операнд записывается в форме (I-адрес, О-адрес) и обязательно заключается в круглые скобки. В этом случае I-адрес определяет адрес устройства ввода, а О-адрес — адрес устройства вывода;

ONLINE — отмену использования в качестве консолей периферийных устройств, определенных первым операндом, и перевод их в оперативное состояние;

OFFLINE — отмену использования в качестве консолей и перевод в автономное состояние периферийных устройств, определенных первым операндом;

CONSOLE — перевод в состояние активной консоли периферийного устройства, определенного первым операндом;

AUTH=x — группы команд оператора дисплея, которые можно выдавать с указанной консоли; x — принимает следующие значения:

ALL — возможность выдавать с указанной консоли все команды, кроме команд основной консоли,

INFO — информационные команды,

SYS — команды управления операционной системой,

IO — команды управления периферийными устройствами,

CONS — команды управления консолями;

ROUT=y — маршрутные коды сообщений, которые должны выводиться на заданную консоль; y принимает следующие значения:

ALL — все маршрутные коды от 1 до 16,

NONE — отмену ранее определенных маршрутных кодов,

(число [число]...) — список присваиваемых консоли маршрутных кодов. Каждый подпараметр может принимать значения от 1 до 16. Если указывается один подпараметр, то скобки могут быть опущены;

ALTCONS=адрес — адрес новой альтернативной консоли со средствами ввода-вывода.

2.5.2. Команды основной консоли

С основной консоли можно вводить любые команды оператора ОС ЕС ЭВМ. Переопределение основной консоли или устройства сборного протокола, получение сведений о режиме аппаратного контроля ЭВМ и его изменении, а также получение дампа основной памяти можно выполнить только с основной консоли посредством команд VARY MSTCONS, VARY HARDCPY, MODE и DUMP, называемых командами основной консоли.

Команда VARY MSTCONS позволяет назначить новую основную консоль. Формат команды:

$$\left\{ \begin{matrix} \text{VARY} \\ \text{V} \end{matrix} \right\} \text{адрес, MSTCONS}$$

где операнды определяют:

адрес — адрес новой основной консоли. Если в качестве основной консоли определяется составная консоль, то этот операнд записывается в форме (I-адрес, O-адрес), где I-адрес определяет адрес устройства ввода; а O-адрес определяет адрес устройства вывода;

MSTCONS — применение команды VARY для назначения новой основной консоли.

Назначение новой основной консоли произойдет только в том случае, если консоль, определяемая параметром «адрес», активна и не является только устройством вывода.

Если операционная система обнаруживает, что основная консоль неработоспособна, то она автоматически пытается назначить основной ее альтернативную консоль. Если альтернативная консоль также неработоспособна, то на все активные консоли выводится сообщение:

IEE141A MASTER AND ALL ALTERNATES UNAVAILABLE-
USSUE VARY MSTCONS.

В этом случае команду VARY MSTCONS можно ввести с любой активной консоли.

Команда **VARY HARDCPY** позволяет переопределить устройство и параметры сборного протокола во время работы операционной системы. Формат команды:

$$\left\{ \begin{matrix} \text{VARY} \\ \text{V} \end{matrix} \right\} \left[\begin{matrix} \text{адрес} \\ \text{SYSLOG} \end{matrix} \right] , \text{HARDCPY} [.,x] [.,\text{ROUT}=y]$$

где операнды определяют:

адрес — адрес нового устройства сборного протокола. Это должен быть адрес консоли, не являющейся дисплей-консолью, или адрес устройства вывода составной консоли. В обоих случаях консоль должна быть активной;

SYSLOG — системный журнал в качестве устройства сборного протокола. Если системный журнал не был определен при генерации операционной системы, то команда отвергается;

HARDCPY — применение команды VARY;

x — множество команд оператора ОС ЕС ЭВМ и сообщений операционной системы, которые должны заноситься в сборный протокол; x принимает следующие значения:

CMDS — занесение в сборный протокол всех команд оператора ОС ЕС ЭВМ и ответов на них,

NOCMDS — запрещает занесение в сборный протокол всех команд оператора ОС ЕС ЭВМ и ответов на них,

INCMDS — занесение в сборный протокол всех команд оператора ОС ЕС ЭВМ и ответов на них, кроме сообщений в ответ на команды DISPLAY и MONITOR,

STCMDS — занесение в сборный протокол всех команд оператора ОС ЕС ЭВМ и ответов на них, кроме сообщений в ответ на команду MONITOR,

OFF — отменяет ведение системного журнала;

ROUT=y — сообщения, которые должны заноситься в сборный протокол; y принимает следующие значения:

ALL — занесение в сборный протокол всех сообщений,

NONE — запрещает занесение в сборный протокол любых сообщений,

(число, число ...) — список маршрутных кодов сообщений, которые должны заноситься в сборный протокол. Каждый подпараметр может принимать значения от 1 до 16.

Если первый операнд опущен, т. е. вместо него указана запятая, то данная команда изменяет свойства ранее определенного сборного протокола. Если же он не был определен, то команда отвергается.

Команда MODE позволяет получить сведения о режиме работы аппаратных средств контроля и изменить его, если необходимо. Формат команды:

$$\text{MODE} \left\{ \begin{array}{l} \text{STATUS} \\ \text{RETRY } x \\ \text{MAIN } y \\ \text{CONTROL } z \end{array} \right\}$$

где операнды определяют:

STATUS — вывод сообщений о режиме функционирования аппаратных средств контроля;

RETRY x, где *x* может принимать значения **RECORD** или **QUIET**:

RETRY RECORD — переключение средств аппаратного повторения команд в режим оповещения;

RETRY QUIET — переключение средств аппаратного повторения команд в режим без оповещения;

MAIN y, где *y* может принимать значения **RECORD**, **QUIET** или **COUNT n**:

MAIN RECORD — переключение средств контроля ошибок основной памяти в режим оповещения. Переключение произойдет только в том случае, если предварительно будет выдана команда **MODE** с операндом **RETRY RECORD**;

MAIN QUIET — определяет переключение средств контроля ошибок основной памяти в режим без оповещения;

MAIN COUNT n — определяет переключение средств контроля ошибок основной памяти ЭВМ EC-1060 в режим счета с порогом *n*. Однако переключение в этот режим произойдет только в том случае, если предварительно будет выдана команда **MAIN RECORD**. Для ЭВМ EC-1055 параметр *n* определяет порог для числа легких ошибок, где *n* принимает значение от 1 до 999;

CONTROL z, где *z* может принимать значения **THRESHOLD** или **QUIET**:

CONTROL THRESHOLD — вызывает переключение средств контроля ошибок памяти микропрограмм в режим, вызывающий прерывание от схем контроля после нескольких первых ошибок. При достижении некоторого порогового числа ошибок аппаратные средства контроля и коррекции ошибок памяти микропрограмм автоматически переключаются в режим без оповещения;

CONTROL QUIET — переключение средств контроля ошибок памяти микропрограмм в режим без оповещения. Операнд **CONTROL** можно применять только для ЭВМ EC-1035.

Команда DUMP позволяет получить дампы основной памяти и поместить его в набор данных **SYS1-DUMP**. Формат команды:

DUMP COMM=(текст)

COMM=(текст) — позволяет поместить в качестве первой записи в наборе данных для дампа текст размером от 1 до 100 символов.

В ответ на сообщение, указывающее на готовность операционной системы выдать дамп, оператор ОС ЕС ЭВМ должен ввести команду REPLY следующего формата:

$$\left\{ \begin{array}{c} \text{REPLY} \\ R \end{array} \right\} \text{идентификатор} \cdot \left\{ \begin{array}{c} \text{ALL} \\ \text{STOR} = (\text{адрес } 1, \text{ адрес } 2, \dots) [\text{SDATA}] \\ \text{SDATA} \end{array} \right\}$$

где операнды определяют:

идентификатор — идентификатор ответа;

ALL — получение дампа всей основной памяти, включая ядро операционной системы и область системных очередей SQA;

STOR = (адрес1, адрес2, ...) — граничные адреса участков основной памяти, причем параметр адрес1 определяет адрес первого байта, а параметр адрес2 — адрес последнего байта участка. Адреса указываются либо шестизначными шестнадцатеричными числами, либо пятизначными десятичными, за которыми должна следовать буква К. В операнде можно определить не более двух участков;

SDATA — получение дампа ядра операционной системы и области системных очередей SQA.

2.6. Управление программой системного ввода (RDR)

Управление программой системного ввода (RDR) осуществляется с помощью команд START, STOP и CANCEL.

2.6.1. Команда START (RDR)

Команда START позволяет запустить программу системного ввода. Формат команды:

$$\left\{ \begin{array}{c} \text{START} \\ S \end{array} \right\} \begin{array}{l} \text{процедура} [\text{идентификатор}] [\text{устройство}] [\text{том}] \\ [\text{задание}] [\text{DD-параметры}] \end{array}$$

где операнды определяют:

процедура — имя процедуры. В библиотеке ОС ЕС ЭВМ с именем SYS1.PROCLIB хранятся три процедуры для запуска программы системного ввода: RDR, RDR400 и RDR3200. Последние две процедуры обеспечивают блокирование записей по 400 и 3200 байт соответственно в создаваемых программой системного ввода наборах данных задания;

идентификатор — в режиме MFT номер зоны выполнения в форме Pxx, где xx — номер зоны от 0 до 15. Указание в качестве идентификатора S позволяет запустить программу системного ввода в первой по порядку зоне размером не менее 48 Кбайт, которая в момент выдачи команды окажется свободной. Использование этого идентификатора позволяет заданиям, для выполнения которых требуется зона выполнения, занятая программой системного ввода, вытеснять ее в свободную зону. Для режимов MVT и SVS

в качестве идентификатора можно задать любую буквенно-цифровую последовательность от 1 до 8 символов;

устройство — адрес или типовое имя устройства, на котором находится входной поток заданий;

том — регистрационный номер тома, на котором находится входной поток заданий;

задание — имя задания во входном потоке, с которого следует начать ввод. Последние три операнда следует располагать строго в указанном порядке, и пропуск любого промежуточного операнда отмечать запятой.

DD-параметры — операнды оператора DD, описывающего набор данных, содержащий входной поток заданий.

Запрещается в одной команде START задавать пары операндов «том» и VOLUME, а также «устройство» и UNIT.

2.6.2. Команда STOP (RDR)

Команда STOP позволяет остановить ввод входного потока заданий программой системного ввода и завершить ее работу. Формат команды описан в 2.3.10. Фактически ввод заданий прекращается и программа системного ввода завершается после того, как будет введено задание, которое обрабатывалось в момент выдачи команды STOP.

2.6.3. Команда CANCEL (RDR)

Команда CANCEL позволяет отменить команду START до того, как закончится распределение периферийных устройств программе системного ввода. Формат команды описан в 2.3.1.

2.7. Управление инициатором выполнения заданий (INIT)

Управление инициатором осуществляется с помощью команд START, MODIFY и STOP.

2.7.1. Команда START (INIT)

Команда START позволяет запустить инициатор. Формат команды:

$\left\{ \begin{array}{c} \text{START} \\ S \end{array} \right\}$ процедура [идентификатор] [.,.,классы]

процедура — имя процедуры. В библиотеке ОС ЕС ЭВМ с именем SYS1. PROC LIB хранится одна стандартная процедура для запуска инициатора — INIT;

идентификатор — идентификатор процедуры (см. 2.3.9);

классы — классы и приоритеты заданий, предназначенных для обработки данным инициатором. Этот операнд имеет следующий формат:

(к[(n)], ... [LIMIT=значение, ...])

k — буква, определяющая класс заданий. Можно указать до восьми классов заданий, причем порядок расположения букв определяет порядок выборки заданий для выполнения. Следующие два примера иллюстрируют возможности задания классов заданий без определения других значений операнда: S INIT. A,,, (A,B), S INIT. A,,, AB;

n — значение приоритета (от 0 до 15), приписываемое всем заданиям класса k;

LIMIT = значение — предельный приоритет (наибольшее значение приоритета, которое может быть приписано заданиям, обрабатываемым данным инициатором).

2.7.2. Команда MODIFY (INIT)

Команда MODIFY позволяет изменить классы и приоритеты заданий, приписанные данному инициатору с помощью команды START. Формат команды:

$\left\{ \begin{matrix} \text{MODIFY} \\ \text{F} \end{matrix} \right\}$ [процедура.] идентификатор, CLASS = классы

где операнды определяют:

процедура. — имя процедуры, указанное в команде START;

идентификатор — идентификатор процедуры, указанный в команде START;

CLASS = классы — новые значения классов и приоритетов заданий, установленных с помощью операнда «классы» команды START.

2.7.3. Команда STOP (INIT)

Команда STOP позволяет остановить работу инициатора после того, как будет завершено задание, выполнение которого было иницировано с его помощью. Формат команды описан в 2.3.10.

2.8. Управление программой системного вывода (WTR)

Основной функцией программы системного вывода (WTR) является поиск наборов данных, определенных в операторах DD задания с помощью операнда SYSOUT, и вывод их на заданное при ее запуске периферийное устройство. Управление программой системного вывода осуществляется с помощью команд START, MODIFY, CANCEL и STOP.

2.8.1. Команда START (WTR)

Команда START позволяет запустить программу системного вывода. Формат команды:

$\left\{ \begin{matrix} \text{START} \\ \text{S} \end{matrix} \right\}$ процедура [идентификатор] [устройство] [том]
[класс] [DD-параметры]

где операнды определяют:

процедура — имя процедуры. В библиотеке ОС ЕС ЭВМ с именем SYS1.PROCLIB хранится одна процедура для запуска программы системного вывода — WTR;

идентификатор — идентификатор процедуры (см. 2.3.9).

устройство — определяет адрес или типовое имя периферийного устройства для вывода данных;

том — регистрационный номер тома для вывода данных;

класс — класс выходной системной очереди (любая из букв от A до Z или цифр от 0 до 9). С помощью этого операнда устанавливается соответствие между набором данных, определенным оператором DD с операндом SYSOUT, и периферийным устройством, которое задано операндом «устройство».

В команде START можно задать до восьми классов выходных системных очередей, причем порядок их обработки будет определяться порядком расположения символов, определяющих классы.

Следующие два примера иллюстрируют правила задания классов:

S WTR.A,,, ABC S WTR.A,,, (ABC).

Последние три операнда команды START следует располагать строго в указанном порядке и пропуск любого промежуточного операнда отмечать запятой.

DD-параметры — операнды оператора DD, определяющего периферийное устройство для вывода данных.

Запрещается в одной команде START задавать пары операндов «том» и VOLUME, а также «устройство» и UNIT.

2.8.2. Команда MODIFY (WTR)

Команда MODIFY позволяет изменять классы выходных системных очередей, определенные в команде START, и приостанавливать выполнение программы системного вывода. Формат команды:

{MODIFY}{F}[процедура.] идентификатор [CLASS=класс] [PAUSE=x]

где операнды определяют:

процедура — имя процедуры, указанное в команде START;

идентификатор — идентификатор процедуры, указанный в команде START;

CLASS=класс — от 1 до 8 классов выходных системных очередей, которыми следует заменить классы, ранее определенные в команде START;

PAUSE=x — паузу в работе программы системного вывода, причем x принимает значения FORMS и DATASET;

PAUSE=FORMS — паузу в работе программы системного вывода и вывод на консоль сообщения IEF382A,

PAUSE=DATASET — паузу в работе программы системного вывода в начале обработки следующего набора данных из выходной системной очереди и вывод на консоль сообщения IEF383A.

2.8.3. Команда CANCEL (WTR)

Команда CANCEL позволяет отменить команду START до того, как закончится распределение периферийных устройств программе системного вывода. В противном случае с помощью команды CANCEL отменяется вывод обрабатываемого набора данных и осуществляется переход к обработке следующего набора данных из выходной системной очереди. Команда CANCEL употребляется в этом случае только с операндом «идентификатор» (см. 2.3.1), в качестве значения которого можно указать идентификатор процедуры, адрес или типовое имя устройства, заданного в команде START.

Для отмены вывода обрабатываемого набора данных следует указывать только одно из двух последних значений операнда.

2.8.4. Команда STOP (WTR)

Команда STOP позволяет остановить работу программы системного вывода после того, как будет выведен текущий обрабатываемый набор данных.

Команда STOP употребляется в этом случае с операндом [процедура.] идентификатор (см. 2.3.10), где в качестве значений последнего параметра можно указать идентификатор процедуры, адрес или типовое имя устройства, заданного в команде START.

2.9. Управление программой прямого системного вывода (DSO)

Программа прямого системного вывода (DSO) обеспечивает непосредственный вывод на периферийное устройство, минуя промежуточные наборы данных выходной системной очереди. Управление работой программы DSO осуществляется с помощью команд START, MODIFY и STOP.

2.9.1. Команда START (DSO)

Команда START позволяет запустить программу DSO. Формат команды:

$$\left\{ \begin{array}{l} \text{START} \\ \text{S} \end{array} \right\} \begin{array}{l} \text{процедура} \text{ [идентификатор]} \\ \text{[классы]} \end{array} \begin{array}{l} \text{[устройство]} \\ \text{[DD=параметры]} \end{array} \begin{array}{l} \text{[том]} \end{array}$$

где операнды определяют:

процедура — имя процедуры. В библиотеке ОС ЕС ЭВМ с именем SYS1.PROCLIB хранится одна процедура для запуска программы прямого системного вывода — DSO;

идентификатор — идентификатор процедуры (см. 2.3.9);

устройство — адрес или типовое имя периферийного устройства, предназначенного для вывода данных. Допустимыми устройст-

вами являются печатающее устройство и накопитель на магнитной ленте;

том — регистрационный номер тома магнитной ленты, предназначенного для вывода данных;

классы — классы заданий и выходных данных. Классы заданий, вывод данных которых должен выполняться с помощью программы DSO, указывается с помощью параметра `JOBCLASS=x`, где `x` — последовательность от 1 до 8 букв, являющихся именами классов. Классы выходных данных указываются с помощью параметра `OUTCLASS=y`, где `y` — одна буква, соответствующая значению параметра в операнде `SYSOUT` оператора `DD`, определяющего выходной набор данных. В качестве значения операнда «классы» могут быть указаны один или оба параметра. Если указываются оба параметра, то они должны быть разделены запятой и заключены в скобки;

DD-параметры — операнды оператора `DD`, определяющего периферийное устройство для вывода данных. Запрещается в одной команде `START` задавать пары операндов «том» и `VOLUME`, а также «устройство» и `UNIT`.

2.9.2. Команда **MODIFY (DSO)**

Команда `MODIFY` позволяет изменять параметры, заданные в команде `START`. Формат команды:

```
{ MODIFY      }  [процедура.] идентификатор [,JOBCLASS=класс-задания]
{ F            }  [,OUTCLASS=класс-вывода]
```

где операнды определяют:

процедура — имя процедуры, заданное в команде `START`;

идентификатор — идентификатор процедуры либо адрес, либо типовое имя устройства, заданного в команде `START`;

`JOBCLASS=класс-задания` — от 1 до 8 классов заданий, которые заменяют классы, заданные в команде `START`;

`OUTCLASS=класс-вывода` — один класс для выходного набора данных, который заменяет класс, заданный в команде `START`.

2.9.3. Команда **STOP (DSO)**

Команда `STOP` позволяет остановить программу DSO после того, как будет завершено задание, данные которого ею обрабатывались. Формат команды описан в 2.8.4.

2.10. Управление работой универсального средства трассировки

Универсальное средство трассировки (GTF) отслеживает и сохраняет в форме, удобной для анализа, информацию о прерываниях ввода-вывода, SVC-прерываниях, внешних и программных прерываниях, а также о переключении задач, привилегированных

командах и событиях, определенных пользователем. Управление универсальным средством трассировки осуществляется с помощью команд START и STOP.

2.10.1. Команда START (GTF)

Команда START позволяет запустить универсальное средство трассировки. Формат команды:

```
{START } {процедура} [идентификатор] [устройство] [том]  
{S      } {,MODE=x} [,BUF=размер] [,TIME=YES]  
           {,DEBUG=YES} [,REG=размер] [,DD-параметры]
```

где операнды определяют:

процедура — имя процедуры. В библиотеке ОС ЕС ЭВМ с именем SYS1.PROCLIB хранится одна процедура для запуска универсального средства трассировки — GTF;

идентификатор — идентификатор процедуры (см. 2.3.9);

устройство — адрес или типовое имя периферийного устройства, которое предполагается использовать для регистрации событий;

том — регистрационный номер тома, на котором предполагается хранить записи регистрации событий;

MODE=x — режим учета событий; если x принимает значение (INT), то записи о событиях не будут заноситься в набор данных SYS1.TRACE. Если x принимает значение (INT, S), то записи будут редактироваться средствами ABDUMP. Если параметр MODE опущен или равен EXT, то записи о событиях будут заноситься в набор данных SYS1.TRACE.

BUF=размер — размер буфера для записи регистрации события;

TIME=YES — занесение в каждую запись регистрации события времени суток, когда оно произошло;

DEBUG=YES — поведение универсального средства трассировки в аварийной ситуации. Если этот параметр задан, то универсальное средство трассировки прекращает работу при обнаружении условия, которое оно не в состоянии обработать. Если параметр DEBUG=YES опущен, то функционирование универсального средства трассировки будет продолжаться с возможными отклонениями от заданного режима работы;

REG=размер — размер зоны основной памяти, занимаемой универсальным средством трассировки в количестве 1024-байтных блоков. Если этот параметр опущен, то выделяется область размером в 26 блоков.

DD-параметры — операнды оператора DD, определяющего набор данных SYS1.TRACE.

2.10.2. Команда STOP (GTF)

Команда STOP позволяет остановить работу универсального средства трассировки. Формат команды описан в 2.8.4.

2.11. Управление работой системы ДУВЗ (CRJE)

Система ДУВЗ выполняется под управлением операционной системы ОС ЕС ЭВМ подобно другим системным программам. Управление работой системы ДУВЗ осуществляется с помощью команд START, MODIFY, STOP, CENOUT, BRDCST, MSG, SHOW, USERID.

2.11.1. Команда START (CRJE)

Команда START позволяет осуществить запуск системы ДУВЗ. Формат команды:

$$\left\{ \begin{matrix} \text{START} \\ S \end{matrix} \right\} \text{ процедура } [\text{идентификатор}], \dots \left(\left\{ \begin{matrix} \text{FORM} \\ \text{NFMT} \\ \text{NONE} \end{matrix} \right\} \right) \left\{ \begin{matrix} \text{ABNO} \\ \text{NORM} \end{matrix} \right\}$$

где операнды определяют:

процедура — имя процедуры. В библиотеке ОС ЕС ЭВМ с именем SYS1.PROCLIB хранится одна процедура для запуска системы ДУВЗ — CRJE;

идентификатор — идентификатор процедуры. Для режимов MVT и SVC указывать идентификатор необязательно. Для режима MFT должно быть указано P0. При этом зона P0 должна быть зоной резидентной программы системного ввода;

FORM — запуск системы ДУВЗ после перезагрузки операционной системы с очисткой набора данных SYS1.SYSJOBQE (в операнде Q команды SET было указано F);

NFMT — запуск системы ДУВЗ после перезагрузки операционной системы без очистки набора данных SYS1.SYSJOBQE;

NONE — запуск системы ДУВЗ, если операционная система не перезагружалась после последнего останова системы ДУВЗ;

ABNO — выполнение системой ДУВЗ процедуры восстановления своей зоны выполнения;

NORM — выполнение системой ДУВЗ инициализации своей зоны выполнения.

2.11.2. Команда MODIFY (CRJE)

Команда MODIFY позволяет перевести канал связи системы ДУВЗ в оперативное или автономное состояние. Формат команды:

$$\left\{ \begin{matrix} \text{MODIFY} \\ F \end{matrix} \right\} [\text{процедура.}] \text{ идентификатор, } \left\{ \begin{matrix} D \\ A \end{matrix} \right\} = (\text{канал 1, } \dots)$$

где операнды определяют:

процедура — имя процедуры, указанное в команде START;

идентификатор — идентификатор процедуры, указанный в команде START;

D = (канал1, ...) — адреса каналов связи, которые необходимо перевести в автономное состояние;

A = (канал1, ...) — адреса каналов связи, которые необходимо перевести в оперативное состояние.

2.11.3. Команда CENOUT

Команда CENOUT позволяет вывести данные завершеного задания системы ДУВЗ из ее выходного класса с помощью программы системного вывода. Формат команды:

CENOUT J=задание, C=класс

J=задание — имя задания, выходные данные которого необходимо вывести на устройство программы системного вывода;

C=класс — класс выходной системной очереди, через которую должен быть осуществлен вывод данных задания.

2.11.4. Команда SHOW

Команда SHOW позволяет вывести на консоль информацию о текущем состоянии системы ДУВЗ. Формат команды:

SHOW x

где x может принимать одно из следующих значений:

JOBS [задание]; USERS [идентификатор]; ACTIVE [NUMBER];
BRDCST; MSGS [идентификатор]; LERB [адрес]; SESS[идентификатор];
SESSREL [идентификатор],

которые определяют:

JOBS — вывод на консоль информации о выполняемых заданиях системы ДУВЗ;

задание — имя задания, о состоянии которого необходимо получить информацию;

USERS — вывод на консоль информации об абонентах системы ДУВЗ;

идентификатор — идентификатор абонента системы ДУВЗ, о котором необходимо получить информацию;

ACTIVE — вывод на консоль информации о всех активных абонентах системы ДУВЗ, т. е. установивших с нею связь с помощью команды LOGON;

NUMBER — вывод на консоль информации о количестве активных абонентов системы ДУВЗ;

BRDCST — вывод на консоль всех сообщений, хранящихся в разделе BRDCST библиотеки CRJE.SYSLIB;

MSGS — вывод на консоль всех сообщений, отправленных ранее с помощью MSG и не принятых абонентами системы ДУВЗ;

идентификатор — идентификатор абонента системы ДУВЗ, все сообщения к которому необходимо вывести на консоль;

LERB — вывод на консоль информации об ошибках, зарегистрированных в каналах связи, и количества выполненных передач данных между абонентами системы ДУВЗ и ЭВМ;

адрес — адрес канала связи, информацию о работе которого необходимо вывести на консоль оператора ЭВМ;

SESS — вывод на консоль информации о всех абонентах системы ДУВЗ в момент выполнения ими команд LOGON и LOGOFF;

идентификатор — идентификатор абонента системы ДУВЗ, информацию о выполнении которым команд LOGON и LOGOFF необходимо получать на консоли;

SESSRELL — отмену команды SHOW SESS;

идентификатор — идентификатор абонента системы ДУВЗ, для которого необходимо отменить действие команды SHOW SESS.

В операционной системе ОС ЕС ЭВМ с мультиконсольным обеспечением рекомендуется после ранее выданной команды SHOW SESS перед остановкой системы ДУВЗ выдать команду SHOW SESSREL.

2.11.5. Команда USERID

Команда USERID позволяет включать абонентов в список допустимых абонентов системы ДУВЗ и исключать их из него. Формат команды:

USERID x

где x может принимать одно из следующих значений:

A[DD]=(идентификатор, пароль), D[DELETE]=(идентификатор, пароль),
S[UPPRESS], R[ESUME]

которые определяют:

A[DD]=(идентификатор, пароль) — абонент для включения в список допустимых абонентов системы ДУВЗ;

идентификатор — идентификатор абонента (любая последовательность от 1 до 7 буквенно-цифровых символов, первый из которых буква);

пароль — пароль на право доступа к системе ДУВЗ (любая последовательность от 1 до 8 буквенно-цифровых символов, первый из которых буква);

D[DELETE]=(идентификатор, пароль) — абонент для исключения из списка допустимых абонентов системы ДУВЗ;

S[UPPRESS] — запрет неактивным абонентам включаться в работу системы ДУВЗ. Команды абонентов LOGON отвергаются;

R[ESUME] — отменяет запрет абонентам включаться в работу системы ДУВЗ.

2.11.6. Команда BRDCST

Команда BRDCST позволяет изменять содержимое раздела BRDCST в наборе данных CRJE.SYSLIB. В этом разделе хранятся сообщения абонентам системы ДУВЗ. Формат команды:

BRDCST x

где x может принимать одно из следующих значений: nnnn, 'текст', 'текст'; nnnn; DELETE, которые определяют:

nnnn, 'текст' — номер записи в разделе BRDCST, куда следует поместить текст, следующий за номером. Текст длиной до 40 символов должен быть заключен в апострофы. Апостроф в тексте указывается парой апострофов;

'текст' — текст сообщения, которое следует добавить в раздел BRDCST;

pppp — номер записи, которую следует удалить из раздела BRDCST;

DELETE — удаление всех записей из раздела BRDCST.

2.11.7. Команда MSG

Команда MSG позволяет осуществлять связь с абонентами системы ДУВЗ. Формат команды:

$$\text{MSG} \quad \left\{ \begin{array}{l} M = \text{'текст'} \quad [U = \text{идентификатор}] \quad [Q] \\ D = \text{идентификатор} \end{array} \right\}$$

где операнды определяют:

M = 'текст' — текст сообщения;

'текст' — текст сообщения (до 40 символов);

U = идентификатор [Q] — абонент, которому отправляется сообщение;

идентификатор — идентификатор абонента;

Q — занесение сообщения в набор данных CRJE.SYSLIB (раздел BRDCST), если абонент в момент выдачи команды MSG был неактивен. Сообщение будет передано абоненту после того, как он выдаст команду LOGON. Если параметр Q опущен и абонент в момент выдачи команды MSG был неактивен, то сообщение теряется. Если операнд U опущен, то сообщение отправляется всем активным абонентам системы ДУВЗ;

D = идентификатор абонента, все сообщения к которому, хранящиеся в наборе данных CRJE.SYSLIB (раздел BRDCST), следует уничтожить.

2.11.8. Команда STOP (CRJE)

Команда STOP позволяет остановить работу системы ДУВЗ. Формат команды описан в п. 2.3.10.

Процедура начальной загрузки

3.1. Общие сведения

Процедура начальной загрузки представляет собой последовательность операций, выполняемых оператором ОС ЕС ЭВМ и управляющей программой операционной системы в процессе подготовки ее к выполнению заданий пользователей.

Оператор ОС ЕС ЭВМ во время выполнения процедуры начальной загрузки использует пульт ЭВМ и основную консоль. С помощью пульта ЭВМ оператор ОС ЕС ЭВМ указывает адрес устройства прямого доступа, на котором установлен резидентный том операционной системы, запускает и определяет состояние выполнения программ начальной загрузки и инициализации ядра операционной системы.

Программа инициализации ядра выполняется в соответствии с параметрами, которые вводятся в качестве ответов на поступающие от нее сообщения через основную консоль. Кроме сообщений программы инициализации ядра, на основную консоль поступают сообщения главного планировщика и ряда системных программ, включаемых в операционную систему в процессе генерации, например сообщения системной мониторинной программы, универсальной программы трассировки, монитора динамической отладки и т. д. Сообщения имеют следующий формат:

идент CCCpppS текст сообщения [Ppp]

где идент — цифровой идентификатор ответа на сообщение, требующее ответа (отсутствует в сообщениях программы инициализации ядра);

CCC — код программного компонента операционной системы; например IEA — сообщения программы инициализации ядра и супервизора, IEE — сообщения главного планировщика, IEF — сообщения планировщика и т. д.;

ppp — номер сообщения внутри группы сообщений программного компонента;

S — тип сообщения:

A — сообщение требует от оператора ОС ЕС ЭВМ немедленного действия;

D — сообщение требует выполнить одно из предлагаемых действий;

- Е — сообщение требует от оператора ОС ЕС ЭВМ действия, выполнение которого может быть отложено на некоторое время;
- I — информационное сообщение, не требующее действий;
- W — информационное сообщение об обстановке работы операционной системы до тех пор, пока оператором ОС ЕС ЭВМ не будут предприняты соответствующие действия;
- [Pnn] — для режима MFT операционной системы ОС ЕС в этом поле указывается номер раздела основной памяти.

3.2. Использование пульта ЭВМ

Для выполнения программы начальной загрузки оператор ОС ЕС ЭВМ должен выполнить следующие действия:

набрать на пульте управления ЭВМ адрес устройства прямого доступа, на котором установлен резидентный том операционной системы;

нажать кнопку «ЗАГРУЗКА».

Причина перехода процессора ЭВМ в состояние ожидания до появления на основной консоли сообщений программы инициализации ядра устанавливается в результате анализа кода состояния ожидания, помещаемого в последние 12 бит слова состояния программы (PSW). Если причиной перехода в состояние ожидания является постоянная ошибка ввода-вывода, то в регистре 10 содержится адрес периферийного устройства, обращение к которому вызвало ошибку.

Возможные коды состояния ожидания перечислены в гл. 18.

3.3. Определение параметров ОС ЕС

После завершения программы начальной загрузки оператор ОС ЕС ЭВМ продолжает загрузку операционной системы, используя для этого основную консоль. Он анализирует сообщения, выдаваемые на основную консоль, и вводит ответы и команды, устанавливая таким образом необходимую рабочую конфигурацию ОС ЕС.

Ниже приводятся сообщения операционной системы и ответы оператора ОС ЕС ЭВМ, приводящие к конфигурации ОС ЕС режима MFT, заданной при генерации.

```
IEA218I      MOD=1020 ASSUMED
IEA101A      SPECIFY SYSTEM PARAMETERS FOR RELEASE 4.13 MFT
IEA103I      DATASET SYS1. DUMP NOT FOUND BY LOCATE
IEA135A      SPECIFY SYS1. DUMP TAPE UNIT ADDRESS OR NO
R 00, 'NO'
IEA208I      SYS1. DUMP FUNCTION INOPERATIVE
* 00 IEE801D CHANGE PARTITIONS—REPLY YES/NO (LIST)
R 00, 'NO'
```

```

IEE866I      DEFINE COMMAND BEING PROCESSED
IEE812I      P 03 HAS 165 888 EXCESS BYTES ADDED
IEE805I      DEFINITION COMPLETED
IEE101A      READY
IEE103I      S WTR.P0, 00E *
IEE103I      S RDR.S, 00C *
IEE103I      S INIT.ALL *
T DATE=79. 231, CLOCK=10. 10. 00, Q=(,F)
* 00 IEF423A SPECIFY JOB QUEUE PARAMETERS
R 00, 'U'

```

Перечисленные сообщения и ответы оператора ОС ЕС ЭВМ в общем случае имеют следующий формат:

IEA218I MOD=nnnn, ALTSYS=ddd ASSUMED

где nnnn — шифр модели ЕС ЭВМ;

ddd — альтернативный адрес устройства прямого доступа для размещения резидентного тома операционной системы.

IEA101A SPECIFY SYSTEM PARAMETERS FOR xx.yy.sss

где xx — номер издания ОС ЕС;

yy — номер модификации ОС ЕС;

sss — режим ОС ЕС (MFT, MVT или SVS).

Чтобы в качестве значений параметров остались заданные при генерации, в ответ на это сообщение следует на консоли нажать клавишу, используемую для ввода информации в ЭВМ, или набрать и ввести

R 00, 'U'

Для изменения параметров следует ввести команду

R 00, 'параметр1, параметр2, ... L'

где 'параметр1, параметр2' — новые значения параметров;

L — вывод на основную консоль списков, задаваемых параметрами. Чтобы отменить какой-либо параметр, заданный при генерации, следует указать только определяющее его ключевое слово и знак равенства. Для продолжения ответа на следующей строке после последнего параметра предыдущей строки через пробел необходимо указать CONT.

Параметры для ОС ЕС режима MFT:

ALTSYS=ddd — адрес устройства прямого доступа для установки резидентного тома в случае выхода из строя устройства, на котором он был установлен сначала;

BLDL=xx[.yy] — построение для библиотек SYS1.LINKLIB и SYS1.SVCLIB резидентных списков модулей с помощью макрокоманды BLDL. Цифры xx, yy добавляются к IEABLD для формирования имен разделов библиотеки SYS1.PARMLIB, содержащей списки модулей;

HARDCPY=(x[,y][,z]) — адрес устройства сборного протокола операционной системы ОС ЕС с мультиконсольным обеспечением.

Параметр x принимает следующие значения:

адрес — адрес устройства для сборного протокола. Это должна быть одна из консолей, не являющихся дисплеями, или устройство вывода составной консоли;

SYSLOG — системный журнал в качестве устройства сборного протокола.

Параметр **y** принимает следующие значения:

(число, число...) — определяет список маршрутных кодов сообщений, которые должны заноситься в сборный протокол;
ALL — определяет занесение в сборный протокол всех сообщений.

Если сборный протокол обязателен, то ему будут всегда приписаны маршрутные коды 1, 2, 3, 4, 7, 8 и 10.

Параметр **z** принимает значение **NOCMDS** и запрещает занесение в сборный протокол всех команд оператора ОС ЕС ЭВМ и ответов на них. Этот параметр игнорируется, если сборный протокол обязателен;

MIN=**пп** — минимальное количество основной памяти, необходимое для инициализации задания; **пп** — число блоков размером в 1 Кбайт;

MOD=**пп** — тип ЭВМ, на которой выполняется процедура начальной загрузки и вызывает включение в резидентную часть ОС ЕС программ восстановления, зависящих от типа ЭВМ. Если этот параметр опущен, то подразумевается тип ЭВМ, заданный при генерации ОС ЕС;

RAM=**аа** — список модулей программ метода доступа, которые должны быть включены в резидентную часть ОС ЕС. Цифры **аа** добавляются к **IEAIGG** для формирования имени раздела библиотеки **SYS1.PARMLIB**. Если этот параметр опущен, то подразумевается раздел с именем **IEAIGG00**;

REPR=**хх** — список программ обработки ошибок, которые должны быть включены в резидентную часть ОС ЕС. Цифры **хх** добавляются к **IEAIGE** для формирования имени раздела библиотеки **SYS1.PARMLIB**. Если этот параметр опущен, то подразумевается раздел с именем **IEAIGE00**;

RSVC=**хх** — список **SVC**-программ, которые должны быть включены в резидентную часть ОС ЕС. Цифры **хх** добавляются к **IEARSV** для формирования имени раздела библиотеки **SYS1.PARMLIB**. Если этот параметр опущен, то подразумевается раздел с именем **IEARSV00**. Если при генерации ОС ЕС определена область резидентных реентерабельных загрузочных модулей, то вместо описываемого здесь параметра следует использовать аналогичный параметр для ОС ЕС режима **MVT**;

SQS=**п** — определяет размер основной памяти для области системных очередей **SQA**; **п** — число байт.

Параметры для ОС ЕС режима MVT:

ALTSYS=**ddd** — то же, что для **MFT**;

BLDL=**хх[yy]** — то же, что для **MFT**;

HARDCPY=**(х[у][z])** — то же, что для **MFT**;

MIN=**пп** — то же, что для **MFT** (**пп** ≥ 52);

MOD=nn — то же, что для MFT;

MPS=nn — размер раздела основной памяти для главного планировщика; nn — число блоков по 2 Кбайта;

QBF=nnn — размер области основной памяти, необходимой для буферов очереди заданий; nnn — число блоков по 1 Кбайт (nnn ≤ 255);

RAM=aa[,bb,cc,dd] — списки программ методов доступа и другие реентерабельные модули, которые должны быть включены в резидентную часть ОС ЕС. Можно указать от одного до четырех разделов библиотеки SYS1.PARMLIB. Если этот параметр опущен, но задан при генерации, то подразумевается раздел с именем IEAIGG00;

REPR=aa[,bb,cc,dd] — списки программ обработки ошибок, которые должны быть включены в резидентную часть ОС ЕС. Можно указать от одного до четырех разделов библиотеки SYS1.PARMLIB. Если этот параметр опущен, но задан при генерации, то подразумевается раздел с именем IEAIGE00;

RSVC=aa[,bb,cc,dd] — списки SVC-программ, которые должны быть включены в резидентную часть ОС ЕС. Можно указать от одного до четырех разделов библиотеки SYS1.PARMLIB. Если этот параметр опущен, но задан при генерации, то подразумевается раздел с именем IEARSV00;

SQS=n — размер дополнительной памяти для области системных очередей SQA; n — число блоков по 2 Кбайта;

TMSL=((nn, xxxx) [(mm, уууу), ...]) — кванты времени для групп задач в зависимости от приоритетов; nn, nnn — приоритеты групп от 0 до 13;

xxxx, уууу — величины квантов времени в миллисекундах.

Параметры для ОС ЕС режима SVS:

BLDL=xx — список модулей библиотеки SYS1.LINKLIB, для которых с помощью макрокоманды BLDL в страничной области памяти должен быть построен резидентный список модулей;

CLPA — удаление старой и создание новой области LPA;

CPQE=nnnn — добавляемое к минимальному число программ канала ввода-вывода для супервизора страниц; nnnn — число от 0 до 9999;

HARDCPY=(x[,y][,z]) — то же, что для MFT;

IOL=NO — отменяет сбалансированное распределение периферийных устройств при работе операционной системы;

LSQACEL=((nnn, m),...) — размеры и количество областей памяти для быстрого выделения каждой из имеющихся областей LSQA; nnnn — размер области от 8 до 256 байт и m — количество областей от 1 до 8;

MOD=nn — то же, что для MFT;

MPA=nnn — число блоков памяти размером по 64 Кбайта, которые должны быть добавлены к минимальному размеру раздела главного планировщика; nnn — число от 1 до 999;

PAGE=(x[,y][,F][,LPA]) — набор данных SYS1.PAGE. Для этого набора данных можно указать до 16 томов прямого доступа.

При этом определение для каждого тома должно заключаться в круглые скобки.

Параметр *x* принимает следующие значения:

D — вывод на консоль информации о всех определяемых томах;

U=устройство — адрес устройства прямого доступа;

V=том — регистрационный номер тома;

Параметр *y* принимает следующие значения:

BLK=nnnn — размер в блоках по 2 Кбайта для набора данных SYS1.PAGE; nnnn — число, не превышающее 4 знаков;

TRK=nnnn — количество дорожек, которые должны быть выделены на томе прямого доступа для набора данных SYS1.PAGE; nnnn — число, не превышающее 4 знаков;

CYL=nnnn — количество цилиндров, которые должны быть выделены на томе прямого доступа для набора данных SYS1.PAGE; nnnn — число, не превышающее 4 знаков;

MAX — выделение на томе прямого доступа для набора данных SYS1.PAGE наибольшего свободного экстенда;

F — набор данных SYS1.PAGE должен быть сформатирован заново;

LPA — размещение области LPA в наборе данных SYS1.PAGE. Этот параметр указывается только для одного тома;

REAL=nnnn — число блоков размером 2 Кбайта каждый, которые должны быть добавлены к минимальному размеру области V=R; nnnn — число от 1 до 9999;

SQA=nnn — число блоков размером 64 Кбайта каждый, которые должны быть добавлены к минимальному размеру области системных очередей SQA; nnn — число от 1 до 999;

SQACEL=(nnnn, m) — размер и число участков для быстрого выделения памяти в SQA; nnnn — размер от 8 до 256 байт и m — число от 1 до 8. Общий размер определяемой области не должен превышать 4 Кбайт.

TMSL=((nn, xxxx)[, (mm, yyyy)...]) — кванты времени для групп задач в зависимости от приоритетов; nn, mm — приоритеты групп от 0 до 13; xxxx, yyyy — величины квантов времени в миллисекундах.

Сообщение

IEA103I DATASET dsn NOT FOUND BY LOCATE

информирует о наборе данных операционной системы, который был определен при генерации, но не обнаружен программой инициализации ядра при выполнении процедуры начальной загрузки.

Сообщение

IEA135A SPECIFY SYS1.DUMP TAPE ADDRESS OR NO

предписывает определить устройство для набора данных SYS1.DUMP, используемого операционной системой для размещения дампа основной памяти в случае ошибки в любой из системных задач. В качестве ответа следует ввести R 00, 'ddd', где ddd — адрес накопителя на магнитной ленте, содержащего том для за-

писи дампа или R 00, 'NO', если в этой функции нет необходимости.

Сообщение

IEA2081 fff FUNCTION INOPERATIVE

информирует об отмене функции fff.

Сообщение

00 IEE801D CHANGE PARTITIONS—REPLY YES/NO(LIST)

предписывает ответить, желательно ли перераспределение основной памяти операционной системы режима MFT.

В качестве ответа следует ввести

R 00, {YES
NO } [LIST]'

где YES — перераспределение основной памяти;

NO — распределение основной памяти, заданное при генерации, остается без изменений.

LIST — вывод на консоль информации о текущем распределении основной памяти.

При положительном ответе на консоль будет выведено сообщение:

идент IEE802A ENTER DEFINITION

В ответе на это сообщение указываются новые размеры разделов основной памяти, классы заданий для каждого раздела и параметры квантования времени. Каждая строка ответа не должна превышать 126 символов. Ответ может состоять из одного и более параметров, разделенных запятыми:

параметр P_n = (класс, размер, LAST) — определяет классы заданий и размер раздела с номером n,

класс — одна, две или три буквы от A до O, RDR или WTR.

Последние два значения подпараметра определяют раздел резидентной программы системного ввода или вывода;

размер — размер раздела основной памяти в байтах или Кбайтах. В последнем случае за числом должна следовать буква K. Для указания неактивного раздела в качестве значения этого подпараметра следует указать 0 или OK;

LAST — последний раздел.

Любой из перечисленных подпараметров может быть опущен, если в его переопределении нет необходимости. Вслед за параметрами P_n или вместо них в любом порядке и сочетании могут быть заданы параметры LIST, TMSL, CLASS, END и CANCEL.

LIST — вывод на консоль списка всех разделов основной памяти.

CLASS — вывод на консоль списка классов заданий для всех активных разделов основной памяти.

TMSL = ($P_1 - P_m$, pppp) — квантование времени для разделов $P_1 - P_m$ ($1 < m$) и величина кванта времени pppp в миллисекундах.

Любой из перечисленных подпараметров, значение которого не изменяется, может быть опущен. Для отмены квантования вре-

мени, определенного при генерации, следует указать TMSL = CANCEL.

CANCEL — отменяет ранее определенные значения параметров, оставляя их такими, как они были заданы при генерации операционной системы.

END — определяет конец ответа. При вводе ответа без этого параметра на консоли появится сообщение IEE803A CONTINUE DEFINITION, позволяющее продолжить переопределение разделов основной памяти.

Если в ответах на сообщения IEE801D и IEE802A, IEE803A будут указаны соответственно параметры NO и END, то на консоли появятся сообщения: IEE866I, IEE812I, IEE805I и IEE101A.

Первое из сообщений указывает, что заданное в ответе распределение разделов основной памяти принято и обрабатывается операционной системой. Второе сообщение информирует о размере остатка основной памяти, приписанного последнему активному разделу. Сообщение IEE805I позволяет сделать вывод о завершении обработки операционной системой ответа оператора ОС ЕС ЭВМ на сообщения: IEE801D, IEE802A и IEE803A.

Независимо от режима работы операционной системы на завершение выполнения программы инициализации ядра и передачу управления главному планировщику указывает сообщение

IEE101A READY

Следующие за ним сообщения в формате команды START (см. 2.3.9) появляются на консоли, если при генерации операционной системы были определены автоматически запускаемые процедуры.

Завершается процедура начальной загрузки вводом команды SET (см. 2.3.8). При этом если в команде SET указывается операнд Q с параметром F, то на консоли появится сообщение:

идент IEE423A SPECIFY JOB QUEUE PARAMETERS

Это сообщение требует определить формат набора данных SYS1.SYSJOBQE. Ответ на это сообщение имеет следующий формат:

R идент, $\left\{ \begin{matrix} U \\ n, t, k, p \end{matrix} \right\}'$

где U — определяет формат набора данных SYS1.SYSJOBQE в соответствии с параметрами, заданными при генерации операционной системы. Параметры n, t, k, p определяют новый формат набора данных SYS1.SYSJOBQE:

n — число записей на одной логической дорожке ($10 \leq n \leq 255$);

t — число записей, приписанных каждому инициатору ($t \leq 9999$);

k — число записей для сохранения информации о заданиях, требующих для инициирования их выполнения более t записей ($k \leq 9999$);

p — число записей для сохранения сообщений оператору ОС ЕС ЭВМ.

Любой из параметров n , t , k или p может быть опущен, если его значение не изменяется. Опускаемые в ответе предыдущие параметры отмечаются запятыми.

3.4. Управление консолями

3.4.1. Уровни консольного обеспечения ОС ЕС

В качестве консолей в операционной системе ОС ЕС ЭВМ используются следующие типы периферийных устройств¹:

пишущие машинки: ЕС-7070, ЕС-7073, ЕС-7077;

алфавитно-цифровые дисплеи: ЕС-7061, ЕС-7063, ЕС-7066, ЕС-7927-01;

графические дисплеи: ЕС-7064, ЕС-7065;

пары периферийных устройств: ЕС-6012 и ЕС-7032, ЕС-6016 и ЕС-7035 и т. д.

При генерации операционной системы определяется одноконсольное или мультиконсольное обеспечение.

Одноконсольное обеспечение предусматривает в составе операционной системы не более двух консолей — основную и альтернативную. При этом в качестве консолей не допускается использование дисплеев. В состоянии активной консоли при одноконсольном обеспечении может находиться только одна консоль, другая остается резервной. Альтернативная консоль может перейти в состояние активной консоли, если во время загрузки операционной системы основная консоль окажется неработоспособной или вследствие выполнения оператором ОС ЕС ЭВМ переключения консолей.

Для переключения консолей оператор ОС ЕС ЭВМ должен выполнить следующие действия:

ввести с активной консоли команду VARY OFFLINE (см. 2.4.3) для устройств, входящих в состав резервной консоли и находящихся в оперативном состоянии;

нажать кнопку ПРЕРЫВАНИЕ на пульте управления ЭВМ.

Мультиконсольное обеспечение предоставляет возможность управлять работой операционной системы, используя для этого до 32 активных консолей, причем в качестве консолей могут использоваться алфавитно-цифровые и графические дисплеи. При генерации операционной системы определяется основная консоль, дополнительные консоли и для каждой из них — альтернативная консоль. Каждая консоль операционной системы с мультиконсольным обеспечением обладает следующими характеристиками:

набором команд оператора ЭВМ, разрешенных для ввода через данную консоль;

составом сообщений, разрешенных для вывода на эту консоль; адресом альтернативной консоли.

¹ Далее в пунктах 3.4.2. — 3.4.9 описываются приемы работы только для устройств, выпускаемых в СССР.

Перечисленные характеристики также задаются во время генерации операционной системы и могут изменяться в процессе работы оператором ОС ЕС ЭВМ. Кроме этого оператор ОС ЕС ЭВМ может изменить состояние консолей, переопределить сборный протокол работы консолей и выполнить переключение консолей в случае их отказа. Все операции, кроме последней, осуществляются с помощью команд управления консолями (см. 2.5).

В случае отказа консолей операционная система пытается осуществить их переключение автоматически, переводя в состояние активной консоли альтернативную консоль. В случае неработоспособной альтернативной консоли переключение будет произведено на основную консоль. Если альтернативная консоль для данной консоли является одной из дополнительных консолей, то к ее характеристикам будут добавлены характеристики отказавшей консоли. Когда операционная система не в состоянии произвести автоматического переключения консолей, то на все дополнительные консоли поступает сообщение IEE141A. В этом случае оператор ОС ЕС ЭВМ с работоспособной консоли вводит команду VARY MSTCONS (см. 2.5.2) и таким образом переопределяет основную консоль. Если операционная система не в состоянии обнаружить отказ основной консоли, то для переключения ее на альтернативную консоль следует нажать кнопку ПРЕРЫВАНИЕ на пульте ЭВМ.

3.4.2. Приемы работы с пишущей машинкой ЕС-7070

Ввод команды в операционную систему осуществляется с помощью алфавитно-цифровой клавиатуры (АЦК) и клавиш ВН, СД и КТ. Для ввода команды необходимо выполнить следующие действия:

нажать последовательно клавиши ВН, СД и дождаться включения индикатора ЧИТ;

набрать на клавиатуре требуемый текст и последовательно нажать клавиши КТ и СД.

Отмена команды до ввода ее в операционную систему выполняется, если последовательно нажать клавиши АН и СД.

3.4.3. Приемы работы с пишущей машинкой ЕС-7077

Ввод команды в операционную систему осуществляется с помощью АЦК и клавиш ВН и КТ. Для ввода команды необходимо выполнить следующие действия:

нажать клавишу ВН и дождаться включения индикатора Ввод;

набрать на клавиатуре требуемый текст и нажать клавишу КТ.

Отмена команды до ввода ее в операционную систему выполняется, если нажать клавишу АН.

3.4.4. Приемы работы с составной консолью

Обычно в качестве составной консоли используются перфокарточное устройство ввода и печатающее устройство.

Для ввода команд в операционную систему с такой составной консоли необходимо выполнить следующие действия:

нажать клавишу СТОП;

вложить в приемный карман колоду перфокарт с командами оператора ОС ЕС ЭВМ;

нажать клавишу ПУСК и КОНЕЦ КАРТОТЕКИ.

Текст команды, вводимой с составной консоли, должен начинаться с первой колонки перфокарты.

Функция «ВНИМАНИЕ» моделируется последовательным нажатием клавиш СТОП и ПУСК.

3.4.5. Ввод команд оператора ОС ЕС ЭВМ во входном потоке

С устройства программы системного ввода в операционную систему могут быть введены все команды, кроме DEFINE, HALT, SWAP и MODE.

В первых двух колонках перфокарты с командой должны быть отперфорированы знаки //, а далее через один или более пробел должен следовать текст команды. Каждая команда должна целиком размещаться на одной перфокарте.

Команды оператора ОС ЕС ЭВМ располагаются во входном потоке перед операторами JOB и EXEC.

Режим обработки команд во входном потоке устанавливается в поле PARM процедуры для запуска программы системного ввода. В частности, можно задать режим, при котором каждая введенная команда будет выполняться после подтверждения оператором ОС ЕС ЭВМ. В этом случае с текстом введенной команды на консоль выводится сообщение:

идент IEF166D REPLY Y/N TO EXECUTE/SUPPRESS COMMAND

Формат ответа оператора ОС ЕС ЭВМ:

$$R \text{ идент. } \left\{ \begin{array}{c} Y \\ N \end{array} \right\}$$

где идент — идентификатор, указанный в сообщении;

Y — разрешает выполнение команды;

N — отменяет выполнение команды.

3.4.6. Дисплеи в качестве консолей

Дисплеи в качестве консолей, или дисплей-консоли, помимо увеличения скорости вывода сообщений, позволяют оператору ОС ЕС ЭВМ корректировать команды перед вводом их в ЭВМ, разделять сообщения операционной системы на классы, а для некоторых типов устройств — вводить команды параллельно с выводом сообщений.

Экран каждой дисплей-консоли делится операционной системой на четыре области: область сообщений, строка инструкций, об-

ласть ввода и строка предостережений. Расположение областей и количество строк экрана, выделяемых для каждой области в зависимости от типа устройства, приводятся в табл. 3.1.

Таблица 3.1

Расположение и размер областей экрана

Тип устройства	Длина строки, в символах	Строки области сообщений	Строка инструкций	Строки области ввода	Строки предостережений
ЕС-7066	80	1—8	9	10—11	12
ЕС-7061	80	1—8	9	10—11	12
ЕС-7063	80	1—8	9	10—11	12
ЕС-7064	74	1—47	49	50—51	52
ЕС-7065	74	1—47	49	50—51	52
ЕС-7927-01	80	1—19	21	22—23	24

Ниже приведен формат строк области сообщений.

Формат строк области сообщений

Позиция в строке	Содержимое
1—2	Номер строки или число сообщений, ожидающих высвечивания
3	Вертикальная черта () для удаляемых сообщений или для сообщений, не требующих дальнейших действий. Горизонтальная черта (—) для информационных сообщений. Пробел для сообщений, требующих принятия решений оператором ОС ЕС ЭВМ
4	Звездочка (*) или символ @ для сообщений, требующих вмешательства оператора ОС ЕС ЭВМ. Пробел для сообщений, не требующих немедленного действия, и сообщений, содержащих вертикальную или горизонтальную черту в позиции 3
5—74 или 5—77	Текст сообщения

В строке инструкций появляются сообщения об ошибках оператора ОС ЕС ЭВМ и инструкции по управлению дисплей-консолью. Кроме этого в этой строке можно высветить номера клавиш функциональной клавиатуры, используемых для ввода команд.

Область ввода используется для набора команд оператора ОС ЕС ЭВМ и ответов на сообщения операционной системы. Текст команды должен располагаться в первых 126 позициях области ввода.

В строке предостережений появляются сообщения, предупреждающие оператора ОС ЕС ЭВМ о сообщениях, ожидающих высвечивания, и других особых случаях.

3.4.7. Приемы работы с дисплеями ЕС-7064 и ЕС-7065

Оператор ОС ЕС ЭВМ может использовать для работы с дисплей-консолью ЕС-7064 и ЕС-7065 световое перо, алфавитно-цифровую (АЦК) и функциональную (ФК) клавиатуры.

Чтобы с помощью светового пера запросить операционную систему, выполнить некоторую функцию, следует указать световым пером соответствующее изображение на экране дисплея. Для этого необходимо направить световое перо на любой элемент изображения так, чтобы он начал мерцать, а затем слегка надавить концевым переключателем светового пера на стекло экрана. При правильном исполнении перечисленных действий изображение на экране на некоторое время гаснет.

Ввод команды из области ввода осуществляется с помощью АЦК, ФК и светового пера. Для ввода команды с помощью АЦК или светового пера необходимо набрать текст команды и выполнить одно из следующих двух действий:

одновременно нажать клавиши РЗР и КОН;

указать световым пером *ENTER* в строке инструкций.

Для ввода команды с помощью ФК следует нажать соответствующую команде функциональную клавишу.

Удаление сообщений с экрана дисплея может быть выполнено с помощью светового пера или команды CONTROL (см. 2.2.1).

Для удаления сообщения, не требующего действия, с помощью светового пера следует указать световым пером любой символ в тексте выбранного сообщения.

Для удаления сообщения, требующего действия, следует указать световым пером звездочку в четвертой позиции строки, содержащей это сообщение.

Сообщения, представляющие собой информацию состояния (ответ операционной системы на ввод команды DISPLAY), удаляются, если указать световым пером символ *E* в строке заголовка текущего кадра. Для высвечивания следующего кадра информации состояния следует указать световым пером символ *F* в строке заголовка текущего кадра.

Отмена запроса к операционной системе или команды, находящейся в области ввода, осуществляется одновременным нажатием клавиш РЗР и ОТМ или указанием слова *CANCEL* в строке инструкций.

3.4.8. Приемы работы с дисплеем ЕС-7066

При работе с дисплеем ЕС-7066 оператор ОС ЕС ЭВМ использует алфавитно-цифровую клавиатуру и клавиши автономного управления экраном. Прежде чем вводить команды, оператор ОС ЕС ЭВМ должен приостановить вывод сообщений на экран дисплея, т. е. ввести запрос на ввод команды. Для этого следует убедиться, что курсор находится в позиции, следующей за символом ▷, и нажать клавишу Вв. В ответ операционная система помещает в 11-ю строку экрана сообщение

IEE167E OUTPUT IN HOLD MODE

Это сообщение означает, что запрос на ввод команды принят операционной системой.

Ввод команды затем осуществляется с помощью АЦК и клавиши Вв. Для этого необходимо набрать текст команды и нажать клавишу Вв.

Для отмены запроса на ввод команды необходимо последовательно нажать клавиши ▷ и Вв.

Удаление сообщений с экрана дисплея ЕС-7066 выполняется с помощью курсора или команды CONTROL (см. 2.2.1).

Для удаления сообщений, не требующих действия, следует ввести запрос на ввод команды, переслать курсор в строку, содержащую сообщение, начиная с которой следует удалить все сообщения этого типа.

Для удаления сообщений, требующих действия, следует ввести запрос на ввод команды, переслать курсор в четвертую позицию строки выбранного сообщения и нажать клавишу Вв.

Сообщения информации состояния удаляются, если курсор переслать в позицию символа *Е* заголовка и нажать клавишу Вв.

Для высвечивания следующего кадра информации состояния следует переслать курсор в позицию символа *F* заголовка и нажать клавишу Вв.

3.4.9. Приемы работы с дисплеем ЕС-7927-01

При работе с дисплеем ЕС-7927-01 оператор ОС ЕС ЭВМ использует алфавитно-цифровую и функциональную клавиатуры, клавиши автономного управления экраном и световое перо. Как и для дисплеев ЕС-7064 и ЕС-7065, чтобы запросить с помощью светового пера выполнение операционной системой некоторой функции, следует указать световым пером соответствующее изображение на экране.

Ввод команды из области ввода осуществляется с помощью АЦК, ФК и светового пера. Для ввода команды с помощью АЦК или светового пера необходимо набрать текст команды и выполнить одно из следующих действий:

нажать клавишу ВВОД;

указать световым пером *ENTER* в строке инструкций.

Для ввода команды с помощью ФК следует нажать соответствующую команде функциональную клавишу.

Удаление сообщений с экрана дисплея может быть выполнено с помощью светового пера (см. 3.4.7) курсора (см. 3.4.8) или команды CONTROL (см. 2.2.1).

Отмена запроса к операционной системе или команды, находящейся в области ввода, осуществляется нажатием клавиши МТКП или указанием световым пером слова *CANCEL* в строке инструкций.

Язык управления заданиями

Язык управления заданиями включает девять операторов, которые называются *управляющими операторами*. Это оператор задания (оператор JOB), оператор пункта задания (оператор EXEC), оператор определения данных (оператор DD), оператор команды, оператор комментариев, ограничительный оператор, пустой оператор, оператор процедуры (оператор PROC) и оператор конца процедуры (оператор PEND). Отличительной особенностью управляющих операторов является то, что они начинаются с комбинации «//» (исключение составляют ограничительный оператор, начинающийся с комбинации «/*», и оператор комментариев, начинающийся с комбинации «// *»).

4.1. Оператор задания (оператор JOB)

Оператор JOB является первым в задании. Формат оператора:

```
// имя JOB [операнды] [комментарии]
```

где имя — имя задания, содержащее от 1 до 8 алфавитно-цифровых символов, причем первым символом должна быть буква. Нельзя использовать в качестве имени задания следующие имена: A, CONSOLES, DSNAME, JOBNAMEs, N, Q, R, SPACE, T, U, являющиеся именами ключевых параметров команды DISPLAY.

Операнды включают два позиционных параметра, которые должны быть первыми, и ряд ключевых; все параметры являются необязательными, однако параметры учетной информации и идентификации программиста могут быть обязательными в некоторых конкретных вариантах операционной системы; комментарии могут включать любую информацию, предусмотренную для комментирования.

4.1.1. Параметр учетной информации

Параметр учетной информации — первый позиционный параметр оператора JOB. Он содержит учетный номер и дополнительную учетную информацию. Если параметр содержит несколько подпараметров, то они заключаются в скобки или апострофы и разделяются запятыми. Необходимость параметра учетной информации устанавливается в каталогизированной процедуре систем-

ного ввода. Он обрабатывается программами обработки учетной информации, которые включаются при генерации либо могут быть включены в сгенерированную операционную систему. Формат параметра:

(учетный-номер, [дополнительная-учетная-информация], ...)

где подпараметры могут состоять из одного или нескольких символов, определяемых программами обработки, но не более чем из 142, включая разделяющие запятые. Если учетная информация содержит символы, отличные от букв, цифр и дефиса, то она заключается в апострофы. Символы апострофа и амперсанда, являющиеся частью учетной информации, кодируются соответственно двумя апострофами или двумя амперсандами.

4.1.2. Параметр идентификации программиста

Параметр идентификации программиста является вторым позиционным параметром оператора JOB. Как и предыдущий параметр, он обрабатывается программой обработки учетной информации. Формат параметра:

идентификатор-программиста

где идентификатор-программиста—произвольная комбинация символов не более 20. Если он содержит символы, отличные от букв, цифр, и точки, то его необходимо заключить в апострофы. Символы апострофа, являющиеся частью идентификатора программиста, кодируются двумя апострофами.

4.1.3. Параметр CLASS

Параметр CLASS определяет класс задания. Формат параметра:

CLASS= класс-задания

где класс-задания указывается латинской буквой от A до O. По умолчанию предполагается A. Класс определяет очередь заданий, в которую попадет задание.

4.1.4. Параметр PRTY

Параметр PRTY определяет приоритет задания, в соответствии с которым оно выбирается из очереди заданий для выполнения. Формат параметра:

PRTY=число

где число — целое число без знака от 0 до 13, определяющее приоритет задания.

4.1.5. Параметр MSGLEVEL

Параметр MSGLEVEL определяет вид выходной информации о задании, которая выводится в листинг, и управляет следующей

выходной информацией: информацией об управляющих операторах языка управления заданиями и сообщениями о распределении и завершении. Формат параметра:

$$\text{MSGLEVEL} = \left(\begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right)$$

где для первого подпараметра: 0 — вывод только операторов JOB; 1 — вывод всех управляющих операторов, включая операторы каталогизированных процедур после подстановки символических параметров; 2 — вывод только входных управляющих операторов;

для второго подпараметра: 0 — сообщения о распределении и завершении не выводятся, если задание не завершается аварийно, в противном случае — выводятся; 1 — сообщения о распределении и завершении выводятся. Значения параметров по умолчанию устанавливаются в процедуре системного ввода.

4.1.6. Параметр MSGCLASS

Параметр MSGCLASS определяет выходной класс системного вывода системных сообщений для текущего задания. Формат параметра:

$$\text{MSGCLASS} = \text{выходной-класс}$$

где выходной-класс — латинская буква от A до Z или цифра от 0 до 9. Выходной класс определяет выходную очередь и устройство вывода, на которое попадают системные сообщения задания. Значение выходного класса по умолчанию определяется в процедуре системного ввода; если же в ней не определено, то предполагается A.

4.1.7. Параметр TYPRUN

Параметр TYPRUN применяется для задержки выборки задания из очереди для выполнения. Формат параметра:

$$\text{TYPRUN} = \text{HOLD}$$

где HOLD — выборка задания задерживается до тех пор, пока оператор не выдаст команду RELEASE.

4.1.8. Параметр ADDRSPC

Параметр ADDRSPC (только для режима SVS) определяет вид основной памяти (виртуальная или реальная), в которой выделяется раздел для задания. Формат параметра:

$$\text{ADDRSPC} = \left\{ \begin{array}{l} \text{VIRT} \\ \text{REAL} \end{array} \right\}$$

где VIRT означает, что раздел памяти должен быть выделен в виртуальной (постранично организованной) памяти;

REAL означает, что раздел памяти должен быть выделен в реальной памяти, страницы которой не участвуют в обмене. По умолчанию предполагается значение, установленное в процедуре системного ввода: если оно там не установлено, то предполагается VIRT. Если данный параметр закодирован в операторе JOB, то значения этих параметров, установленные в операторах EXEC задания, игнорируются.

4.1.9. Параметр REGION

Параметр REGION определяет в режимах MVT и SVS размер раздела основной памяти, выделяемой для задания, в режиме MFT — игнорируется. Формат параметра:

REGION=число K

где число — размер раздела в килобайтах. По умолчанию используется значение, определенное в процедуре системного ввода. В режиме SVS раздел выделяется в виртуальной или реальной памяти в зависимости от значения параметра ADDRSPC. В режиме SVS при указании виртуальной памяти размер раздела округляется до очередного числа, кратного 64, в остальных случаях округляется до очередного числа, кратного 2. Если данный параметр закодирован в операторе JOB, то значения этих параметров, установленные в операторах EXEC задания, игнорируются.

4.1.10. Параметр ROLL

Параметр ROLL в режиме MVT управляет средством свертка-развертка, в остальных режимах игнорируется.

Формат параметра:

ROLL= [{YES } {YES }
 {NO } {NO }]

где для первого подпараметра: YES — каждый пункт задания может быть свернут; NO — пункты задания не могут быть свернуты;

для второго подпараметра: YES — каждый пункт задания может вызвать свертку пункта другого задания; NO — пункты задания не могут вызвать свертку пункта другого задания.

В системных процедурах системного ввода определено следующее стандартное значение параметра: ROLL=(YES, NO).

Если данный параметр закодирован в операторе JOB, то значения этих параметров, установленные в операторах EXEC, игнорируются.

4.1.11. Параметр TIME

Параметр TIME определяет максимальное количество времени, в течение которого задание может использовать центральный процессор. Формат параметра:

TIME= { (число1, число2) }
 1440

где число1 — количество минут — должно быть меньше 1440;

число2 — количество секунд — должно быть меньше 60;

1440 — определяет, что ограничение времени использования центрального процессора не устанавливается.

Если параметр TIME не закодирован в операторе JOB, то ограничение времени использования центрального процессора для задания не устанавливается, однако для каждого пункта задания такое ограничение может быть установлено.

Если заданное ограничение превышено, то возникает либо аварийное завершение задания, либо передача управления на программу выхода, если такая предусмотрена в системной мониторинной программе. В случае отсутствия системной мониторинной программы существует ограничение на время непрерывного состояния ожидания, которое составляет 30 мин. Системная мониторинная программа может установить другое значение этого ограничения.

4.1.12. Параметр COND

Параметр COND предназначен для проверки кодов возврата, выдаваемых программами пунктов текущего задания. В параметре может быть указано от 1 до 8 условий. Если хотя бы одно условие при проверке кода возврата программы пункта задания окажется истинным, то оставшиеся пункты не выполняются, а задание завершается. Формат параметра:

COND= ((число, отношение), ...)

где число — десятичное число от 0 до 4095. Это число сравнивается с кодом возврата, выдаваемым программами пунктов задания; отношение — тип операции отношения в условии: GT (больше), GE (больше или равно), EQ (равно), NE (не равно), LT (меньше), LE (меньше или равно). Каждое условие проверки формируется следующим образом:

число отношение код-возврата-программы-пункта-задания.

Параметр COND оператора JOB проверяется раньше параметра COND оператора EXEC.

4.1.13. Параметр RD

Параметр RD определяет возможности рестарта, которые можно использовать при выполнении задания. Формат параметра:

RD= код-запроса

где код-запроса — одно из следующих значений:

R — разрешается автоматический рестарт в случае аварийного завершения пункта задания; автоматический рестарт производится либо с начала текущего пункта задания, если в нем не была выполнена ни одна макрокоманда СНКРТ, либо с последней выполненной макрокоманды СНКРТ в данном пункте задания. Если действие контрольной точки, определенной макрокомандой СНКРТ,

будет подавлено, то произойдет автоматический рестарт пункта задания;

RNC — разрешается автоматический рестарт пункта задания и не разрешается автоматический рестарт с контрольной точки;

NR — выполнение макрокоманды СНКРТ для определения контрольной точки разрешается, автоматический рестарт не разрешается;

NC — подавляется действие макрокоманд СНКРТ и никакие виды рестарта не разрешаются.

Если данный параметр закодирован в операторе JOB, то значения этих параметров в операторах EXEC игнорируются.

4.1.14. Параметр RESTART

Параметр RESTART используется при повторном предъявлении задания для выполнения в случае отсроченного рестарта пункта задания или с контрольной точки. Формат параметра:

RESTART= имя-пункта [идентификатор-контрольной-точки]

где имя-пункта — имя пункта задания или составное имя:

имя пункта задания.имя пункта процедуры или знак *, указывающее имя пункта, с (или внутри) которого должен произойти отсроченный рестарт (* означает имя первого пункта задания);

идентификатор-контрольной-точки — имя, идентифицирующее контрольную точку, указанное в макрокоманде СНКРТ, с которой производится отсроченный рестарт. Если идентификатор контрольной точки отсутствует, то производится рестарт пункта задания, имя которого указано в первом подоперанде. Если отсроченный рестарт выполняется с контрольной точки, то в задание необходимо включить дополнительный оператор DD с именем SYSCHK, который описывает набор данных контрольной точки.

4.1.15. Параметр NOTIFY

Параметр NOTIFY предназначен для передачи пользователю терминала, работающему в системе разделения времени, сообщения о завершении его задания, которое выполнялось в пакетном режиме. Формат параметра:

NOTIFY=идентификатор-пользователя

где идентификатор-пользователя — идентификатор, который указывается в начале сеанса в команде LOGON и содержит от 1 до 7 алфавитно-цифровых символов, первым из которых должна быть буква.

4.2. Оператор пункта задания (оператор EXEC)

Оператор EXEC является первым в пункте задания. Формат оператора:

// [имя] EXEC операнды [комментарий]

где имя — имя пункта задания или имя пункта процедуры, содержащее от 1 до 8 алфавитно-цифровых символов, причем первым должна быть буква. Имя пункта задания является необязательным, однако рекомендуется каждому пункту присваивать уникальное имя, так как ряд функций управляющей программы идентифицирует пункты задания по их имени;

операнды включают два взаимно исключающих позиционных параметра, которые должны быть первыми, причем один из них должен быть указан обязательно, и ряд ключевых параметров, которые необязательны;

комментарии могут включать любую информацию.

4.2.1. Параметр PGM

Параметр PGM — первый позиционный параметр, определяющий программу, которая должна быть выполнена (программу пункта задания). Программа должна быть разделом библиотеки. Она может быть задана либо путем указания имени раздела или дополнительного имени библиотеки, либо путем обратной ссылки на оператор DD, в котором определен раздел библиотеки. Формат параметра:

$$\text{PGM} = \left\{ \begin{array}{l} \text{имя-программы} \\ *.\text{имя-пункта.имя-dd} \\ *.\text{имя-пункта.имя-пункта-процедуры.имя-dd} \end{array} \right\}$$

где имя-программы — имя или дополнительное имя раздела библиотеки. Программа может быть разделом либо общей библиотеки SYS1.LINKLIB, либо библиотеки задания (//JOB LIB DD...), либо библиотеки пункта задания (//STEPLIB DD...), либо личной библиотеки;

*.имя-пункта.имя-dd — обратная ссылка на оператор DD, определяющий раздел библиотеки в одном из предыдущих пунктов данного задания. Данная форма обычно используется для указания раздела временной библиотеки;

*.имя-пункта.имя-пункта-процедуры.имя-dd — обратная ссылка на оператор DD, определяющий раздел библиотеки в пункте каталогизированной процедуры, вызываемой в одном из предыдущих пунктов данного задания.

4.2.2. Параметр PROC

Параметр PROC является первым позиционным параметром, определяющим каталогизированную процедуру, которая должна быть вызвана для выполнения. Формат параметра:

$$\left\{ \begin{array}{l} \text{PROC=имя-процедуры} \\ \text{имя-процедуры} \end{array} \right\}$$

где имя-процедуры — имя или дополнительное имя каталогизированной процедуры или имя процедуры во входном потоке, которая

вызывается для выполнения. Имя процедуры содержит от 1 до 8 алфавитно-цифровых символов, первым из которых должна быть буква.

4.2.3. Параметр PARM

Параметр PARM предназначен для передачи параметров программе пункта задания во время ее выполнения. Формат параметра:

PARM= параметры

где параметры — до 100 символов информации, которую система передает в качестве параметров программе пункта задания. Для передачи параметров организуется список параметров в соответствии с соглашением о связях. Если параметры содержат несколько выражений, разделенных запятыми, они должны быть заключены в скобки или апострофы. Параметры, заключенные в апострофы, не могут быть перенесены на другую строку. В скобки могут быть заключены параметры, состоящие только из алфавитно-цифровых символов.

4.2.4. Параметр ACCT

Параметр ACCT содержит учетную информацию. Если параметр содержит несколько подпараметров, то они разделяются запятыми и заключаются в скобки. Учетная информация обрабатывается программами обработки учетной информации, которые включаются при генерации либо могут быть включены в сгенерированную операционную систему. Формат параметра:

ACCT=(учетная-информация, ...)

где учетная-информация — один или более подпараметров для программ обработки учетной информации. Общая длина параметра, включая разделяющие запяты, не должна превышать 142 символов. Подпараметры, содержащие символы, отличные от букв, цифр и дефиса, заключаются в апострофы. Символ апострофа, являющийся частью учетной информации, кодируется двумя апострофами.

4.2.5. Параметр DPRTY

Параметр DPRTY устанавливает текущий приоритет задачи пункта задания. Формат параметра:

DPRTY=(число1, число2)

где число1, число2 — число от 0 до 15.

Число 15 указывать не рекомендуется, так как оно используется системными задачами. Если число1 опущено, по умолчанию предполагается 0; если число2 опущено, по умолчанию предполагается 11.

Текущий приоритет вычисляется по формуле: $\text{число}1 \times 16 + \text{число}2$. Если данный параметр опущен, то текущий приоритет вычисляется на основании значения приоритета задания: $\text{значение параметра PRTY} \times 16 + 11$.

В режиме MFT без подзадач данный параметр игнорируется.

4.2.6. Параметры ADDRSPC, REGION, TIME и RD

Параметры ADDRSPC, REGION, TIME и RD в операторе EXEC имеют тот же формат и значение для пункта задания, что и аналогичные параметры в операторе JOB для задания.

4.2.7. Параметр ROLL

Параметр ROLL в операторе EXEC имеет те же формат и значение, но только лишь в отношении данного пункта задания, что и аналогичный параметр в операторе JOB.

4.2.8. Параметр COND

Параметр COND предназначен для проверки кодов возврата, выдаваемых программами предшествующих пунктов текущего задания, а также для выполнения текущего пункта задания в случаях аварийных завершений предшествующих пунктов. В параметре может быть указано от 1 до 8 условий проверки кодов возврата предшествующих пунктов либо условия могут отсутствовать. Если хотя бы одно условие окажется истинным, то текущий пункт задания не выполняется. Формат:

$$\text{COND} = (\left[\begin{array}{l} \text{(число, отношение)} \\ \text{(число, отношение, имя пункта)} \\ \text{(число, отношение, имя-пункта. имя-п-проц.)} \end{array} \right] \dots [\left[\begin{array}{l} \text{EVEN} \\ \text{ONLY} \end{array} \right])$$

где число — десятичное число от 0 до 4095. Это число сравнивается с кодами возврата программ пунктов задания;

отношение — тип операции отношения в условии: GT (больше), GE (больше или равно), EQ (равно), NE (не равно), LT (меньше), LE (меньше или равно). Каждое условие проверки формируется следующим образом:

число отношение код-возврата-программы-пункта-задания;

имя-пункта — определяет предшествующий пункт задания, для которого производится проверка условия;

имя-пункта. имя-п-проц — определяют предшествующий пункт задания (только в процедуре), для которого производится проверка условия. Если предшествующий пункт не указан, то производится проверка условий для кодов возвратов всех предшествующих программ пунктов задания;

EVEN — указывает, что текущий пункт задания должен быть выполнен, даже если один или несколько предшествовавших пунк-

тов задания завершились аварийно. Если выполняются проверки условий кодов возврата и хотя бы одно из них будет истинным, то текущий пункт задания не выполняется;

ONLY — указывает, что текущий пункт задания должен быть выполнен, только если один или несколько предшествовавших пунктов задания завершилось аварийно. Если выполняются проверки условий кодов возврата и хотя бы одно из них будет истинным, то текущий пункт задания не выполняется.

Параметр COND оператора EXEC проверяется только в том случае, если проверки условий параметра COND оператора JOB текущего задания не были удовлетворены.

4.3. Оператор определения данных (оператор DD)

Оператор DD предназначен для определения наборов данных, используемых в пункте задания. Операторы DD следуют (как правило) за оператором EXEC того пункта задания, наборы данных которого они определяют. Формат оператора:

//[имя] DD операнды [комментарии]

имя — имя оператора DD, содержащее от 1 до 8 алфавитно-цифровых символов, первый из которых — буква. Имя оператора DD может быть опущено только в двух случаях: во втором и последующих операторах DD, определяющих сцепленные наборы данных; во втором и третьем операторах DD, определяющих индексно-последовательный набор данных. Во всех остальных случаях имя оператора DD является обязательным.

Существуют 5 специальных имен: JOBLIB, STEPLIB, SYSABEND, SYSUDUMP, SYSCHK. Имя JOBLIB предназначено для оператора DD, определяющего библиотеку задания; этот оператор должен непосредственно следовать за оператором JOB. Имя STEPLIB предназначено для оператора DD, определяющего библиотеку пункта задания; этот оператор может быть расположен в любом месте среди операторов DD пункта задания. Имена SYSABEND и SYSUDUMP предназначены для операторов DD, определяющих набор данных для выдачи дампа в случае аварийного завершения задачи. Если используется имя SYSABEND, то дамп будет включать содержимое области памяти раздела, ядра, общих регистров, таблицы прослеживания и управляющих блоков. Если используется имя SYSUDUMP, то дамп будет включать те же элементы, кроме ядра. Эти операторы могут быть расположены в любом месте среди операторов DD пункта задания. Имя SYSCHK предназначено для оператора DD, определяющего набор данных контрольной точки, который формируется при начальном выполнении задания и затем будет использован при рестарте. Этот оператор должен быть расположен непосредственно перед первым оператором EXEC текущего задания;

операнды — включают четыре взаимно исключающих параметра, которые могут использоваться только в качестве первого опе-

ранда, и ряд ключевых параметров. Все параметры являются необязательными, однако хотя бы один из них должен быть указан обязательно. Исключение составляет случай, когда производится замещение операторов DD каталогизированной процедуры, определяющих сцепленные наборы данных (см. 4.10); комментарии могут включать любую информацию.

4.3.1. Параметр *

Параметр * указывает на то, что за текущим оператором DD следует набор данных (набор данных во входном потоке). Признаком конца набора данных являются либо ограничительный оператор /*, либо любой следующий управляющий оператор потока заведений, признаком которого является комбинация //. В связи с этим записи набора данных не могут содержать комбинаций /* и // в 1-й и 2-й позициях. За оператором DD с параметром * могут следовать несколько групп данных, разделенных оператором /*. В этом случае каждой группе, которой не предшествует оператор DD *, операционная система обеспечивает оператор DD * с именем SYSIN.

Формат оператора DD с параметром *:

//имя DD *

4.3.2. Параметр DATA

Параметр DATA указывает на то, что за текущим оператором DD следует набор данных (набор данных во входном потоке). Признаком конца набора данных может служить только ограничительный оператор /*. В связи с этим записи набора данных не могут содержать комбинацию /* в 1-й и 2-й позициях.

Параметр DATA используется для ввода наборов данных, содержащих управляющие операторы (например, процедура, которая должна быть каталогизирована). За оператором DD с параметром DATA может следовать несколько групп данных, разделенных оператором /*. Формат оператора DD с параметром DATA:

//имя DD DATA

4.3.3. Параметр DUMMY

Параметр DUMMY дает возможность не выполнять операции ввода-вывода, распределение периферийных устройств и внешней памяти и диспозицию для набора данных, к которому обращаются через последовательные методы доступа. При попытке чтения такого набора данных выдается признак конца файла. При попытке записать в такой набор данных данные не передаются. Формат оператора DD с параметром DUMMY:

//имя DD DUMMY

4.3.4. Параметр DYNAM

Параметр DYNAM кодируется, чтобы указать на использование динамического распределения наборов данных, что позволяет отложить определение набора данных до тех пор, пока он не потребуются. Во время обработки процедуры LOGON никакие периферийные устройства или внешняя память не распределяются набору данных, описанному оператором DD DYNAM. Формат оператора DD с параметром DYNAM:

//имя DD DYNAM

4.3.5. Параметр DDNAME

Параметр DDNAME позволяет отложить определение набора данных на последующие операторы того же самого задания. В каталогизированных процедурах этот параметр позволяет отложить определение набора данных и произвести его в пункте задания, вызывающем процедуру. Формат параметра:

DDNAME=имя-dd

4.3.6. Параметр SYSOUT

Параметр SYSOUT определяет выходной класс системного вывода для выходного набора данных. Формат параметра:

SYSOUT=(выходной-класс [имя-программы] [номер-формата])

где выходной-класс — латинская буква от A до Z или цифра от 0 до 9. Выходной класс определяет выходную очередь и устройство вывода, на которое выводится набор данных;

имя-программы — имя программы, которая должна использоваться для вывода набора данных вместо программы системного вывода. Если подоперанд опущен, то используется программа системного вывода;

номер-формата — от 1 до 4 алфавитно-цифровых символов, определяющих специальный формат носителя (бумаги на АЦПУ или перфокарты), на который будет производиться вывод набора данных.

4.3.7. Параметр DSNAME

Параметр DSNAME определяет имя набора данных, описываемого оператором DD. Формат параметра:

$$\left. \begin{array}{l} \{ \text{DSNAME} \\ \{ \text{DSN} \end{array} \right\} = \left\{ \begin{array}{l} \text{имя-набора-данных} \\ \text{имя-набора-данных (имя-раздела)} \\ \text{имя набора-данных (область)} \\ \text{имя набора-данных (номер)} \\ \text{*имя-dd} \\ \text{*имя-пункта.имя-dd} \\ \text{*имя-пункта. имя-пункта-процедуры. имя-dd} \end{array} \right\}$$

где имя-набора-данных состоит из алфавитно-цифровых символов, дефиса и +0 (12-0 пробивки), причем первым должна быть буква. Имя набора данных может быть либо простым, либо составным. Простое имя содержит от 1 до 8 символов. Имя временного набора данных должно начинаться с одного или двух амперсандов. Составное имя, применяемое только для каталогизированных наборов данных, состоит из нескольких простых имен, разделенных точкой, каждое из которых соответствует индексным уровням в каталоге. Составное имя может содержать не более 44 символов (для составного имени поколения набора данных — не более 35), включая разделяющие точки;

имя-раздела — имя раздела или дополнительное имя библиотеки. В этом случае имя набора данных должно определять библиотеку;

область — имя одной из трех областей индексно-последовательного набора данных: INDEX, PRIME или OVFLOW. В этом случае имя набора данных должно определять индексно-последовательный набор данных;

номер — ноль или целое со знаком, определяющее номер поколения набора данных;

*.имя-dd — имя набора данных, которое было использовано в одном из предшествующих операторов DD того же пункта задания;

*.имя-пункта.имя-dd — имя набора данных, которое было использовано в операторе DD одного из предшествующих пунктов задания с указанными именами;

*.имя-пункта.имя-пункта-процедуры.имя-dd — имя набора, которое было использовано в операторе DD процедуры, вызванной в одном из предшествующих пунктов задания с указанными именами.

Если параметр DSNAME отсутствует, это означает, что оператор DD определяет временный набор данных, уникальное имя которому вырабатывает операционная система.

4.3.8. Параметр UNIT

Параметр UNIT определяет тип и количество устройств, которые должны быть распределены для набора данных. Формат параметра:

$$\left\{ \begin{array}{l} \text{UNIT} = \left(\begin{array}{l} \text{адрес-устройства} \\ \text{тип-устройства} \\ \text{групповое-имя} \end{array} \right) \left[\begin{array}{l} \text{,число} \\ \text{,P} \end{array} \right] [\text{,DEFER}] [\text{,SEP} = (\text{имя-dd}, \dots)] \\ \text{UNIT} = \text{AFF} = \text{имя-dd} \end{array} \right\}$$

где адрес-устройства — трехбайтовый адрес устройства;

тип-устройства — номер типа устройства, соответствующий номерам моделей устройств ввода-вывода, установленный в операционной системе;

групповое-имя — имя, установленное при генерации для какой-либо группы устройств. Имя включает от 1 до 8 алфавитно-цифровых символов;

число — указывает число устройств, распределенных для многоотомного набора данных. Если требуется только одно устройство, то подпараметр опускается;

P — означает, что требуется параллельная установка томов для многоотомного набора данных, при которой каждому тому должно быть распределено устройство; отсутствие второго подпараметра означает запрос одного устройства;

DEFER — определяет отсроченное монтирование тома (томов) на устройство (устройства), которое должно происходить в момент, когда набор данных открывается;

SER — указывает, что набору данных должны быть распределены устройства, отличные от устройств, на которых распределены наборы данных, определенные операторами DD, указанными в списке. В списке допускается до 8 операторов DD, которые должны предшествовать текущему оператору;

AFF — указывает, что набору данных должно быть распределено то же самое устройство, которое было распределено набору данных, определенному в операторе DD того же пункта задания, указанного в правой части (общность устройств). Общность устройств предполагает использование наборов данных и связанных с ним томов в последовательном порядке, а также отсроченное монтирование томов.

4.3.9. Параметр VOLUME

Параметр VOLUME определяет том, на котором будет размещен (или размещается) набор данных. Формат параметра:

{ VOLUME } = ([PRIVATE], [,RETAIN] [,порядковый-номер] [,число] [,

[SER=(регистрационный-номер,...)
REF=имя-набора-данных
REF=*имя-dd
REF=*имя-пункта.имя-dd
REF=*имя-пункта.имя-пункта-процедуры.имя-dd])

где PRIVATE — указывает на то, что том является личным. Это значит, что на этот том не могут быть распределены наборы данных, которые не имеют специального запроса на этот том. Личный том демонтируется после последнего его использования в пункте задания, за исключением случаев, когда том является постоянно резидентным или резервированным, или закодировано RETAIN в параметре VOLUME, или набор данных является передаваемым;

RETAIN — указывает, что том не должен быть демонтирован после последнего его использования в пункте задания;

порядковый-номер — указывает, с какого тома многоотомного набора данных необходимо начать обработку; отсутствие подпараметра определяет первый том;

число — максимальное число томов, требуемых для выходного набора данных; отсутствие подпараметра определяет запрос на один том;

SER = (регистрационный-номер,...) — регистрационные номера томов, на которых набор данных размещен или будет размещаться;

REF = имя-набора-данных — имя каталогизированного или передаваемого набора данных. Определяемый в операторе DD набор данных операционная система размещает на том же томе, на котором размещен указанный в этом подпараметре набор данных;

REF = *.имя-dd — том, который был указан в одном из предшествующих операторов DD того же пункта задания;

REF = *.имя-пункта.имя-dd — том, который был указан в операторе DD одного из предшествующих пунктов задания с указанными именами;

REF = *.имя-пункта.имя-пункта-процедуры.имя-dd — том, который был указан в операторе DD процедуры, вызванной в одном из предшествующих пунктов задания с указанными именами; отсутствие подпараметра определяет неспециальный запрос на том.

4.3.10. Параметр DCB

Параметр DCB предназначен для определения информации, заносимой в блок управления данными (DCB) во время выполнения программы. Формат параметра:

$$DCB = \left\{ \begin{array}{l} \text{(список-характеристик)} \\ \text{(имя-набора-данных} \\ \text{*.имя-dd} \\ \text{*.имя-пункта.имя-dd} \\ \text{*.имя-пункта.имя-процедуры.имя-dd} \end{array} \right\} [\text{список-характеристик}]$$

где список-характеристик — список ключевых подпараметров, обеспечивающих информацию, заносимую в блок управления данными (DCB) во время выполнения программы;

имя-набора-данных — имя каталогизированного набора данных, информация DCB которого должна быть скопирована. Информация копируется из метки набора данных;

*.имя-dd — один из предшествующих операторов DD, из которого копируется информация DCB;

*.имя-пункта.имя-dd — оператор DD в одном из предшествующих пунктов задания, из которого копируется информация DCB;

*.имя-пункта.имя-пункта-процедуры.имя-dd — оператор DD в процедуре, вызываемой в одном из предшествующих пунктов задания, из которого копируется информация DCB.

4.3.11. Параметр LABEL

Параметр LABEL позволяет определить тип метки набора данных, указать номер набора данных на магнитной ленте, если он не первый, указать парольную защиту набора данных, указать

время хранения набора данных на томе, перекрыть спецификации макрокоманды OPEN. Формат параметра:

$$\text{LABEL} = ([\text{номер}] \left[\begin{array}{l} \text{.SL} \\ \text{.SUL} \\ \text{.NSL} \\ \text{.NL} \\ \text{.BLP} \end{array} \right] \left[\begin{array}{l} \text{.PASSWORD} \\ \text{.NOPWREAD} \end{array} \right] \left[\begin{array}{l} \text{.IN} \\ \text{.OUT} \end{array} \right] \left[\begin{array}{l} \text{.} \\ \text{.} \end{array} \right] \left[\begin{array}{l} \text{EXPDT}=\text{ггддд} \\ \text{RETPD}=\text{пппп} \end{array} \right])$$

где номер — определяет порядковый номер набора данных на магнитной ленте. По умолчанию предполагается 1;

SL — стандартные метки;

SUL — стандартные метки и метки пользователя;

NSL — нестандартные метки на магнитной ленте;

NL — набор данных без меток на магнитной ленте;

BLP — операционная система не должна обрабатывать метки на магнитной ленте;

если второй подпараметр опущен, то предполагаются стандартные метки;

PASSWORD — парольная защита вновь создаваемого набора данных. Без указания пароля набор данных нельзя читать, изменять содержимое, расширять или удалять;

NOPWREAD — набор данных можно читать без указания пароля. Однако для изменения содержимого, расширения или удаления набора данных требуется указание пароля. Отсутствие подпараметра означает отсутствие парольной защиты;

IN — набор данных должен быть использован только для ввода (вместо метода обработки, указанного в макрокоманде OPEN);

OUT — набор данных должен быть использован только для вывода (вместо метода обработки, указанного в макрокоманде OPEN); отсутствие подпараметра означает сохранение метода обработки, указанного в макрокоманде OPEN;

EXPDT=ггддд — дата, до которой набор данных не может быть удален или перекрыт другим набором данных; гг — год, дд — порядковый номер дня в году;

RETPD=пппп — период времени в днях, в течение которого набор данных не может быть удален или перекрыт другим набором данных. Отсутствие подпараметра означает, что набор данных может быть удален или перекрыт другим набором данных в любое время.

4.3.12. Параметр DISP

Параметр DISP описывает состояние набора данных, а также определяет действия, которые необходимо выполнить с набором данных после нормального или аварийного завершения пункта задания. Первый подпараметр описывает состояние, второй — диспозицию после нормального завершения, третий — диспозицию по-

сле аварийного завершения (условную диспозицию). Формат параметра:

$$\text{DISP} = \left(\begin{bmatrix} \text{NEW} \\ \text{OLD} \\ \text{MOD} \\ \text{SHR} \end{bmatrix} \begin{bmatrix} \text{,KEEP} \\ \text{,CATLG} \\ \text{,UNCATLG} \\ \text{,PASS} \\ \text{,DELETE} \end{bmatrix} \right) \begin{bmatrix} \text{,KEEP} \\ \text{,CATLG} \\ \text{,UNCATLG} \\ \text{,DELETE} \end{bmatrix}$$

где NEW — набор данных создается в текущем пункте задания;

OLD — набор данных существует перед выполнением текущего пункта задания;

MOD — набор данных модифицируется в текущем пункте задания. Механизм считывания-записи должен быть установлен после последней записи в наборе данных. Если набор данных отсутствует, то предполагается, что он создается в текущем пункте задания;

SHR — набор данных существует перед выполнением текущего пункта задания и может быть совместно использован (только для чтения) другими заданиями;

KEEP — набор данных необходимо сохранить на томе после завершения пункта задания;

CATLG — набор данных необходимо сохранить на томе и каталогизировать после завершения пункта задания;

UNCATLG — набор данных необходимо сохранить на томе, но информацию о нем удалить из каталога после завершения пункта задания;

PASS — набор данных передается для использования последующим пунктом того же задания;

DELETE — набор данных необходимо удалить из тома после завершения пункта задания (пространство, занятое набором данных на томе, считается свободным). При этом для томов магнитных дисков информация о наборе данных удаляется из оглавления тома; на томах магнитных лент никаких изменений не производится, и окончательное решение об удалении набора данных принимает оператор.

Состояние по умолчанию предполагается NEW. Диспозиция по умолчанию выбирается таким образом, чтобы состояние набора данных после обработки совпадало с состоянием до обработки (существующие наборы данных сохраняются, новые — удаляются и т. д.). Если набор данных создается и удаляется в пункте задания, параметр DISP можно не кодировать.

4.3.13. Параметр SPACE

Параметр SPACE предназначен для распределения памяти на томах магнитных дисков. Формат параметра:

$$\left\{ \begin{array}{l} \text{SPACE} = (\text{единица}, (\text{количество-1} [\text{приращение}] [\text{оглавление}] [\text{индекс-1}])) \\ \left\{ \begin{array}{l} [\text{,RLSE}] [\text{,CONTIG}] [\text{,ROUND}] \\ [\text{,MXIG}] \\ [\text{,ALX}] \end{array} \right\} \\ \text{SPACE} = (\text{ABSTR}, (\text{количество-2}, \text{адрес} [\text{оглавление}] [\text{индекс-2}])) \end{array} \right\}$$

где единица — единицы, в которых заказывается память: CYL — цилиндр, TRK — дорожка, число — средняя длина блока в байтах; количество-1 — количество памяти, выделяемое на томе;

приращение — количество памяти, выделяемое на томе в том случае, если набор данных превысил выделенное ранее количество памяти. Допускается до 16 участков (экстентов) на томе для набора данных. Если память заказывается в блоках, то для вычисления приращения используется максимальная длина блока, указанная в подпараметре BLKSIZE параметра DCB. Для многотомных наборов данных приращение может выделяться на следующих томах, но таким образом, чтобы на каждом томе было не более 16 участков;

оглавление — число 256-байтовых записей для оглавления библиотеки;

индекс-1 — количество цилиндров для индекса индексно-последовательного набора данных;

RLSE — неиспользованная в выделенном пространстве память после закрытия набора данных должна быть освобождена;

CONTIG — выделяемая память должна быть непрерывной;

MXIG — должно быть выделено наибольшее свободное пространство на томе, но оно должно быть равным или больше заказанного количества памяти;

ALX — необходимо выделить до пяти участков памяти, каждый из которых равен или больше заказанного количества памяти;

ROUND — заказанное в блоках количество памяти должно округляться до целого числа цилиндров и выделяться в границах цилиндра;

ABSTR — набор данных должен быть размещен в заданных адресах тома;

количество-2 — количество памяти в дорожках, выделяемое на томе;

адрес — адрес первой дорожки участка, выделяемого на томе;

индекс-2 — количество дорожек для индекса индексно-последовательного набора данных.

Первая форма параметра SPACE предполагает автоматическое распределение памяти на томах магнитных дисков и является предпочтительной. Вторая форма параметра SPACE позволяет программисту самому выделять память на томе магнитных дисков и рекомендуется только для наборов данных, зависящих от местоположения. Если набору данных распределяется устройство, отличное от магнитных дисков, параметр SPACE игнорируется.

4.3.14. Параметр SPLIT

Параметр SPLIT предназначен для расположения двух или более новых наборов данных на разделенных цилиндрах. Формат параметра:

$$\text{SPLIT} = \left\{ \begin{array}{l} (n1, \text{CYL}, (\text{количество } [\text{приращение}])) \\ n2 \\ \%1, \text{длина-блока}, (\text{количество } [\text{приращение}])) \\ \%2 \end{array} \right\}$$

где $n1$ — количество дорожек на цилиндре, распределенных для первого набора данных;

CYL — указывает, что память заказывается в цилиндрах;

количество — количество памяти, выделяемое для набора данных;

приращение — количество памяти, выделяемое на томе в том случае, если набор данных превысил выделенное ранее количество памяти;

$n2$ — количество дорожек на цилиндре, распределенных для второго и следующих наборов данных;

$\%1$ — процент дорожек на цилиндре, распределенных для первого набора данных;

длина-блока — указывает, что память заказывается в блоках (указывается средняя длина блока);

$\%2$ — процент дорожек на цилиндре, распределенных для второго и следующих наборов данных.

4.3.15. Параметр SUBALLOC

Параметр SUBALLOC предназначен для размещения нескольких наборов данных на одном и том же томе в определенной последовательности в непрерывном участке памяти. Для этого сначала определяется первый набор данных с помощью параметра SPACE таким образом, чтобы выделить достаточно памяти для всех наборов данных. После этого для каждого следующего набора данных кодируются операторы DD с параметром SUBALLOC. Формат параметра:

$$\text{SUBALLOC} = (\text{единица}, (\text{количество-1 } [\text{,приращение}] [\text{,оглавление}]))$$

$$\left\{ \begin{array}{l} \text{,имя-dd} \\ \text{,имя-пункта.имя-dd} \\ \text{,имя-пункта.имя-пункта-процедуры.имя-dd} \end{array} \right\}$$

где подпараметры единица, количество-1, приращение и оглавление имеют то же назначение, что и в параметре SPACE;

имя-dd — определяет, что выделение памяти должно производиться из того пространства, которое указано в одном из предшествующих операторов DD того же пункта задания;

имя-пункта.имя-dd — указывает, что выделение памяти должно производиться из того пространства, которое определено в

операторе DD предшествующего пункта задания, имена которых указаны;

имя-пункта.имя-пункта-процедуры.имя-dd — указывает, что выделение памяти должно производиться из того пространства, которое определено в операторе DD пункта процедуры, вызываемой в предшествующем пункте задания, имена которых указаны.

4.3.16. Параметр SEP

Параметр SEP предназначен для запроса на разделение каналов для наборов данных, определенных в пункте задания. Формат параметра:

SEP = (имя-dd,...)

где имя-dd — имя предшествующего оператора DD того же пункта задания. Допускается до восьми операторов DD. Кодирование параметра SEP означает, что желательно, чтобы доступ к набору данных, определяемому текущим оператором DD, производился через канал, отличный от каналов, через которые производится доступ к наборам данных, определяемым операторами DD, указанными в списке.

4.3.17. Параметр AFF

Параметр AFF предназначен для указания общего запроса на разделение каналов. Формат параметра:

AFF=имя-dd

где имя-dd — имя оператора DD того же самого пункта задания, в котором указан запрос на разделение каналов. Текущий оператор DD должен иметь такой же запрос на разделение каналов.

4.3.18. Параметр OUTLIM

Параметр OUTLIM позволяет установить предел для числа логических записей, выводимых в выходной набор данных через выходной поток. Формат параметра:

OUTLIM=число

где число — максимальное число логических записей, которые могут быть включены в выходной набор данных.

4.3.19. Параметр DLM

Параметр DLM позволяет определить новый ограничитель данных во входном потоке, отличный от /*. Определив новый ограничитель, можно вводить в качестве данных стандартный ограничитель. Формат параметра:

DLM=ограничитель

где ограничитель — два знака, определяющие новый ограничитель.

4.3.20. Параметр QNAME

Параметр QNAME дает возможность доступа к сообщениям, получаемым через общий телекоммуникационный метод доступа для обработки в программе. Формат параметра:

QNAME=имя

где имя — имя макрокоманды TPROCESS, определяющей очередь пункта назначения сообщений, которые будут обрабатываться в программе. Имя содержит от 1 до 8 алфавитно-цифровых символов, причем первым должна быть буква.

4.3.21. Параметр TERM

Параметр TERM предписывает операционной системе осуществить ввод и вывод данных, используя абонентский пункт системы разделения времени. Формат параметра:

TERM=TS

4.3.22. Параметр COPIES

Параметр COPIES позволяет запросить несколько копий выходных наборов данных. Формат параметра:

COPIES=nnn

где nnn — количество копий набора данных SYSOUT. Параметр можно кодировать только с параметром SYSOUT. По умолчанию предполагается одна копия.

4.3.23. Параметр FCB

Параметр FCB определяет образ формата страниц, который должен быть использован для печати выходного набора данных на печатающем устройстве, имеющем средство управления форматом страниц (EC-7039). Формат параметра:

FCB=(идентификатор [,ALIGN]
[,VERIFY])

где идентификатор — код, идентифицирующий образ формата страниц, который загружается в буфер управления форматом страниц; содержит от 1 до 4 алфавитно-цифровых символов, первым из которых должна быть буква;

ALIGN — запрос оператору на проверку выравнивания форм печатающего устройства перед печатью набора данных;

VERIFY — запрос оператору на визуальную проверку требуемого образа формата страниц, загруженного в буфер печатающего устройства. Оператору дается также возможность выравнивать формы печатающего устройства.

4.3.24. Параметр UCS

Параметр UCS определяет образ набора знаков, используемый для вывода набора данных на печатающее устройство (ЕС-7039).
Формат оператора:

$$\text{UCS} = (\text{идентификатор} \quad [, \text{FOLD}] [, \text{VERIFY}])$$

где идентификатор — от 1 до 4 символов, идентифицирующих специальный набор знаков, который необходимо использовать для печати набора данных;

FOLD — вызывает подавление бит 0 и 1 при сравнении буфера печатающего устройства с буфером печатаемой строки. Эта возможность позволяет прописные буквы печатать как строчные, если будет установлена соответствующая цепочка;

VERIFY — запрос оператору на визуальную проверку используемого образа набора знаков.

4.4. Оператор команды

Оператор команды осуществляет ввод оператора для выполнения команды через входной поток заданий. Формат оператора:

$$// \text{ команда операнды} \quad [\text{комментарии}]$$

где команда — идентификатор вводимой команды. Допускается ввод через входной поток следующих команд: CANCEL, DISPLAY, HOLD, LOG, MODIFY, MOUNT, RELEASE, REPLY, RESET, SET, START, STOP, UNLOAD, VARY, WRITELOG;

операнды — операнды команды;

комментарии — могут включать любую информацию. Продолжение оператора на вторую строку не допускается. Оператор команды может появиться перед операторами JOB, EXEC, пустыми и другими операторами команды.

4.5. Оператор комментариев

Оператор комментариев используется для записи любой информации, предназначенной для комментирования. Формат оператора:

$$/* \text{ комментарии}$$

4.6. Ограничительный оператор

Ограничительный оператор предназначен для указания конца набора данных во входном потоке, начало которого указывается оператором DD * или DD DATA. Формат оператора:

$$/* \text{ [комментарии]}$$

Продолжение оператора на вторую строку не допускается.

4.7. Пустой оператор

Пустой оператор отмечает завершение управляющих операторов задания. Формат оператора:

//

Использование пустого оператора не является обязательным.

4.8. Оператор процедуры (PROC)

Оператор PROC — первый оператор процедуры во входном потоке. Он может быть также первым в каталогизированной процедуре. Оператор используется для присваивания стандартных значений (по умолчанию) символическим параметрам процедуры. Формат оператора:

//[имя] PROC [операнды] [комментарии]

где имя — имя, содержащее от 1 до 8 алфавитно-цифровых символов, первым из которых должна быть буква. Для процедур во входном потоке — это имя процедуры. Для каталогизированных процедур имя может отсутствовать;

операнды — в поле операндов следует кодировать список символических параметров и их стандартные значения в следующем формате:

{ [символический параметр=значение
символический параметр=] , . . . }

где первый формат присваивает некоторое значение по умолчанию символического параметра; второй формат присваивает пустое стандартное значение символического параметра; поле операндов является необязательным для процедур во входном потоке и обязательным для каталогизированных процедур;

комментарии — любая информация.

Оператор PROC обязателен в процедурах во входном потоке и не обязателен в каталогизированных процедурах. В последнем случае он требуется только для присваивания стандартных значений символическим параметрам, причем имя оператора в этом случае не является именем процедуры.

Стандартные значения символических параметров могут быть перекрыты путем присваивания новых значений этим символическим параметрам в операторе EXEC, вызывающем процедуру.

4.9. Оператор конца процедуры (PEND)

Оператор PEND отмечает конец процедуры во входном потоке. Формат оператора:

//имя PEND — [комментарии]

где имя — имя, содержащее от 1 до 8 алфавитно-цифровых символов, первым из которых должна быть буква;
комментарии — любая информация.
Продолжение оператора на вторую строку не допускается.

4.10. Процедуры

Процедура состоит из одного или нескольких пунктов процедуры. Пункт процедуры есть оператор EXEC, за которым следуют связанные с ним операторы DD. Процедура не может содержать:

- операторов EXEC, вызывающих каталогизированную процедуру;

- операторов JOB, ограничительных и пустых операторов;

- операторов DD с именем JOBLIB;

- операторов DD с операндами * или DATA.

Каталогизированная процедура представляет собой процедуру, имеющую имя и помещенную в библиотеку процедур. Системная библиотека процедур имеет имя SYS1.PROCLIB. Могут быть добавлены другие библиотеки процедур. Каталогизированная процедура может начинаться с оператора PROC. Имя каталогизированной процедуры определяется именем процедуры в библиотеке процедур.

Процедура во входном потоке представляет собой процедуру, начинающуюся с оператора PROC и завершающуюся оператором PEND. Имя процедуры во входном потоке определяется именем оператора PROC. Процедуру во входном потоке можно поместить после оператора JOB или операторов DD с именами JOBLIB или SYSCHK. Процедуру во входном потоке нельзя помещать перед оператором JOB или после оператора EXEC, вызывающего процедуру. Нежелательно помещать процедуру во входном потоке после оператора SYSIN DD *.

Символический параметр процедуры представляет собой имя, которому предшествует амперсанд; причем имя должно начинаться с буквы и содержать от 1 до 7 алфавитно-цифровых символов.

Символические параметры используются в процедуре для ее модификации. Они могут появляться в поле операндов операторов процедуры. Конец символического параметра отмечается точкой, если затем следует буква, цифра, левая скобка или точка.

Стандартные значения символических параметров присваиваются в операторе PROC. Значения символических параметров присваиваются в операторе EXEC, вызывающем процедуру (в операторах PROC и EXEC символические параметры употребляются без амперсанда). Присваивание значений символическим параметрам является обязательным для параметров, не имеющих стандартного значения, и необязательным в противном случае. Формат присваивания значений символическим параметрам в операторе EXEC совпадает с форматом присваивания стандартных значений символическим параметрам в операторе PROC.

Замещение, добавление или удаление параметров оператора EXEC процедуры производятся путем использования в операторе EXEC, вызывающем процедуру, запроса в следующем формате:

параметр. имя-пункта-процедуры = [значение]

где параметр — мнемоника замещаемого, добавляемого или удаляемого параметра;

имя-пункта-процедуры — имя пункта процедуры, в котором производится замещение, добавление или удаление параметра;

значение — либо новое значение замещаемого параметра, либо значение добавляемого параметра. В случае удаления параметра значение должно отсутствовать.

Замещение, добавление или удаление параметров оператора DD процедуры производятся путем включения в пункт задания, вызывающего процедуру, оператора DD следующего формата:

```
//имя-пункта-процедуры.имя-оператора-dd DD
{ параметр = [значение]
  взаимно-исключающий-параметр = значение }
```

где имя-пункта-процедуры — имя пункта процедуры, в котором производится замещение, добавление или удаление параметров оператора DD;

имя-оператора-dd — имя оператора DD пункта процедуры, в котором производятся замещение, добавление или удаление параметров;

параметр = [значение] — замещаемый, добавляемый или удаляемый параметр и либо новое значение замещаемого параметра, либо значение добавляемого параметра. В случае удаления параметра значение должно отсутствовать;

взаимно-исключающий-параметр-значение — используется для замещения параметра оператора DD процедуры путем задания значения взаимно исключающему параметру.

Для замещения отдельных подпараметров параметра DCB не требуется кодировать весь параметр DCB. Необходимо кодировать только замещаемые подпараметры либо взаимно исключающие подпараметры. Неизменяемые подпараметры повторять не требуется. В этом случае параметр DCB в рассматриваемом операторе DD должен иметь следующий формат:

$$DCB = \left\{ \begin{array}{l} \text{замещаемый-подпараметр} \\ \text{взаимно-исключающий-подпараметр} \end{array} \right\} = \text{значение, ...}$$

Если нужно заместить операторы DD процедуры для сцепленных наборов данных, следует кодировать замещающие операторы в том же порядке, в котором расположены замещаемые операторы, причем если какой-либо оператор не подлежит замещению, то поле операндов в замещающем операторе должно быть пустым. Второй и последующие операторы не должны иметь имени.

Операторы DD, которые производят замещение, добавление или удаление параметров операторов DD процедуры, следует распо-

лазать в том же порядке, что и соответствующие операторы DD в процедуре.

Добавление операторов DD к процедуре производится путем включения в пункт задания, вызывающего процедуру, оператора DD следующего формата:

// имя-пункта-процедуры.имя-dd DD параметры

где имя-пункта-процедуры — имя пункта процедуры, к которому производится добавление оператора DD;

имя-dd — имя добавляемого оператора DD (не должно совпадать с именами уже имеющихся в пункте процедуры операторов DD);

параметры — список параметров добавляемого оператора DD.

Добавляемые операторы DD следует располагать вслед за операторами DD, которые производят замещение, добавление или удаление параметров операторов DD процедуры.

Макрокоманды супервизора

Макрокоманды супервизора ОС ЕС являются средством программирования на языке Ассемблера запросов на выполнение различных функций супервизора. Запись макрокоманды состоит из поля метки, поля операции, поля операндов и поля комментариев.

Все поля разделяются между собой не менее чем одним пробелом. Операнды должны отделяться друг от друга запятыми. Описания макрокоманды, приводимые ниже, группируются по функциям супервизора.

5.1. Управление задачами

5.1.1. Макрокоманда ATTACH

Макрокоманда ATTACH предназначена для создания новой задачи, которая является подзадачей задачи, выдавшей ATTACH. Адрес TCB созданной задачи помещается в регистр 1. Новая задача создается только в режимах MVT, SVS и MFT с подзадачами. В режиме MFT без подзадач макрокоманда ATTACH выполняется как макрокоманда LINK. Формат макрокоманды:

```
[метка] ATTACH { EP=имя
                  EPLOC=адрес-имени
                  DE=адрес-элемента-списка } [,DCB=адрес-dcb]
                  [,LPMOD=число] [,DPMOD={число
                                           (R)}]
                  [,PARAM=(адрес,...) [,VL=1]]
                  [,ECB=адрес-ecb] [,ETXR=адрес-программы-выхода]
                  [ , { GSPV=число
                        GSPL=адрес-списка } ] [ , { SHSPV-число
                        SHSPL=адрес-списка } ]
                  [,SZERO={ YES
                           NO } ]
                  [,STAI=(адрес-программы-выхода [,адрес-списка-параметров])]
                  [,PURGE={ QUIESCE
                           HALT
                           NONE } ] [,ASYNCH={ YES
                                                NO } ]
                  [,TASKLIB=адрес-dcb]
```


где EP — имя точки входа в загрузочном модуле, в которую должно передаваться управление;

EPLOC — адрес поля, где хранится имя точки входа;

DE — адрес поля имени точки входа в элементе списка, построенного макрокомандой BLDL;

DCB — адрес блока управления данными (DCB) для библиотеки, содержащей вызываемую программу. Адрес DCB для общей библиотеки и библиотек задания и пункта задания обозначается нулем или пропуском этого операнда;

LPMOD — число без знака, которое вычитается из значения граничного приоритета порождающей задачи, в результате чего получается значение граничного приоритета новой задачи. Если операнд опущен, граничный приоритет новой задачи совпадает с граничным приоритетом порождающей задачи;

DPMOD — число со знаком, которое складывается со значением текущего приоритета порождающей задачи, в результате чего получается значение текущего приоритета новой задачи. Если операнд опущен, то текущим приоритетом новой задачи будет минимальное из двух значений: граничного приоритета новой задачи и текущего приоритета порождающей задачи;

PARAM — один или несколько адресных параметров, передаваемых вызываемой программе в виде списка параметров;

VL=1 — означает, что старший разряд последнего адресного параметра в списке параметров устанавливается в 1;

ECB — адрес блока управления событием, который используется управляющей программой для указания завершения новой задачи;

ETXR — адрес программы выхода, которой будет передано управление после того, как новая задача завершится (нормально или аварийно);

GSPV — номер подпула основной памяти, передаваемого новой задаче в монопольное использование;

GSPL — адрес списка номеров подпулов основной памяти, передаваемых новой задаче в монопольное использование;

SHSPV — номер подпула основной памяти, передаваемого новой задаче в совместное с порождающей задачей использование;

SHSPL — адрес списка номеров подпулов основной памяти, передаваемых новой задаче в совместное с порождающей задачей использование;

SZERO — определяет, будет ли новая задача совместно с порождающей задачей использовать нулевой подпул (YES — будет, NO — нет). По умолчанию подразумевается YES;

STAI — адрес программы выхода и адрес списка параметров в случае аварийного завершения новой задачи;

PURGE — определяет порядок выполнения операций ввода-вывода при переходе на программу выхода STAI. QUIESCE — означает, что неудовлетворенные запросы на операции ввода-вывода сохраняются при переходе к программе выхода. HALT — означает, что все неудовлетворенные запросы на операции ввода-вывода не

сохраняются при переходе к программе выхода. NONE — означает, что обработка операций ввода-вывода продолжается нормально при переходе к программе выхода; по умолчанию — QUIESCE.

ASYNCH — определяет, разрешена ли обработка асинхронных прерываний при переходе на программу выхода STAI (YES — разрешена, NO — нет). По умолчанию подразумевается NO;

TASKLIB — адрес DCB библиотеки, которая будет просматриваться в поисках модуля, составляющего подзадачу, и тех модулей, к которым будет обращаться подзадача (библиотека задачи).

Операнды GSPV, GSPL и следующие за ними в приведенном формате используются только в режимах MVT и SVS.

Описательная форма макрокоманды ATTACH отличается от стандартной отсутствием операндов PARAM и VL и добавлением операнда SF=L. Остальные операнды необязательны.

Исполнительная форма макрокоманды ATTACH отличается от стандартной добавлением следующих операндов:

$$\left. \begin{array}{l} \text{MF} = (E, \left\{ \begin{array}{l} \text{адрес} \\ (1) \end{array} \right\}) \\ \text{SF} = (E, \left\{ \begin{array}{l} \text{адрес} \\ (15) \end{array} \right\}) \\ \text{MF} = (E, \left\{ \begin{array}{l} \text{адрес} \\ (1) \end{array} \right\}) \quad , \quad \text{SF} = (E, \left\{ \begin{array}{l} \text{адрес} \\ (15) \end{array} \right\}) \end{array} \right\}$$

где MF — указывает на исполнительную форму макрокоманды, использующую вынесенный список параметров пользователя, адрес этого списка указывается либо в операнде, либо загружается в регистр 1;

SF — указывает на исполнительную форму макрокоманды, использующую вынесенный список параметров управляющей программы, адрес этого списка указывается либо в операнде, либо загружается в регистр 15.

Остальные операнды необязательны.

5.1.2. Макрокоманда DETACH

Макрокоманда DETACH предназначена для удаления задачи из операционной системы. Макрокоманда DETACH обязательно выдается для подзадач, образованных макрокомандой ATTACH с операндами ЕСВ или ETRR. Для подзадач, образованных макрокомандой ATTACH без этих операндов, макрокоманду DETACH выдавать не требуется. В последнем случае удаление подзадачи происходит автоматически после ее завершения.

Перед завершением порождающей задачи все ее подзадачи должны быть удалены. В противном случае порождающая задача вместе с подзадачами завершается аварийно. В режиме MFT без подзадач макрокоманда DETACH рассматривается как пустая операция.

Формат макрокоманды:

$$[\text{метка}] \quad \text{DETACH} \quad \text{адрес-адреса-тcb} [, \text{STAE} = \left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\}]$$

где адрес-адреса-*tcb* — адрес полного слова в границах слова, содержащего адрес TCB удаляемой задачи;

STAE — указывает, получит ли управление программа выхода, определенная макрокомандой STAE при аварийном завершении подзадачи в момент ее удаления (YES — получит, NO — нет). По умолчанию подразумевается NO.

Операнд STAE используется только в режимах MVT и SVS.

5.1.3. Макрокоманда СНАР

Макрокоманда **CHAP** предназначена для изменения текущего приоритета задачи или любой ее подзадачи. Макрокоманда **CHAP** используется также для увеличения граничного приоритета подзадач (не выше граничного приоритета порождающей задачи). В режиме **MFT** без подзадач макрокоманда **CHAP** рассматривается как пустая операция. Формат макрокоманды:

[метка] CHAP величина-изменения-приоритета [,адрес-адреса-tcb
 : 'S']

где величина-изменения-приоритета — величина со знаком, которая прибавляется к значению приоритета, в результате чего получается новое значение текущего приоритета;

адрес-адреса-`tsb` — адрес полного слова в границах слова, содержащий адрес ТСВ задачи, приоритет которой меняется.

'S' — указывает на изменение приоритета активной задачи. Если в операнде указан нулевой адрес или операнд опущен, то подразумевается 'S'.

5.1.4. Макрокоманда EXTRACT

Макрокоманда EXTRACT предназначена для получения информации из ряда полей TCB активной задачи или ее подзадач. Формат макрокоманды:

[метка] EXTRACT адрес-поля-ответа [адрес-адреса-tcb], FIELDS = (коды)

где адрес-поля-ответа — адрес в основной памяти одного или нескольких слов (в границах слова) по числу полей, указанных в операнде FIELDS (если указано FIELDS = (ALL), требуется семь полных слов);

адрес-адреса-*tsb* — адрес полного слова в границах слова, содержащий адрес ТСВ задачи, из которого извлекается информация;

'S' — указывает на ТСВ активной задачи. Если в операнде указан нулевой адрес или операнд опущен, то подразумевается 'S';

FIELDS — один или несколько наборов символов, разделенных запятыми и определяющих следующие запрашиваемые поля TCB:

ALL — информация запрашивается из всех указанных ниже полей:

GRS — адрес области сохранения общих регистров задачи;

FRS — адрес области сохранения регистров с плавающей запятой задачи;

AETX — адрес программы выхода, обрабатывающий завершение задачи и указанный в макрокоманде ATTACH при образовании задачи;

PRI — граничный и текущий приоритет задачи;

CMC — код завершения задачи. Если задача не завершилась, то ноль;

TIOT — адрес таблицы ввода-вывода задачи.

В режиме MFT без подзадач можно указывать только значения ALL или TIOT. Если в этом режиме закодировано ALL, то выдается только адрес таблицы ввода-вывода задачи.

Описательная форма макрокоманды EXTRACT отличается от стандартной добавлением операнда MF=L. Остальные операнды необязательны.

Исполнительная форма макрокоманды EXTRACT отличается от стандартной добавлением операнда

$$MF = (E, \left\{ \begin{array}{l} \text{адрес} \\ (1) \end{array} \right\})$$

указывающего на исполнительную форму, использующую вынесенный список параметров управляющей программы; адрес этого списка указывается либо в операнде, либо загружается в регистр 1. Остальные операнды необязательны.

5.2. Синхронизация событий

5.2.1. Макрокоманда WAIT

Макрокоманда WAIT предназначена для указания о том, что выполнение активной программы может быть продолжено только при выполнении одного или нескольких указанных событий, каждое из которых представляется блоком управления событием. Перед использованием нулевой разряд блока управления событием должен быть установлен в ноль. Если заданное число указанных событий к моменту выполнения макрокоманды WAIT уже произошло, то выполнение активной задачи продолжается. В противном случае задача переходит в состояние ожидания. Формат макрокоманды:

$$[\text{метка}] \quad \text{WAIT} \quad [\text{число-событий},] \left\{ \begin{array}{l} \text{ECB} = \text{адрес} \\ \text{ECBLIST} = \text{адрес} \end{array} \right\}$$

где число-событий — любое положительное число не более 255, обозначающее число ожидаемых событий. Если операнд опущен, то предполагается 1. Если закодирован 0, то макрокоманда WAIT рассматривается как пустая операция. Число событий не должно превышать числа блоков управления событием;

ECB — адрес блока управления событием. Копируется, если указывается только одно событие;

ECBLIST — адрес области основной памяти, содержащей одно или несколько слов в границах слова, каждое из которых содержит адрес блока управления событием. Старший разряд в последнем слове должен быть равен 1. Число блоков управления событием должно быть больше указанного числа событий или равно ему.

5.2.2. Макрокоманда WAITR

Макрокоманда WAITR кодируется и выполняется точно так же, как макрокоманда WAIT.

5.2.3. Макрокоманда POST

Макрокоманда POST устанавливает совершение некоторого события, определенного блоком управления событием. При этом нулевой разряд блока устанавливается в 0, первый разряд — в 1, а в разряды 2—31 записывается код завершения. Формат макрокоманды:

[метка] POST адрес-есб [код-завершения]

где адрес-есб — адрес блока управления событием;

код-завершения — число в интервале от 0 до $2^{24}-1$. Если код завершения не указан, то подразумевается 0.

5.3. Управление связями

5.3.1. Макрокоманда LINK

Макрокоманда LINK предназначена для передачи управления с возвратом в динамических последовательных структурах. После выполнения вызываемой программы управление возвращается команде, следующей за макрокомандой LINK. Формат макрокоманды:

[метка] LINK $\left\{ \begin{array}{l} EP = \text{имя} \\ EPLOC = \text{адрес-имени} \\ DE = \text{адрес-элемента-списка} \\ [PARAM = (\text{адрес}, \dots) \quad [VL = 1]] \quad [ID = \text{число}] \end{array} \right\} [DCB = \text{адрес-dcb}]$

где EP — имя точки входа в загрузочном модуле, в которую должно передаваться управление;

EPLOC — адрес поля, где хранится имя точки входа;

DE — адрес поля имени точки входа в элементе списка, построенного макрокомандой BLDL;

DCB — адрес блока управления данными (DCB) для библиотеки, содержащей вызываемую программу. Если операнд DCB опущен или приравнен 0 и макрокоманда LINK используется в задаче пункта задания, то поиск загрузочного модуля производится сначала в библиотеках операторов DD с метками STEPLIB или JOBLIB и, если его там нет, — в общей библиотеке. Если же мак-

рокоманда LINK используется в подзадаче, то поиск загрузочного модуля производится сначала в библиотеке задачи, указанной в макрокоманде ATTACH, образовавшей подзадачу.

PARAM — один или несколько адресных параметров, передаваемых вызываемой программе в виде списка параметров;

VL=1 — старший разряд последнего адресного параметра в списке параметров устанавливается в 1;

ID — число от 0 до $2^{16}-1$, идентифицирующее конкретное использование макрокоманды.

Описательная и исполнительная формы макрокоманды LINK образуются так же, как для макрокоманды ATTACH.

5.3.2. Макрокоманда XCTL

Макрокоманда XCTL предназначена для передачи управления без возврата в вызывающую программу в динамических последовательных структурах. Прежде чем выдается макрокоманда XCTL, регистры со 2-го по 12-й, область управления программными прерываниями и маска программы должны быть приведены в состояние, в котором они находились в момент, когда текущая программа получила управление. Формат макрокоманды:

$$[\text{метка}] \quad \text{XCTL} \quad [(\text{per1} \text{ } [\text{,per2}])], \left\{ \begin{array}{l} \text{EP}=\text{имя} \\ \text{EPLOC}=\text{адрес-имени} \\ \text{DE}=\text{адрес-элемента-списка} \end{array} \right\} [\text{,DCB}=\text{адрес}]$$

где per1, per2 — регистры от 2—12, которые восстанавливаются макрокомандой XCTL;

операнды EP, EPLOC, DE и DCB имеют то же назначение, что в макрокоманде LINK.

Описательная и исполнительная формы макрокоманды XCTL образуются так же, как для макрокоманды ATTACH.

5.3.3. Макрокоманда LOAD

Макрокоманда LOAD предназначена для загрузки программы, содержащей указанную точку входа в основную память, если в ней нет копии этой программы, пригодной для использования. Счетчик обращений к программе увеличивается на 1. Управление в точку входа не передается. Адрес точки входа загружается в регистр 0. Загрузочный модуль остается в основной памяти до тех пор, пока счетчик обращений не станет равным 0 после завершения задания или использования макрокоманды DELETE. Формат макрокоманды:

$$[\text{метка}] \quad \text{LOAD} \quad \left\{ \begin{array}{l} \text{EP}=\text{имя} \\ \text{EPLOC}=\text{адрес-имени} \\ \text{DE}=\text{адрес-элемента-списка} \end{array} \right\} [\text{,DCB}=\text{адрес-dcb}]$$

где операнды EP, EPLOC, DE и DCB имеют то же назначение, что и в макрокоманде LINK.

5.3.4. Макрокоманда DELETE

Макрокоманда DELETE уменьшает на 1 счетчик обращений для программы с указанной точкой входа. Когда счетчик обращения становится равным 0, основная память, занимаемая программой, освобождается. Таким образом эта макрокоманда выполняет функции, обратные функциям макрокоманды LOAD. Формат макрокоманды:

$$[\text{метка}] \text{ DELETE } \left\{ \begin{array}{l} \text{EP}=\text{имя} \\ \text{EPLOC}=\text{адрес-имени} \\ \text{DE}=\text{адрес-элемента-списка} \end{array} \right\}$$

где операнды EP, EPLOC и DE имеют то же назначение, что и в макрокоманде LINK.

Если выполнение макрокоманды завершилось успешно, регистр 15 содержит 0. Если задача, выдавшая макрокоманду DELETE, не выдавала предварительно макрокоманды LOAD или если счетчик обращений перед этим уже стал равен 0, регистр 15 содержит 4.

5.3.5. Макрокоманда IDENTIFY

Макрокоманда IDENTIFY используется для добавления точки входа к копии загрузочного модуля, находящегося в текущий момент в основной памяти. В режимах MVT и SVS добавленная точка входа может использоваться макрокомандами ATTACH, LINK, LOAD, DELETE или XCTL; в режиме MFT — только макрокомандой ATTACH. В режиме MFT макрокоманду IDENTIFY нельзя выдавать в программе, в которую вошли по добавленной точке входа. Формат макрокоманды:

$$[\text{метка}] \text{ IDENTIFY } \left\{ \begin{array}{l} \text{EP}=\text{имя} \\ \text{EPLOC}=\text{адрес-имени} \end{array} \right\}, \text{ENTRY-адрес-точки-входа}$$

где EP — имя точки входа;

EPLOC — адрес имени точки входа;

ENTRY — адрес в основной памяти добавляемой точки входа.

О результатах выполнения макрокоманды IDENTIFY при возврате управления можно судить по содержимому регистра 15.

5.3.6. Макрокоманда SAVE

Макрокоманда SAVE производит запоминание содержимого указанных регистров в области сохранения, адрес которой находится в регистре 13. Макрокоманда SAVE может находиться только в точках входа программы. Формат макрокоманды:

$$[\text{метка}] \text{ SAVE } (\text{per1}[\text{,per2}]), [\text{T}][\text{,идентификатор}]$$

где per1 и per2 — указывают диапазон регистров, содержимое которых необходимо запомнить в области сохранения (регистры указываются в порядке их запоминания: 15, 14, 0, 1, ..., 12);

Т — указывает, что содержимое регистров 14 и 15 необходимо запомнить (используется только в том случае, если эти регистры не были указаны в первом операнде);

идентификатор — идентификатор до 70 символов или *. Если закодирован *, то в качестве идентификатора используется метка макрокоманды SAVE, а если ее нет, то имя программной секции (имя соответствующей команды CSECT или START). Идентификатор идентифицирует конкретную макрокоманду SAVE и соответствующую точку входа в программу.

5.3.7. Макрокоманда RETURN

Макрокоманда RETURN предназначена для возврата управления вызывающей программе и передаче ей кода завершения. Возврат управления осуществляется командой безусловной передачи управления по регистру 14. Формат макрокоманды:

[метка] RETURN [(per1[,per2))][,T][,RC=число
[,RC=(15)]

где per1, per2 — указывают диапазон регистров, содержимое которых должно быть восстановлено из области сохранения, адрес которой находится в регистре 13 (регистры указываются в порядке запоминания регистров: 14, 15, 0, 1, ..., 12);

T — отмечает область сохранения, как принадлежащую программе, возвратившей управление (код X'FF' в старшем байте четвертого слова области сохранения);

RC — код возврата, представляющий собой число в диапазоне от 0 до 4095, которое засылается в регистр 15. Если перед выполнением макрокоманды код возврата уже находится в регистре 15, следует кодировать: RC=(15).

5.3.8. Макрокоманда CALL

Макрокоманда CALL используется для передачи управления программной секции командой передачи управления по регистру 15 с загрузкой адреса возврата в регистр 14. Управляющая программа не участвует в передаче управления. Формат макрокоманды:

[метка] CALL {имя-точки-входа}[, (адрес,...)[,VL]][,ID=число]
(15)

где имя точки входа — имя точки входа, в которую передается управление. В этом случае создается константа типа V, в которой имя точки входа используется в качестве операнда. Загрузочный модуль, содержащий указанную точку входа, должен быть включен Редактором связей в тот же загрузочный модуль, в котором находится макрокоманда CALL;

(15) — перед выполнением макрокоманды CALL регистр 15 уже содержит адрес точки входа, в которую передается управление; при этом загрузочный модуль, которому передается управление, должен уже находиться в основной памяти;

(адрес,...) — один или несколько адресных параметров, передаваемых вызываемой программе в виде списка параметров;

VL — означает, что старший разряд последнего адресного параметра в списке параметров устанавливается в 1;

ID — число от 0 до $2^{16}-1$, идентифицирующее конкретное использование макрокоманды.

Описательная форма макрокоманды CALL отличается от стандартной добавлением операнда MF=L. Первый операнд стандартной формы, указывающий точку входа, должен отсутствовать.

Исполнительная форма макрокоманды CALL отличается от стандартной добавлением операнда

$$MF = (E, \left\{ \begin{array}{l} \text{адрес} \\ (1) \end{array} \right\})$$

указывающего на исполнительную форму, использующую вынесенный список параметров; адрес этого списка указывается либо в операнде, либо загружается в регистр 1.

5.4. Управление основной памятью

5.4.1. Макрокоманда GETMAIN

Макрокоманда GETMAIN используется для запроса одной или нескольких областей основной памяти. Каждая область выравнивается по границе двойного слова и при выделении не очищается. Размер запрашиваемой области указывается в байтах. Количество запрашиваемых байт должно быть кратно 8. Если в макрокоманде указано количество байт, не кратное 8, то управляющая программа принимает в качестве количества запрашиваемых байт ближайшее большее число, кратное 8.

Существуют 4 типа запросов и соответственно 4 формата макрокоманды, которые различаются первой буквой первого операнда: регистровый (R), элементный (E), списковый (L), переменный (V). Запросы могут быть двух видов, различающихся второй буквой первого операнда: условный (C) и безусловный (U). При выполнении условных запросов регистр 15 содержит один из следующих кодов возврата: 0 — если запрос удовлетворен, 4 — если запрос не удовлетворен. Затем управление передается команде, следующей за макрокомандой GETMAIN. При выполнении безусловных запросов в случае, если запрос не удовлетворен, возникает аварийное завершение задачи; в противном случае управление передается команде, следующей за макрокомандой GETMAIN. Регистровый запрос всегда является безусловным. Употребление макрокоманды GETMAIN регистрового запроса не нарушает реентерабельность программы. Запрошенные и выделенные области основной памяти освобождаются либо после завершения задачи пункта задания, либо по макрокоманде FREEMAIN.

Регистровый формат макрокоманды GETMAIN:

[метка] GETMAIN R, {LV=число [,SP=число] }
 { LV=(0) }

где LV=число — размер запрашиваемой области основной памяти в байтах;

SP — номер подпула, из которого должна быть выделена область основной памяти;

LV=(0) — регистр 0 содержит в трех младших байтах размер запрашиваемой области основной памяти в байтах, а в старшем байте — номер подпула, из которого должна быть выделена область основной памяти. Адрес выделенной области основной памяти управляющая программа помещает в регистр 1.

Элементный формат макрокоманды GETMAIN:

[метка] GETMAIN {EC } , LV=число, A=адрес [, SP=число]
 {EU }

где LV — размер запрашиваемой области основной памяти в байтах;

A — адрес слова, в которое управляющая программа помещает адрес выделенной области основной памяти;

SP — номер подпула, из которого должна быть выделена область основной памяти.

Списковый формат макрокоманды GETMAIN:

[метка] GETMAIN {LC } , LA=адрес, A=адрес [, SP=число]
 {LU }

где LA — адрес списка полных слов, каждое из которых содержит размер запрашиваемой области основной памяти в байтах. Старший разряд последнего слова в списке содержит 1. Количество слов в списке соответствует количеству запрашиваемых областей;

A — адрес списка полных слов, в которые управляющая программа помещает адреса выделенных областей основной памяти. Количество слов в данном списке равно количеству слов в списке LA;

SP — номер подпула, из которого должны быть выделены области основной памяти.

Списковый запрос считается удовлетворенным, если выделяют все запрашиваемые области основной памяти. Не допускается использование спискового запроса в режиме MFT.

Переменный формат макрокоманды GETMAIN:

[метка] GETMAIN {VC } , LA=адрес, A=адрес [, SP=число]
 {VU }

где LA — адрес списка из двух слов, первое из которых содержит минимальный размер запрашиваемой области основной памяти в байтах, а второе — максимальный;

A — адрес списка из двух слов, в первое из которых управляющая программа помещает адрес выделенной области основной памяти, а во второе — фактический размер выделенной области в байтах;

SP — номер подпула, из которого должна быть выделена область основной памяти.

Для всех форматов макрокоманды GETMAIN номер подпула в режиме MFT не используется. Если операнд SP опущен, то по умолчанию подразумевается нулевой подпул.

Описательная и исполнительная формы макрокоманды GETMAIN элементного, спискового и переменного форматов образуются так же, как для макрокоманды EXTRACT.

5.4.2. Макрокоманда FREEMAIN

Макрокоманда FREEMAIN освобождает одну или несколько областей основной памяти либо целый подпул, предварительно запрошенные по макрокоманде GETMAIN. Так же, как для макрокоманды GETMAIN, существуют аналогичные 4 формата макрокоманды FREEMAIN.

Регистровый формат макрокоманды FREEMAIN для освобождения подпула:

$$[\text{метка}] \text{ FREEMAIN } R, SP = \begin{cases} \text{число} \\ (0) \end{cases}$$

где SP — номер освобождаемого подпула. (0) означает, что номер находится в старшем байте регистра 0.

Регистровый формат макрокоманды FREEMAIN для освобождения области основной памяти:

$$[\text{метка}] \text{ FREEMAIN } R, \begin{cases} LV = \text{число}, A = \begin{cases} \text{адрес} \\ (1) \end{cases} [SP = \text{число}] \\ LV = (0), A = \begin{cases} \text{адрес} \\ (1) \end{cases} \end{cases}$$

где LV=число — размер освобождаемой области основной памяти в байтах;

A — адрес слова, в котором находится адрес освобождаемой области основной памяти;

SP — номер подпула, в котором освобождается область основной памяти;

LV=(0) — регистр 0 содержит в трех младших байтах размер освобождаемой области основной памяти, а в старшем байте — номер подпула, в котором освобождается область основной памяти.

Элементный формат макрокоманды FREEMAIN:

$$[\text{метка}] \text{ FREEMAIN } E, LV = \text{число}, A = \text{адрес}, [SP = \text{число}]$$

где LV — размер освобождаемой области основной памяти в байтах;

A — адрес слова, в котором находится адрес освобождаемой области основной памяти;

SP — номер подпула, в котором освобождается область основной памяти.

Списковый формат макрокоманды FREEMAIN:

$$[\text{метка}] \text{ FREEMAIN } L, LA = \text{адрес}, A = \text{адрес}, [SP = \text{число}]$$

где LA — адрес списка полных слов, каждое из которых содержит размер освобождаемой области основной памяти в байтах. Старший разряд последнего слова в списке содержит 1. Количество слов в списке соответствует количеству освобождаемых областей;

A — адрес списка полных слов, в которых находятся адреса освобождаемых областей основной памяти. Количество слов в данном списке соответствует количеству областей в списке LA;

SP — номер подпула, в котором освобождается область основной памяти.

Не допускается использование спискового формата в режиме MFT.

Переменный формат макрокоманды FREEMAIN:

[метка] FREEMAIN V,A=адрес[SP=число]

где A — адрес списка из двух слов, первое из которых содержит адрес освобождаемой области основной памяти, а второе — ее размер;

SP — номер подпула, в котором освобождается область основной памяти.

Для всех форматов макрокоманды FREEMAIN номер подпула в режиме MFT не используется. Если операнд SP опущен, то по умолчанию подразумевается нулевой подпул.

Описательная и исполнительная формы макрокоманды FREEMAIN элементного, спискового и переменного форматов образуются так же, как для макрокоманды EXTRACT.

5.5. Служба времени

5.5.1. Макрокоманда TIME

Макрокоманда TIME позволяет получить текущую дату и текущее время дня. Управляющая программа помещает в регистр I текущую дату в формате десятичных упакованных чисел: 00YYDDDC. Здесь YY — означает две последние цифры года, DDD — порядковый номер дня в году, C — знак плюс. В регистр 0 управляющая программа помещает текущее время дня в формате, зависящем от закодированного операнда. Формат макрокоманды:

[метка] TIME $\left[\begin{array}{c} \text{DEC} \\ \text{BIN} \\ \text{TU} \end{array} \right]$

где DEC — означает, что текущее время дня в регистре 0 должно быть представлено в формате десятичного упакованного числа: HHMMSSth. Здесь HH — часы, MM — минуты, SS — секунды, t — десятые доли секунды, h — сотые доли секунды;

BIN — означает, что текущее время дня в регистре 0 должно быть представлено как двоичное число без знака, цена единицы которого — одна сотая секунды;

TU — означает, что текущее время дня в регистре 0 должно быть представлено как двоичное число без знака в единицах таймера (цена единицы таймера 26,04 мкс).

5.5.2. Макрокоманда STIMER

Макрокоманда STIMER позволяет установить временной интервал или интервал, который будет исчерпан к заданному времени дня. С задачей может быть связан только один временной интервал. Следовательно, каждая следующая макрокоманда STIMER, выданная той же задачей, отменяет действие предыдущей. Формат макрокоманды:

$$\begin{aligned} [\text{метка}] \quad \text{STIMER} \quad & \left\{ \begin{array}{l} \text{REAL} [\text{,адрес-программы-выхода}] \\ \text{TASK} [\text{,адрес-программы-выхода}] \\ \text{WAIT} \end{array} \right\} \\ & \left\{ \begin{array}{l} \text{,DINTVL} = \text{адрес} \\ \text{,BINTVL} = \text{адрес} \\ \text{,TUINTVL} = \text{адрес} \\ \text{,TOD} = \text{адрес} \end{array} \right\} \end{aligned}$$

где REAL — означает, что временной интервал должен отсчитываться непрерывно. После истечения заданного интервала управление должно быть передано на указанную программу выхода;

TASK — указывает, что временной интервал должен отсчитываться, если задача является активной (чистое процессорное время). После истечения заданного интервала управление должно быть передано на указанную программу выхода;

WAIT — означает, что временной интервал должен отсчитываться непрерывно и задача должна быть переведена в состояние ожидания на заданный интервал времени. После его истечения управление должно быть передано команде, следующей за макрокомандой STIMER;

адрес-программы-выхода — адрес программы асинхронного выхода, по которому передается управление после истечения заданного временного интервала. Если этот операнд опущен, то об истечении интервала не сообщается;

DINTVL — адрес двойного слова в основной памяти (в границах двойного слова), которое содержит временной интервал в формате десятичного неупакованного числа: HHMMSSth. Здесь HH — часы, MM — минуты, SS — секунды, t — десятичные доли секунды, h — сотые доли секунды;

BINTVL — адрес полного слова в основной памяти (в границах слова), содержащего временной интервал в виде двоичного числа без знака, цена единицы которого — одна сотая секунды;

TUINTVL — адрес полного слова в основной памяти (в границах слова), содержащего временной интервал в виде двоичного числа без знака, цена единицы которого равна одной единице таймера (26,04 мкс);

TOD — адрес двойного слова в основной памяти (в границах двойного слова), содержащего время дня, когда должен завер-

шиться временной интервал. Формат представления времени дня такой же, как формат представления интервала в операнде DINTVL. Если операнд TOD задан с операндом TASK, то он рассматривается как операнд DINTVL.

5.5.3. Макрокоманда TTIMER

Макрокоманда TTIMER позволяет запросить управляющую программу о количестве времени, оставшегося до истечения временного интервала, заданного макрокомандой STIMER, или отменить заданный интервал. Количество оставшегося времени управляющая программа передает в регистре 0 в виде двоичного числа без знака в единицах таймера (26,04 мкс). Если временной интервал не был установлен, то регистр 0 содержит нулевое значение. Формат макрокоманды:

[метка] TTIMER [CANCEL]

где CANCEL — означает, что оставшийся временной интервал и программа выхода, если они были установлены макрокомандой STIMER, должны быть отменены. Если в макрокоманде STIMER был указан операнд WAIT, то задача не выводится из состояния ожидания. Если операнд CANCEL не был закодирован, то отсчет заданного временного интервала будет продолжаться.

5.6. Управление последовательно используемыми ресурсами

5.6.1. Макрокоманда ENQ

Макрокоманда ENQ позволяет для активной задачи получить монопольное или совместное управление одним или несколькими последовательно используемыми ресурсами. Ресурс представляется q- и г-именами и с каким-либо физическим ресурсом не связан. Если же ресурс занят другими задачами и не доступен, то активная задача попадает в состояние ожидания. С помощью макрокоманды ENQ можно также проверить состояние ресурса. Освобождение ресурса производится макрокомандой DEQ. Формат макрокоманды ENQ:

[метка] ENQ (адрес-q-имени, адрес-г-имени, [E S],
[длина-г-имени], [SYSTEM], ...) [,RET=TEST
[STEP] [,RET=USE
[,RET=HAVE]

где адрес-q-имени — адрес имени из 8 символов в основной памяти;

адрес-г-имени — адрес имени от 1 до 255 символов в основной памяти. Имя может быть составным;

E — запрос на монопольное управление ресурсом;

S — запрос на совместное использование ресурсов; по умолчанию предполагается E;

длина-г-имени — количество знаков в г-имени. Если указан 0, длина г-имени должна содержаться в первом байте адреса г-имени. Если данный операнд опущен, то используется длина г-имени, полученная при работе Ассемблера;

SYSTEM — ресурс может использоваться программами нескольких заданий;

STEP — ресурс используется только внутри пункта задания программы, выдавшей макрокоманду ENQ; по умолчанию предполагается STEP;

RET=TEST — условный запрос, проверяющий состояние доступности ресурсов, не требуя управления ими. Результат проверки указывается кодом возврата;

RET=USE — условный запрос, определяющий, что управление ресурсами должно быть приписано задаче только в том случае, если ресурс доступен немедленно. Если ресурс не доступен, выдается соответствующий код возврата, и задача в состоянии ожидания не переводится;

RET=HAVE — условный запрос, определяющий, что управление ресурсом требуется только в том случае, если в этой задаче такие запросы не были сделаны раньше.

В случае условных запросов выдаются следующие коды возврата: 0 — ресурс доступен немедленно, 4 — ресурс не доступен, 8 — в этой задаче уже был сделан ранее запрос на тот же ресурс. Если все коды возврата равны 0, то регистр 15 содержит 0, иначе он содержит адрес таблицы, в которой находятся коды возврата для каждого ресурса.

Описательная и исполнительная формы макрокоманды ENQ образуются так же, как для макрокоманды EXTRACT. В исполнительной форме макрокоманды ENQ допускается кодировать RET=NONE, что указывает безусловный запрос на управление ресурсами.

5.6.2. Макрокоманда DEQ

Макрокоманда DEQ производит освобождение активной задачи от управления одним или несколькими последовательно используемыми ресурсами, а также позволяет определить, присвоено ли управление ресурсами активной задаче и запрошен ли ресурс активной задачей. Формат макрокоманды:

```
[метка] DEQ (адрес-г-имени,адрес-г-имени,  
              [длина-г-имени], [STEP  
                                SYSTEM  ] ,...)  
              [,RET=HAVE]
```

где операнды в круглых скобках имеют тот же смысл, что и в макрокоманде ENQ;

RET=HAVE — условный запрос, определяющий, что требование освободить управление всеми ресурсами, указанными в макрокоманде, должно быть выполнено, если управление этими ресурсами было предоставлено активной задаче. Результаты условных запросов выдаются следующими кодами возврата: 0 — задача освобождена от управления ресурсом, 4 — задача запрашивала управление ресурсом, но не получила его; 8 — перед выдачей макрокоманды DEQ задача не имела управления ресурсом. Если все коды возврата равны 0, то регистр 15 содержит 0, иначе он содержит адрес таблицы, в которой находятся коды возврата для каждого ресурса.

Описательная и исполнительная формы макрокоманды DEQ образуются так же, как для макрокоманды ENQ.

5.7. Обработка программных прерываний

5.7.1. Макрокоманда SPIE

Макрокоманда SPIE используется для указания адреса программы обработки прерываний и типа программных прерываний, которые должны вызвать передачу управления на эту программу. Каждая следующая макрокоманда SPIE отменяет действие предыдущей макрокоманды SPIE, выданной в той же задаче. Формат макрокоманды:

[метка] SPIE [адрес-программы-обработки-прерываний, (прерывания)]

где адрес-программы-обработки-прерываний — адрес, по которому должно быть передано управление после возникновения программного прерывания, указанного в операнде «прерывания»;

прерывания — одно или несколько десятичных чисел, разделенных запятыми и определяющих соответствующие типы прерываний. При этом могут быть заданы как одиночные числа (перечисление), так и пары чисел в скобках, разделенные запятой, которые обозначают интервал соответствующих типов прерываний.

Описательная и исполнительная формы макрокоманды SPIE образуются так же, как для макрокоманды EXTRACT.

5.8. Аварийное завершение задач

5.8.1. Макрокоманда ABEND

Макрокоманда ABEND вызывает аварийное завершение текущего пункта задания или текущей задачи вместе с ее подзадачами. Если завершается текущий пункт задания, то раздел основной памяти, занятый этим пунктом, освобождается. Если завершается текущая задача, то управление передается программе выхода, определенной операндом STAI макрокоманды ATTACH, которая образовала данную задачу.

[метка] ABEND код-завершения[, [DUMP] [, STEP]]

DUMP — определяет выдачу содержимого области памяти раздела, ядра, общих регистров, таблицы прослеживания и управляющих блоков (дампа) завершаемого пункта задания в набор данных, определенный оператором DD с именем SYSABEND, или содержимого области памяти раздела, общих регистров, таблицы прослеживания и управляющих блоков завершаемого пункта задания в набор данных, определенный оператором DD с именем SYSUDUMP. Если упомянутые наборы данных не были определены в пункте задания, то на системное устройство вывода выдается только содержимое общих регистров и некоторых управляющих блоков завершаемого пункта задания;

5.8.2. Макрокоманда STAE

$$\begin{aligned} \text{[метка] STAE } & \left\{ \begin{array}{l} 0 \\ \text{адрес-программы-выхода} \end{array} \right\} \left[\begin{array}{l} .OV \\ .CT \end{array} \right] \\ & \text{[PARAM=адрес-списка] } \left[\begin{array}{l} XCTL = \{ YES \\ NO \} \end{array} \right] \\ & \left[\begin{array}{l} PURGE = \{ QUIESCE \\ HALT \\ NONE \} \end{array} \right] \left[\begin{array}{l} ASYNCH = \{ YES \\ NO \} \end{array} \right] \end{aligned}$$

где адрес-программы-выхода — адрес программы асинхронного выхода STAE, которой должно быть передано управление, если задача, выдавшая эту макрокоманду, аварийно завершается. Адрес может быть загружен в один из общих регистров 2—12. Если указано 0, то отменяется самый поздний запрос STAE;

OV — параметры данной макрокоманды STAE должны заменить параметры предыдущего запроса;

CT — образование нового запроса STAE. По умолчанию предполагается CT;

PARAM — адрес списка параметров для программы асинхронного выхода. Адрес может быть загружен в один из общих регистров от 2—12;

XCTL — определяет, отменяется ли режим STAE при выдаче макрокоманды XCTL (YES — не отменяется, NO — отменяется). По умолчанию предполагается NO.

Назначение параметров PURGE и ASYNCH аналогично назначению соответствующих параметров макрокоманды ATTACH.

О результатах выполнения макрокоманды STAE можно судить по содержимому регистра 15.

Описательная и исполнительная формы макрокоманды STAE образуются так же, как для макрокоманды EXTRACT. В описательной форме макрокоманды STAE не допускается указание 0 в первом операнде и использование операндов OV и CT.

5.9. Выдача дампа

5.9.1. Макрокоманда SNAP

Макрокоманда SNAP используется для выдачи дампа. По этой макрокоманде управляющая программа выводит некоторые области основной памяти, слово состояния программы и содержимое общих регистров. Для дампа должен быть отведен набор данных, для которого необходимо предусмотреть оператор DD и открыть блок управления данными DCB. Формат макрокоманды:

```
[метка] SNAP DCB=адрес[,TCB=адрес][,ID=число]  
[,SDATA=(код-для-блоков-управляющей-программы)]  
[,PDATA=(код-для-областей-программы-пользователя)]  
[,STORAGE=(начальный-адрес,конечный-адрес,...)]  
[,LIST=адрес-списка]
```

где DCB — адрес блока управления данными DCB;

TCB — адрес полного слова в границах слова, содержащего адрес блока управления данными TCB. Пропуск или нулевое значение этого операнда указывает на TCB активной задачи;

ID — число от 0 до 255 (в MFT от 1 до 127), идентифицирующее конкретное использование макрокоманды. Это число распечатывается в заголовке дампа;

SDATA — следующие наборы символов, определяющие запросы на вывод полей информации управляющей программы: **ALL** — все перечисленные далее поля; **NUC** — ядро без таблицы прослеживания; **TRT** — таблица прослеживания; **CB** — все управляющие блоки задачи; **Q** — блоки управления очередями и элементы очередей;

PDATA — следующие наборы символов, определяющие запросы на вывод полей информации программы пользователя: **ALL** — все перечисленные далее поля; **PSW** — слово состояния программы в момент выдачи макрокоманды **SNAP**; **REGS** — содержимое общих регистров в момент выдачи макрокоманды **SNAP**; **SA** или **SAH**, где **SA** — информация о связях и областях сохранения; **SAH** — только информация о связях; **LPA** или **ALLPA** или **JPA**, где **JPA** — содержимое всей основной памяти пункта задания (**MFT**), содержимое области задания (**MVT**, **SVS**), **LPA** — содержимое области резидентных реентерабельных программ (**MFT**), содержимое общей области (**MVT**, **SVS**), **ALLPA** — содержимое обеих областей как **JPA**, так и **LPA**; **SPLS** — содержимое всей основной памяти пункта задания (**MFT**), все подпулы основной памяти от 0 до 127 (**MVT**, **SVS**);

STORAGE — одна или несколько пар начальных и конечных адресов областей основной памяти, которые должны быть выданы;

LIST — адрес списка начальных и конечных адресов областей, которые должны быть выданы. Старший разряд слова последнего конечного адреса в списке должен содержать 1.

О результатах выполнения макрокоманды **SNAP** при возврате управления можно судить по содержимому регистра 15.

Описательная и исполнительная формы макрокоманды **SNAP** образуются так же, как для макрокоманды **EXTRACT**. В описательной форме макрокоманды **SNAP** не допускается кодирование операнда **TCB**.

5.10. Вывод сообщений

5.10.1. Макрокоманда **WTL**

Макрокоманда **WTL** вызывает запись сообщения в системный журнал. Сообщение может содержать любые символы, допустимые в операторе **DC** типа **C** языка Ассемблера. Максимальная длина сообщения — 126 символов. Формат макрокоманды:

[метка] **WTL** 'сообщение'

где сообщение — текст сообщения, которое должно быть записано в системный журнал.

Описательная и исполнительная формы макрокоманды **MTL** образуются так же, как для команды **EXTRACT**. Сообщение может быть указано только в описательной форме.

5.10.2. Макрокоманда WTO

Макрокоманда WTO вызывает вывод сообщения на консоль оператора. Формат макрокоманды:

```
[метка WTO { 'сообщение'  
  { ('текст'[тип]) [ ('текст'[тип]) ], ... }  
  [, ROUTCDE = (число [число], ...)]  
  [, DESC = число] [, AREAID = имя]  
  [, MCSFLAG = (имя [имя], ...)]  
  [, MSGTYP = { N  
                Y  
                JOB NAMES  
                STATUS } ] }
```

где сообщения — текст сообщения, которое должно быть выведено на консоль оператора. Сообщение может содержать любые символы, допустимые в операторе DC типа C языка Ассемблера, за исключением символа новой строки. Максимальная длина сообщения — 122 символа для MFT и 126 символов для MVT и SVS, но при DESC=1 или DESC=2 максимальная длина уменьшается на 1;

('текст'[тип]) [('текст'[тип])]... — используется для выдачи многострочных сообщений в операционной системе с мультиконсольным обеспечением. Сообщение может содержать до 10 строк;

текст — одна строка многострочного сообщения, которое может содержать любые символы, допустимые в операторе DC типа C языка Ассемблера, за исключением символа новой строки. Максимальная длина строки зависит от ее типа: для типа C — 34 символа, для типов L, D и DE — 70 символов;

тип — индикатор типа информации, содержащейся в тексте. Возможны следующие типы информации:

C — строка содержит управляющую информацию,

L — строка содержит заголовок сообщения. Допускается не более двух строк заголовка. Им может предшествовать только строка типа C,

D — строка содержит часть сообщения,

DE — строка является последней в сообщении,

E — предыдущая строка была последней;

ROUTCDE — маршрутные коды сообщения;

DESC — код дескриптора, присваиваемый сообщению;

AREAID — область состояния на экране дисплей-консоли;

MCSFLAG и MSGTYP — операнды системного программиста.

Описательная и исполнительная формы макрокоманды WTO образуются так же, как для макрокоманды EXTRACT. В исполнительной форме допускается употребление только операнда MF. В описательной форме не допускается употребление операнда AREAID. Первый операнд обязателен.

5.10.3. Макрокоманда WTOR

Макрокоманда WTOR вызывает вывод сообщения на консоль оператора с ответом. Формат макрокоманды:

[метка] WTOR 'сообщение' адрес-ответа, длина-ответа,
адрес-есб[,ROUTCDE= (число[,число]...)]
[,DESC=число]
[,MCSFLAG=(имя[,имя]...)]
[,MSGTYP={
N
Y
JOBNAME
STATUS
}]

где сообщение — текст сообщения, которое должно быть выведено на консоль оператора. Сообщение может содержать любые символы, допустимые в операторе DC типа С языка Ассемблера, за исключением символа новой строки. Максимальная длина сообщения — 118 символов для MFT и 122 символа для MVT и SVS;

адрес-ответа — адрес области основной памяти, в которую управляющая программа должна поместить ответ оператора;

длина-ответа — длина ответного сообщения оператора в байтах. Максимальная длина — до 119 байт;

адрес-есб — адрес блока управления событием для использования управляющей программой для указания о завершении передачи ответа;

операнды ROUTCDE, DESC, MCSFLAG и MSGTYP имеют аналогичное назначение соответствующих операндов макрокоманды WTO.

Описательные и исполнительные формы макрокоманды WTOR образуются так же, как для макрокоманды EXTRACT. В исполнительной форме допускаются операнды: адрес-ответа, длина-ответа, адрес-есб и MF.

5.10.4. Макрокоманда DOM

Макрокоманда DOM позволяет удалить сообщения, выводимые на дисплей-консоль оператора в операционных системах с мультikonсольным обеспечением. При выдаче сообщений по макрокомандам WTO и WTOR операционная система присваивает сообщениям 24-разрядные идентификационные номера, которые возвращаются исходной программе в регистре 1.

Формат макрокоманды DOM:

[метка]DOM {MSG=регистр
MSGLIST=адрес-списка }

где MSG — задает регистры 1—12, в которых содержится идентификационный номер сообщения, подлежащего уничтожению;

MSGLIST — задает адрес списка одного или нескольких слов, каждое из которых содержит идентификационный номер сообщения, подлежащего уничтожению. Последнее слово в списке отмечается 1 в старшем разряде. В списке допускается до 60 слов.

В системе без мультikonсольного обеспечения макрокоманда DOM рассматривается как пустая операция.

В регистре 15 выдается код возврата: 0 — успешное выполнение макрокоманды, 4 — запрос на освобождение не выполнен.

Описательная и исполнительная формы макрокоманды PGRLSE образуются так же, как для макрокоманды EXTRACT.

5.12.2. Макрокоманда STATUS

Макрокоманда STATUS вызывает изменение признаков задачи путем изменения полей признаков в блоке управления задачами TCB. Формат макрокоманды:

[метка] STATUS { START
 STOP } [,TCB=адрес]

где START — увеличение на единицу счетчика START/STOP в TCB в режиме SVS;

STOP — уменьшение на единицу счетчика START/STOP в TCB в режиме SVS;

TCB — адрес слова основной памяти в границах слова, содержащего адрес TCB, в котором выполняются изменения признаков. Если этот операнд опущен, то изменения признаков выполняются для всех подзадач активной задачи и самой активной задачи.

В регистре 15 выдается код возврата: 0 — успешное выполнение макрокоманды, 4 — макрокоманда не выполнена.

5.13. Контрольная точка

5.13.1. Макрокоманда СНКРТ

Макрокоманда СНКРТ фиксирует информацию, необходимую для рестарта пункта задания с контрольной точки. Повторение начинается с команды, следующей за макрокомандой. Для записи информации контрольной точки должен быть предусмотрен набор данных. Формат макрокоманды:

[метка] СНКРТ { адрес-dcb[,адрес-идентификатора
 CANCEL
 [,длина-идентификатора]]
 [, 'S'] }

где адрес-dcb — адрес блока управления данными для набора данных контрольной точки;

адрес-идентификатора — адрес поля, содержащего идентификатор контрольной точки;

длина-идентификатора — количество байт в поле, содержащем идентификатор контрольной точки; по умолчанию — 8 байт;

'S' — идентификатор контрольной точки вырабатывается операционной системой; идентификатор выдается в сообщении оператору;

CANCEL — аннулирует запрос на автоматический рестарт для самой последней контрольной точки.

Описательная и исполнительная формы макрокоманды СНКРТ образуются так же, как для макрокоманды EXTRACT. Операнд CANCEL не допускается в описательной и исполнительной формах.

Управление данными ОС ЕС

6.1. Общие сведения

6.1.1. Форматы данных

Управление данными ОС ЕС обеспечивает обработку данных на уровне логических записей и блоков, иначе называемых *физическими записями*. Логическая запись — это единица информации, обрабатываемая программой как одно целое. Обмен данными между периферийными устройствами и основной памятью осуществляется блоками. Совокупность логически связанных блоков на периферийном устройстве образует набор данных.

Средствами управления данными обрабатываются три формата логических записей: записи фиксированной длины, записи переменной длины и записи неопределенной длины.

Записи фиксированной длины характеризует их постоянный размер для всего набора данных. Идентификаторы формата логических записей фиксированной длины: F, FB, FS и FBS.

В каждом блоке набора данных, состоящего из записей формата F, содержится ровно одна логическая запись.

В каждом блоке набора данных, состоящего из записей формата FB, может содержаться более одной логической записи.

В наборе данных на устройстве прямого доступа, состоящем из записей формата FS, нет незаполненных дорожек, кроме последней, и каждый блок содержит ровно одну запись.

В наборе данных, состоящем из записей формата FBS, нет укороченных блоков и незаполненных дорожек, за исключением последнего блока или дорожки. Все блоки такого набора данных, кроме последнего, состоят из одинакового числа логических записей.

Записи переменной длины состоят из двух полей дескриптора записи и данных, следующих за дескриптором. Идентификаторы формата логических записей переменной длины: V, VB, VS и VBS.

В каждом блоке набора данных, состоящего из записей формата V, содержится ровно одна запись. Дескриптор записи представляет собой 4-байтовое поле, в первых двух байтах которого содержится длина логической записи, включая длину дескриптора, а во вторых двух байтах — нули. Дескриптор записи является одновременно и дескриптором блока. Длина записи должна быть не менее 4 и не более 32 756 байт.

Каждый блок набора данных, состоящего из записей формата VB, включает в себя дескриптор блока и одну или более записей формата V. Дескриптор блока представляет собой 4-байтовое поле, в первых двух байтах которого содержится общая длина всех логических записей в блоке и дескриптора блока. Длина блока должна быть не менее 4 (для магнитной ленты не менее 18) и не более 32 760 байт.

Логические записи формата VS называются *расширенными записями переменной длины*. Каждая запись этого формата может состоять из нескольких блоков и даже занимать несколько томов. Часть записи формата VS, расположенная в одном блоке, называется *сегментом*. Каждый сегмент записи формата VS состоит из дескриптора сегмента и данных, следующих за дескриптором. Первые два байта дескриптора сегмента по содержанию совпадают с дескриптором записи. Два младших разряда третьего байта содержат управляющий код сегмента, указывающий положение сегмента в логической записи. Остальные разряды третьего байта и четвертый байт должны быть нулевыми.

Возможны следующие управляющие коды:

00 — полная логическая запись;

01 — первый сегмент записи, состоящей из нескольких сегментов;

10 — последний сегмент записи, состоящей из нескольких сегментов;

11 — сегмент записи, отличный от первого и последнего сегментов.

В одном блоке записей формата VBS может находиться более одной логической записи формата VS или последний и первый сегменты двух различных логических записей. При блокировании записей формата VS дескриптор каждого блока содержит длину всех логических записей или сегментов в блоке.

Записи неопределенной длины (формат U) не содержат в себе информации о длине и обрабатываются как отдельные блоки. Средствами операционной системы ОС ЕС не предусмотрено выделение отдельных логических записей из блока.

6.1.2. Управляющие символы логических записей

Операционная система ОС ЕС будет интерпретировать первый символ данных каждой логической записи перечисленных выше форматов как команду управления печатающим устройством **или** перфокарточным устройством вывода, если наряду с форматом записей в операторе DD, макрокоманде DCB или метке набора данных указано A или M (см. 6.3,7.2).

Управляющие символы в коде ДКОИ (A) и двоичном машинном коде (M) приведены в табл. 6.1.

Таблица 6.1

А	М	Значение	А	М	Значение
Пробел	0B	Пропустить одну строку	7	BB	Перейти к дорожке 7
0	13	Пропустить две строки	8	C3	То же 8
—	1B	Пропустить три строки	9	CB	» 9
+	03	Подавить пропуск строки	A	D3	» 10
1	8B	Перейти к дорожке 1	B	DB	» 11
2	93	То же 2	C	E3	» 12
3	9B	» 3	V	23	Выбрать приемный карман 1
4	A3	» 4	W	63	Выбрать приемный карман 2
5	AB	» 5			
6	B3	» 6			

Примечание. Любой другой символ, отличный от перечисленных в табл. 6.1, будет интерпретироваться как «пробел» или V.

6.1.3. Формат и емкость дорожек устройств прямого доступа

Каждая дорожка устройств прямого доступа всех типов, помимо данных, содержит вспомогательную информацию: маркер начала оборота, собственный адрес, промежутки между блоками, запись, описывающую дорожку и блоки данных.

Маркер начала оборота на устройстве прямого доступа указывает физическое начало каждой дорожки.

Собственный адрес — это семибайтовая область на каждой дорожке, состоящая из следующих полей.

Размер в байтах

Содержимое

0	Признак, определяющий состояние дорожки — рабочая или дефектная, основная или запасная
1—2	Номер цилиндра, в который входит дорожка
3—4	Номер головки, обслуживающей эту дорожку
5—6	Информация для циклического контроля

Промежутки между блоками изменяются в зависимости от типа устройства, местоположения промежутка и длины предшествующего блока.

Запись, описывающая дорожку (R0) располагается вслед за собственным адресом и содержит информацию о занятости дорожки для записей набора данных с прямой организацией.

Блоки данных могут быть с ключами и без ключей. В первом случае каждый блок состоит из четырех областей: адресного маркера, ключа и данных, а во втором случае — из трех: адресного маркера, счетчика и данных.

Адресный маркер — двухбайтовая область, которая позволяет устройству управления определить место начала блока при его чтении.

Область счета имеет размер 11 байт и состоит из следующих байтовых полей:

Размер в байтах

Содержимое

0	Признак, определяющий состояние дорожки—рабочая или дефектная, основная или запасная
1—2	Номер цилиндра, в который входит дорожка
3—4	Номер головки
5	Номер блока (от 1 до FF)
6	Длина ключа
7—8	Длина данных в двоичном представлении от 0 до 65 535
9—10	Информация для циклического контроля

Область ключа заполняется программно. Размер этой области может меняться в пределах от 1 до 255. Для записей без ключей в области счета в поле длины ключа содержится 0.

В ОС ЕС память для наборов данных на устройствах прямого доступа задается в операнде SPACE оператора DD в блоках, дорожках или цилиндрах. Емкость дорожек для устройств прямого доступа ОС ЕС ЭВМ приводится в табл. 6.2. Под емкостью дорожки подразумевается количество байт для блоков данных, исключая области вспомогательной информации, необходимые для формирования одного блока (собственный адрес, маркер адреса, запись R0 и т. д.).

Таблица 6.2

Емкость дорожек устройств прямого доступа

Тип устройства	Емкость дорожки	Число дорожек в одном цилиндре	Число цилиндров	Полная емкость тома, байт
ЕС-5050	3 625	10	200	7 250 000
ЕС-5052	3 625	10	200	7 250 000
ЕС-5055	3 625	10	200	7 250 000
ЕС-5056	3 625	10	200	7 250 000
ЕС-5061	7 294	20	200	29 176 000
ЕС-5066	13 030	19	404	100 018 280

Для расчета количества блоков заданной длины на дорожке следует пользоваться формулами, приведенными в табл. 6.3. При этом для устройств ЕС-5052, ЕС-5055 и ЕС-5056 следует использовать формулы, приведенные для устройства ЕС-5050.

Таблица 6.3

Формулы расчета количества блоков данных на дорожке устройства прямого доступа

Устройство прямого доступа	Число байт для б. о. ка данных			
	блоки с ключами		блоки без ключей	
	любой блок на дорожке	последний блок на дорожке	любой блок на дорожке	последний блок на дорожке
ЕС-5050	$81 + 1,049 \times (KL + DL)$	$20 + KL + DL$	$61 + 1,049(DL)$	DL
ЕС-5061	$146 + 1,043 \times (KL + DL)$	$45 + KL + DL$	$101 + 1,043(DL)$	DL
ЕС-5066	$191 + KL + DL$	$191 + KL + DL$	$135 + DL$	$135 + DL$

Примечание. KL — длина ключа; DL — длина данных.

6.2. Организация данных в ОС ЕС

Программные средства операционной системы ОС ЕС обеспечивают работу с четырьмя типами организации наборов данных: последовательной, прямой, библиотечной и индексно-последовательной.

6.2.1. Последовательный набор данных

Последовательная организация предполагает расположение логических записей набора данных на носителе информации в физической последовательности. Носители информации для последовательных наборов данных: перфокарты, перфоленга, магнитная лента, накопитель на магнитных дисках. Структура последовательного набора данных ОС ЕС ЭВМ на магнитной ленте со стандартными метками представлена на рис. 6.1.

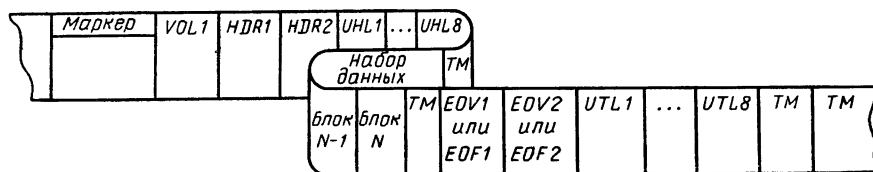


Рис. 6.1. Последовательный набор данных на томе магнитной ленты

На рис. 6.1 использованы следующие обозначения:

маркер — физическая отметка начала тома;

VOL1 — VOL8 — метки тома, 80-байтовые записи, первая из которых создается и проверяется операционной системой;

HDR1 — HDR2 — метки формата 1 и 2 начала набора данных; 80-байтовые записи создаются и проверяются операционной системой;

UHL1 — UHL8 — метки пользователя начала набора данных; 80-байтовые записи создаются и поддерживаются программой пользователя;

ТМ — ленточная марка;

Блок 1 — блок N — блоки набора данных;

EOV1 — EOV2 — метки формата 1 и 2 конца тома; 80-байтовые записи создаются и проверяются операционной системой для многотомного набора данных при обнаружении конца тома;

EOF1 — EOF2 — метки формата 1 и 2 конца набора данных; 80-байтовые записи, создаются и проверяются операционной системой;

UTL1 — UTL8 — метки пользователя конца набора данных; 80-байтовые записи создаются и поддерживаются программой пользователя.

После последней метки конца тома или метки конца набора данных следует ленточная марка, а если набор данных последний и не многотомный, то — две ленточные марки.

6.2.2. Прямой набор данных

Прямая организация набора данных возможна только на устройствах, допускающих непосредственную выборку данных. Таими устройствами в ОС ЕС ЭВМ являются накопители на магнитных дисках. Взаимосвязи внутри прямого набора данных устанавливаются программистом. Поиск необходимого блока в прямом наборе данных осуществляется по ключу либо по идентификатору блока.

6.2.3. Библиотечный набор данных

Библиотечный набор данных состоит из разделов, каждый из которых представляет собой последовательный набор данных, имя которого, адрес относительно начала набора данных в форме TTR (ТТ — номер дорожки, а R — номер записи) и другие характеристики хранятся в справочнике, представляющем собой также последовательный набор данных. Структура библиотечного набора данных представлена на рис. 6.2.

6.2.4. Индексно-последовательный набор данных

Индексно-последовательная организация набора данных позволяет обрабатывать логические записи форматов F, FB, V и VB последовательно и в любом порядке удалять старые и добавлять новые записи. Такая организация набора данных допустима только для устройств прямого доступа. Индексно-последовательный набор данных может занимать до трех областей: основную область,

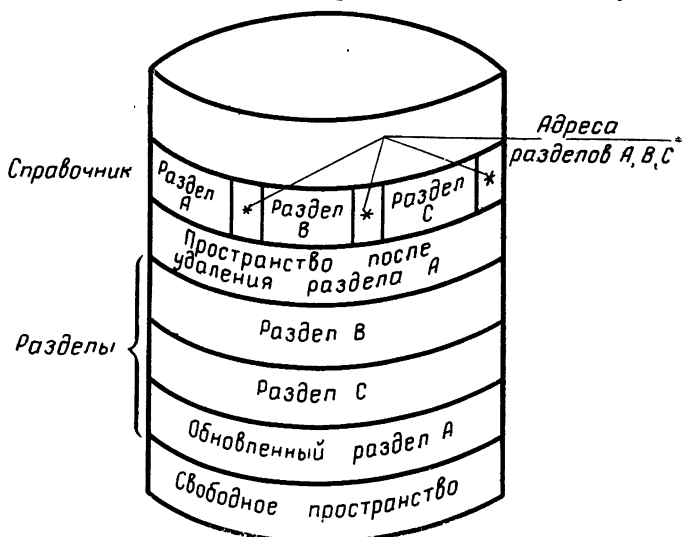


Рис. 6.2. Структура библиотечного набора данных

область индексов и независимую область переполнения. При создании индексно-последовательного набора данных все три области описываются с помощью сцепленных операторов DD, в операндах DSNAMES которых вместе с именем набора данных в скобках указываются имена трех областей: PRIME, INDEX и OVFLOW.

Основная область (PRIME) используется для размещения логических записей данных и индексов дорожек. Индекс дорожек создается для каждого цилиндра основной области и располагается на первых дорожках цилиндра. Индекс дорожек состоит из последовательности парных элементов, первый из которых адресует дорожку основной области, а второй — дорожку независимой области переполнения или области переполнения цилиндра. Второй элемент связывает таким образом последнюю логическую запись на дорожке основной области с первой логической записью, вытесненной в область переполнения после включения на дорожку основной области новой логической записи.

Область индексов используется для размещения индексов цилиндров и главных индексов. Индекс цилиндров создается для каждого цилиндра основной области. Каждый элемент индекса ци-

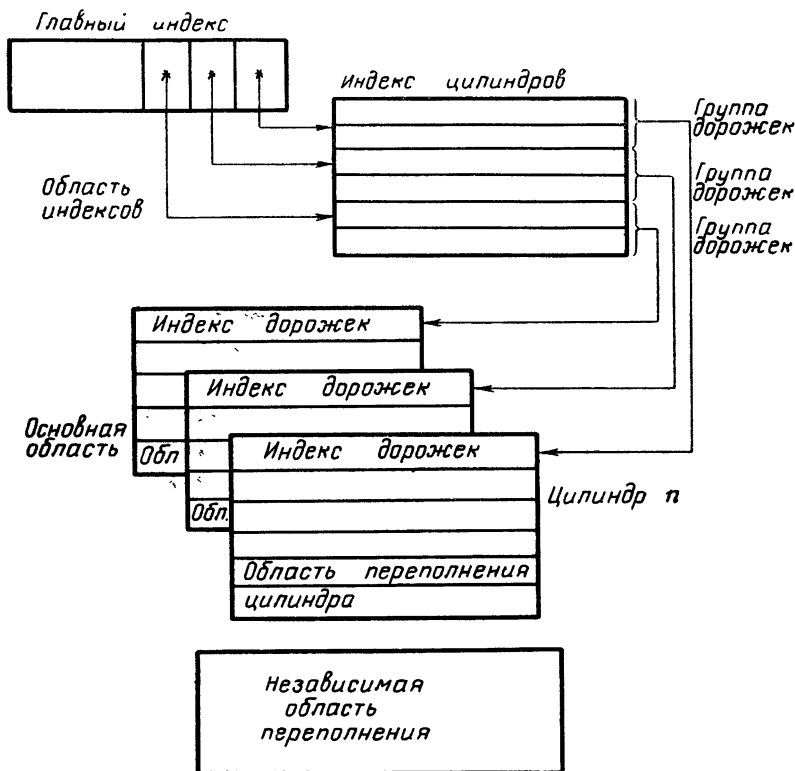


Рис. 6.3. Структура индексно-последовательного набора данных

линдров адресует один индекс дорожек соответствующего цилиндра основной области. Для сокращения времени просмотра индекса цилиндров можно запросить построение главных индексов.

Главный индекс строится в том случае, если в операнде OPTCD макрокоманды DCB указан параметр M и индекс цилиндров занимает число дорожек, большее, чем указано в операнде NTM этой макрокоманды. Каждый элемент главного индекса адресует группу дорожек индекса цилиндров, число которых определено операндом NTM. Если размер области, занимаемой главным индексом, превышает число дорожек, указанное в операнде NTM, то операционная система построит главный индекс более высокого уровня, указывающий на элементы главного индекса. Структура индексно-последовательного набора данных представлена на рис. 6.3.

6.2.5. Оглавление тома прямого доступа

Информация о наборах данных на томе прямого доступа хранится в оглавлении тома (VTOC), адрес которого находится в первой метке тома (VOL1). Первая метка тома всегда является третьей записью на нулевой дорожке нулевого цилиндра тома прямого доступа.

Оглавление тома состоит из 96-байтовых блоков данных (DSCB) с 44-байтовыми областями ключей. В зависимости от назначения и формата DSCB делятся на 7 типов.

DSCB формата 0 заполняют неиспользуемую часть области оглавления тома и состоят из двоичных нулей. DSCB формата 1 содержат описания характеристик наборов данных, находящихся на томе прямого доступа. DSCB формата 2 содержат описание характеристик областей индексов индексно-последовательных наборов данных. DSCB формата 3 создаются для наборов данных, заполняющих более трех несмежных областей (экстентов) на томе прямого доступа. DSCB формата 4 содержит описание характеристик экстента, выделенного для оглавления тома. DSCB формата 5 описывают свободные экстенты на томе прямого доступа. DSCB формата 6 описывают экстенты, совместно используемые несколькими наборами данных.

Первым блоком оглавления тома всегда является единственный DSCB формата 4. В DSCB формата 4 содержится поле адреса DSCB формата 6. За DSCB формата 4 следуют один или более DSCB формата 5. Все DSCB формата 5 связываются между собой в цепочку с помощью поля адреса следующего DSCB формата 5. DSCB формата 1 можно отыскать в оглавлении тома, указав в качестве аргумента поиска имя набора данных, которое помещается в область ключа при создании набора данных. Метки пользователя для набора данных на томе прямого доступа занимают первый экстент размером в одну дорожку, описанный в DSCB формата 1.

В этом же DSCB содержится поле адреса DSCB формата 2 или DSCB формата 3.

6.3. Метки

6.3.1. Метки томов

Формат первой метки тома

Размер поля в байтах	Содержимое
0—2	Идентификатор метки тома (VOL)
3	Номер метки тома (1)
4—9	Регистрационный номер тома
10	Признак защиты тома
11—15	Адрес в форме CCHNR оглавления тома прямого доступа; для магнитной ленты — пробелы
16—20	Пробелы
21—30	Идентификатор изготовителя или пробелы
31—40	Пробелы
41—50	Идентификатор владельца тома или пробелы
51—79	Пробелы

Формат дополнительных меток тома

Размер поля в байтах	Содержимое
0—2	Идентификатор метки тома (VOL)
3	Номер метки (2—8)
4—79	Данные пользователя

6.3.2. Метки набора данных на магнитной ленте

Описание стандартной метки формата 1

Размер поля в байтах	Содержимое
0—2	Идентификатор метки: HDR — метка начала набора данных, EOY — метка конца тома, EOF — метка конца набора данных
3	Номер метки — 1
4—20	Имя набора данных
21—26	Регистрационный номер первого тома набора данных
27—30	Порядковый номер тома для многотомного набора данных
31—34	Порядковый номер набора данных
35—38	Номер поколения набора данных
39—40	Номер версии поколения набора данных
41—46	Дата создания набора данных в форме byydd, где b — пробел, yy — год (00—99), ddd — день (001—366)
47—52	Дата истечения срока хранения набора данных в форме byydd
53	Признак разграничения доступа: F0 — доступ к набору данных не разграничивается, F1 — доступ к набору данных разрешается по предъявлении пароля, F3 — чтение набора данных разрешается без пароля
54—59	Для метки начала набора данных (HDR) — нули, для метки конца — EOF1 или EOY1, число блоков в наборе данных или на текущем томе для многотомного набора данных
60—72	Идентификатор версии операционной системы ОС ЕС
73—79	Пробелы

Описание стандартной метки формата 2

Размер поля в байтах	Содержимое
0—2	Идентификатор метки: HDR — метка начала набора данных, EOY — метка конца тома, EOF — метка конца набора данных

3	Номер метки — 2
4	Формат записей: F — фиксированной длины, V — переменной длины, U — неопределенной длины, D — переменной длины в коде КОИ-8
5—9	Длина блока
10—14	Длина логической записи
15	Плотность записи на магнитной ленте в коде ДКОИ: 0 — 8 знаков/мм, 2—32 знака/мм, 3 — 63 знака/мм
16	1 — имело место переключение томов, 0 — переключения томов не было
17—24	Имя задания
25	Косая черта (/)
26—33	Имя пункта задания
34—35	Зарезервировано
36	Код символа управления печатью: A — ДКОИ, M — машинный код, пробел — управляющий символ отсутствует
37	Зарезервировано
38	V — заблокированные записи, S — расширенные записи, R — заблокированные расширенные записи, пробел — записи не заблокированные и не расширенные
39—79	Пробелы для магнитной ленты в коде ДКОИ; в коде КОИ-8: 39—48 пробелы; 50—51 длина префикса блока; 52—79 пробелы

6.3.3. Форматы блоков оглавления тома прямого доступа

Описание DSCB формата 1

Размер поля в байтах	Содержимое
0—43	Имя набора данных в коде ДКОИ
44	F1 — идентификатор формата DSCB
45—50	Регистрационный номер первого тома набора данных
51—52	Порядковый номер тома набора данных
53—55	Дата создания набора данных в упакованном десятичном формате в форме ydd, где y — год (0—99); dd — день (1—366)
56—58	Дата истечения срока хранения набора данных в форме ydd
59	Общее число экстенгов на томе, занимаемых набором данных.
60	В это число не входит экстенс с метками пользователя
61	Число байт, использованных в последнем блоке справочника библиотечного набора данных
62—74	Зарезервировано
75—81	Идентификатор версии операционной системы ОС ЕС
82	Зарезервировано
83	Организация набора данных: X'80'—IS, X'40'—PS, X'20'—DA, X'02'—P0, X'01' — признак перемещаемости набора данных
84	Зарезервировано
85	Формат записей: X'80'—F, X'40'—V, X'C0'—U, X'20'—T (переходящие записи); X'10' — заблокированные записи; X'88'—FS; X'48'—VS; X'04' — управляющий символ в коде ДКОИ; X'02' — управляющий символ в машинном коде (нули в 5—6 разрядах означают, что управляющий символ не используется)
86—87	Содержимое байта совпадает с содержимым поля OPTCD блока DCB
88—89	Длина блока для записей формата FB и максимальная длина блока для записей формата V или U
	Для формата F — длина записи, для U — нули, для V — максимальная длина записи; для VS, если размер записей не превышает 32 756 байт, максимальная длина записи, если превышает X'8000'

90	Длина ключа
91—92	Положение ключа относительно начала логической записи индексно-последовательного набора данных
93	X'80' — последний том набора данных, X'20' — длина блока набора данных кратна 8, X'10' — доступ к набору данных разрешается по предъявлении пароля, X'14' — запись в набор данных разрешается по предъявлении пароля
94	Тип первоначального запроса памяти для набора данных: нули в 0—1 разрядах ABSTR, X'40' — в блоках, X'80' — в дорожках; X'C0' — в цилиндрах. X'08' — непрерывная память, X'04' — непрерывная память максимального размера, имеющаяся на томе; X'02' — запрос на 5 или меньше экстенгов, больших или равных заданному минимальному; X'01' — в блоках с округлением до полного цилиндра
95—97	Параметры вторичного распределения в числе блоков, дорожек или цилиндров
98—100	Относительный адрес в форме TTR последнего блока, записанного в последовательный или библиотечный набор данных
101—102	Число байт, оставшихся на дорожке после записи последнего блока в набор данных
103—104	Зарезервировано
105—114	Описание первого экстента набора данных:
105	X'00' — последующие 9 байт не описывают экстент набора данных; X'01' — экстент содержит блоки данных или относится к основной области данных индексно-последовательного набора данных; X'02' — экстент содержит область переполнения индексно-последовательного набора данных; X'04' — экстент содержит область индексов индексно-последовательного набора данных; X'40' — экстент меток пользователя; X'80' — экстент, используемый совместно несколькими наборами данных; X'81' — экстент занимает целое число цилиндров
106	Порядковый номер экстента; для экстента, состоящего из дорожки с метками пользователя — 0, для первого экстента индексно-последовательного набора данных — 1, для первого экстента остальных типов наборов данных — 0
107—110	Адрес в форме CCHN первой дорожки экстента
111—114	Адрес в форме CCHN последней дорожки экстента
115—124	Описание второго экстента набора данных (см. байты 105—114)
125—134	Описание третьего экстента набора данных (см. байты 105—114)
135—139	Адрес в форме CCHNR DSCB формата 2 для индексно-последовательного набора данных или адрес DSCB формата 3 для набора данных, состоящего более чем из трех экстенгов

Описание DSCB формата 2

Размер поля в байтах	Содержимое
0	X'02' — идентификатор DSCB формата 2
1—7	Адрес в форме MBVCCCHN первой дорожки главного индекса второго уровня
8—12	Адрес в форме CCHNR последнего активного элемента в главном индексе второго уровня
13—19	Адрес в форме MBVCCCHN первой дорожки главного индекса третьего уровня
20—24	Адрес в форме CCHNR последнего активного элемента в главном индексе третьего уровня
25—43	Зарезервировано
44	Идентификатор DSCB формата 2 — F2

45	Число уровней главного индекса
46	Число дорожек, определяющих расширение главного индекса
47—49	Адрес в форме NHR первой записи данных на каждом цилиндре
50—51	Адрес в форме НН последней дорожки данных на каждом цилиндре
52	Количество дорожек в области переполнения цилиндра на каждом цилиндре
53	Наибольший возможный номер записи на дорожке индекса высшего уровня
54	Наибольший возможный номер записи формата F или FB на дорожке основной области
55	Наибольший возможный номер записи формата F или FB на дорожке области переполнения цилиндра
56	Номер последней записи данных на дорожке, используемой также для размещения индекса дорожек
57	Наибольший возможный номер записи на дорожке, используемой для индекса дорожек
58	Наибольший возможный номер записи формата F или FB на дорожке независимой области переполнения
59—60	Количество записей, помеченных как удаленные
61—63	Счетчик числа обращений к записям переполнения, не являющимся первыми в цепочке записей переполнения
64—65	Размер области основной памяти, необходимой для размещения индекса высшего уровня
66	Число дорожек, занятых индексом высшего уровня
67—70	Количество записей в основной области
71	X'40' — необходим контроль последовательности ключей; X'20' — признак первого обращения к набору данных; X'02' — заполнен последний блок; X'01' — заполнена последняя дорожка
72—78	Адрес в форме MBVCCNN первой дорожки индекса цилиндров
79—85	Адрес в форме MBVCCNN первой дорожки главного индекса первого уровня
86—92	Адрес в форме MBVCCNN первой дорожки главного индекса высшего уровня
93—100	Адрес в форме MBVCCNNR последней записи в основной области
101—105	Адрес в форме CCNNR последнего элемента в индексе дорожек, содержащего адрес последней записи в основной области
106—110	Адрес в форме CCNNR последнего элемента в индексе цилиндров
111—115	Адрес в форме CCNNR последнего элемента в главном индексе
116—123	Адрес в форме MBVCCNNR последней записи в независимой области переполнения
124—125	Число свободных байт на текущей дорожке независимой области переполнения
126—127	Число свободных дорожек в независимой области переполнения
128—129	Число записей в областях переполнения
130—131	Число заполненных областей переполнения цилиндров
132—134	Адрес в форме NHR фиктивного элемента, указывающего на конец индекса дорожек
135—139	Адрес в форме CCNNR DSCB формата 3 или двоичные нули

Описание DSCB формата 3

Размер поля в байтах	Содержимое
0—3	3X'03' — идентификатор DSCB формата 3
4—43	Описание четырех экстенгов набора данных в формате байт 105—114 DSCB формата 1
44	X'F3' — идентификатор формата DSCB
45—134	Описание девяти экстенгов набора данных (см. байт 105—114 DSCB формата 1)
135—139	Зарезервировано

Описание DSCB формата 4

Размер поля в байтах	Содержимое
0—43	44X'04' — идентификатор DSCB формата 4
44	X'F4' — идентификатор формата DSCB
45—49	Адрес в форме CCHNR последней записи DSCB формата 1
50—51	Общее число DSCB формата 0 в оглавлении тома
52—55	Адрес в форме CCHN следующей свободной запасной дорожки
56—57	Число оставшихся запасных дорожек
58	X'80' — в оглавлении тома отсутствуют DSCB формата 5 или не отражают действительного состояния тома;
	K'08' — DSCB формата 5 и 6 отражают состояние тома, но на томе, возможно, находятся наборы данных, созданные операционной системой ДОС ЕС; X'04' — в оглавлении тома возможны ошибки
59	X'01' — оглавление тома занимает один экстенг
60—61	Зарезервировано
62—63	Число цилиндров на томе
64—65	Число дорожек в цилиндре
66—67	Емкость дорожки
68	Число байт, занимаемых полем счета, межблочными промежутками и байтами циклического контроля для записей с ключом, на каждой дорожке, кроме последней
69	Число байт, занимаемых полем счета, межблочными промежутками и байтами циклического контроля для последней записи на дорожке
70	Число, которое следует вычесть из содержимого байт 68 и 69, чтобы получить значения этих байт для записей без ключей
71	X'01' — для определения фактической длины не последнего на дорожке блока следует использовать содержимое байт 72—73 данного DSCB
72—73	Количество байт вспомогательной информации для определения фактической длины блока на дорожке
74	Число блоков DSCB, которые можно разместить на одной дорожке
75	Число блоков справочника библиотечного набора данных, которые можно разместить на одной дорожке
76—99	Зарезервировано
100—104	Адрес в формате CCHNR DSCB формата 6 или двоичные нули
105—114	Описание экстенга, выделенного для оглавления тома (см. байты 105—114 DSCB формата 1)
115—139	Зарезервировано

Описание DSCB формата 5

Размер поля
в байтах

Содержимое

0—3	4X'05' — идентификатор DSCB формата 5
4—8	Описание свободного экстенда: 4—5 — адрес первой дорожки экстенда относительно первой дорожки тома, адрес которой 0; 6—7 — число цилиндров в экстенде; 8 — число дорожек в экстенде сверх числа цилиндров
9—43	Описание свободных экстендов (см. байты 4—8)
44	X'F5' — идентификатор формата DSCB
45—134	Описание свободных экстендов (см. байты 4—8)
135—139	Адрес в форме CCHNR следующего DSCB формата 5 или двойные нули

Описание DSCB формата 6

Размер поля
в байтах

Содержимое

0—3	4X'06' — идентификатор DSCB формата 6
4—8	Описание экстенда, цилиндры которого используются совместно несколькими наборами данных: 4—5 — относительный адрес дорожки первого цилиндра экстенда; 6—7 — число совместно используемых цилиндров; 8 — число наборов данных, совместно использующих экстенд
9—34	Описание экстендов, каждый из которых используется совместно несколькими наборами данных (см. байты 4—8)
44	X'F6' — идентификатор формата DSCB
45—134	Описание экстендов, каждый из которых совместно используется несколькими наборами данных (см. байты 4—8)
135—139	Адрес в форме CCHNR следующего DSCB формата 6 или двойные нули

Макрокоманды управления данными

7.1. Общие сведения

Макрокоманды управления данными ОС ЕС являются средствами программирования на языке Ассемблера операций ввода-вывода. Расширениями макрокоманд являются цепочки команд, области данных, константы и команды SVC или перехода, с помощью которых осуществляется передача управления программам методов доступа.

7.2. Макрокоманда DCB

Макрокоманда DCB строит блок управления данными (DCB) и заполняет его поля значениями, заданными в операндах макрокоманды. Обязательными операндами DCB являются DSORG и MACRF. Информация для заполнения других полей выбирается из оператора DD, меток набора данных или заносится в блок DCB обрабатывающей программой. Операнды макрокоманды DCB — ключевые, разделяются между собой запятыми и могут располагаться в любом порядке. Операнды и их значения, необходимые для обработки набора данных, определяются его организацией и методом доступа.

7.2.1. Операнды макрокоманды DCB для BISAM

В макрокоманде DCB для BISAM используются следующие операнды:

BFALN=x, где x принимает одно из следующих значений:

F — начало каждого буфера на границе слова, которая не является границей двойного слова;

D — начало каждого буфера на границе двойного слова.

Если этот операнд опущен, то предполагается, что буфер начинается на границе двойного слова.

BUFCB=адрес — адрес блока управления буферным пулом. Этот операнд задается только в том случае, если буферный пул создается с помощью макрокоманды BULD.

BUFL=p, где p — длина в байтах каждого буфера в буферном пуле (не более 32 760). Этот операнд задается только в том случае,

если буферный пул создается с помощью макрокоманды BUILD или GETPOOL.

BUFNO= n , где n — количество буферов в буферном пуле ($n \leq 255$). Этот операнд является обязательным, если буферный пул создается с помощью макрокоманды BUILD.

DDNAME=имя — имя оператора DD для обрабатываемого набора данных. Соответствующее поле в блоке DCB должно быть заполнено до выдачи макрокоманды OPEN.

DSORG=IS — индексно-последовательная организация набора данных.

EXLST=адрес — адрес списка выходов. В список включаются адреса программ обработки меток пользователя, модификации блока DCB, обработки итоговых данных, конца тома, счетчика блоков и т. д.

MACRF= x — макрокоманды и средства, используемые для обработки набора данных; x представляет собой одно из выражений: (R[S][C]); (W[U][A][C]) или (R[U][S][C], W[U][A][C]), которые определяют:

A — добавление записи в набор данных;

C — использование макрокоманды CHECK для контроля выполнения операций ввода-вывода;

R — использование макрокоманды READ;

S — динамическую буферизацию;

U — обновление записей;

W — использование макрокоманды WRITE.

Вместе с W обязательными являются параметры U, A или оба.

MSHI=адрес — адрес области основной памяти, в которую управляющей программой помещается индекс самого высокого уровня для обрабатываемого набора данных. Этот операнд не является обязательным, однако его задание сокращает время обработки.

MSWA=адрес — адрес области основной памяти, используемой управляющей программой для более эффективной работы при добавлении в набор данных записей фиксированной длины.

NCP= n , где n — максимальное число макрокоманд READ или WRITE, которые могут быть выданы перед выдачей макрокоманд CHECK или WAIT ($n \leq 99$). Если этот операнд опущен, то его значение полагается равным 1.

SMSI= n , где n — размер области основной памяти, которая выделена для хранения индекса самого высокого уровня для обрабатываемого набора данных ($n \leq 65535$). Этот операнд задается в том случае, если задан операнд MSHI.

SMSW= n , где n — размер области основной памяти, адрес которой определен операндом MSWA ($n \leq 65535$). Размер области должен быть достаточным для размещения области счета, данных в случае заблокированных записей одной логической записи.

SYNAD=адрес — адрес программы пользователя, получающей управление в случае возникновения постоянной ошибки ввода-вывода.

7.2.2. Операнды макрокоманды DCB для BDAM

В макрокоманде DCB для BDAM используются следующие операнды:

BFALN=x (см. 7.2.1).

BFTEK=R — определяет запрос на интерфейс логической записи для записей формата VS.

BLKSIZE=n, где n — максимальный размер блока в байтах ($n \leq 32760$).

BUFCB=адрес (см. 7.2.1).

BUFL=n, где n — длина в байтах каждого буфера в буферном пуле, получаемого автоматически или динамически ($n \leq 32760$). Этот операнд обязателен в случае использования динамической буферизации. Если в макрокоманде READ или WRITE в качестве значения операнда, определяющего адрес ключа, указано 'S', то в длину буфера следует включить и длину ключа.

BUFNO=n (см. 7.2.1).

DDNAME=имя (см. 7.2.1).

DSORG=DA или DAU при обработке существующего набора данных с прямой организацией и PS или PSU — при его создании. Параметр U указывает на неперемещаемость набора данных.

EXLST=адрес (см. 7.2.1).

KEYLEN=n, где n — длина ключей записей в обрабатываемом наборе данных ($n \leq 255$).

LIMCT=n, где n — число блоков или дорожек на устройстве прямого доступа, определяющее область расширенного поиска при выполнении макрокоманд READ или WRITE (параметр E в операнде OPTCD). Поиск начинается с начала дорожки, заданной в макрокоманде READ или WRITE, и заканчивается, когда будет просмотрена указанная область либо найден требуемый блок или свободное место в наборе данных. Если n превышает размер области набора данных, то поиск продолжается от его начала. Операнд LIMCT игнорируется, если в операнде OPTCD не задан параметр E. Если $n=0$ и в операнде OPTCD задан параметр E, то операция ввода-вывода не выполняется, а в блоке DECB устанавливается признак, указывающий причину.

MACRF=x — макрокоманды и средства, используемые для обработки набора данных; x представляет собой одно из выражений: R[K][I][X][S][C], W[A][K][I][C] или (R[K][I][X][S][C], W[A][K][I][C]). Одна из комбинаций параметров A, K, I должна указываться обязательно.

Значения параметров определяют:

A — добавление блока в набор данных;

C — использование макрокоманды CHECK для контроля выполнения операции ввода-вывода (если не задано, то предполагается WAIT);

I — поиск по идентификатору блока;

K — поиск по ключу;

R — использование макрокоманды READ;

- S — динамическую буферизацию;
- W — использование макрокоманды WRITE;
- X — монопольное управление блоком.

OPTCD=x — необязательные средства, предоставляемые управляющей программой. Соответствующее поле в блоке DCB должно быть заполнено до выдачи макрокоманды OPEN. В качестве значения параметра x может быть задана любая комбинация, составленная из символов W, E, F, R, A, причем два последних символа — взаимоисключающие.

Каждый из перечисленных символов определяет:

- A — форму адреса блока, возвращаемого макрокомандами READ или WRITE;
- E — расширенный поиск; игнорируется, если находится в одной комбинации с символом A;
- F — форму адреса блока, возвращаемого макрокомандами READ и WRITE при определенной обратной связи. Форма адреса блока должна быть аналогична той, которая задана в этих макрокомандах. Если символ F опущен, то возвращается физический адрес блока;
- R — относительную адресацию блоков в макрокомандах READ и WRITE;
- W — проверку правильности операций записи.

Если символы A и R опущены, то предполагается, что в макрокоманде READ или WRITE задан относительный адрес дорожки;

RECFM=x — формат записей набора данных, где x представляет собой одно из следующих выражений: V, VS, VBS, U, F или FT.

Значения параметра x определяют:

- V — записи переменной длины;
- VS — расширенные записи переменной длины;
- VBS — заблокированные расширенные записи переменной длины;
- U — записи неопределенной длины;
- F — записи фиксированной длины;
- FT — переходящие записи фиксированной длины;

SYNAD=адрес (см. 7.2.1).

7.2.3. Операнды макрокоманды DCB для BRAM

В макрокоманде DCB для BRAM используются следующие операнды:

BFALN=x (см. 7.2.1);

BLKSIZE=n (см. 7.2.2);

BUFCB=адрес (см. 7.2.1);

BUFLN=n, где n — длина в байтах каждого буфера в буферном пуле ($n \leq 32760$). Если этот операнд опущен и буферы выделяются автоматически, то для определения длины буфера используются значения операндов BLKSIZE и KEYLEN;

BUFNO=n, где n — число буферов в буферном пуле ($n \leq 255$).

Этот операнд обязателен, если буферный пул создается с помощью макрокоманды BUILD или автоматически.

DDNAME=имя (см. 7.2.1).

DSORG=PO или POU — библиотечная организация набора данных.

EODAD=адрес — определяет адрес программы, получающей управление при достижении конца раздела библиотечного набора данных. Адрес этой программы следует поместить в соответствующее поле блока DCB до того, как будет достигнут конец набора данных.

EXLST=адрес (см. 7.2.1).

KEYLEN=n (см. 7.2.2).

LRECL=n, где n — длина логической записи в байтах ($n \leq 32760$). Этот операнд задается только для записей форматов F и FB.

MACRF=x — определяет макрокоманды управления операциями ввода-вывода для обработки библиотечного набора данных; x принимает одно из следующих значений: R, W или (R, W).

Значения параметра x определяют:

R — использование макрокоманды READ;

W — использование макрокоманды WRITE.

При всех значениях параметра подразумевается, что адрес блока в наборе данных определяется с помощью макрокоманды NOTE, а установка устройства в положение для считывания или записи блока производится с помощью макрокоманды POINT.

NCP=n (см. 7.2.1).

OPTCD=x — необязательные средства, предоставляемые управляющей программой. Соответствующее поле в блоке DCB должно быть заполнено до выдачи макрокоманды OPEN. Параметр x принимает одно из следующих значений: W, C или WC.

Значения параметра x определяют:

C — планирование цепочками;

W — проверку правильности операций записи.

RECFM=x — формат записей набора данных. Параметр x представляет собой одно из выражений:

$$V[T] \left[\left\{ \begin{matrix} A \\ M \end{matrix} \right\} \right]; \left\{ \begin{matrix} V \\ F \end{matrix} \right\} [B][T] \left[\left\{ \begin{matrix} A \\ M \end{matrix} \right\} \right]; U[T] \left[\left\{ \begin{matrix} A \\ M \end{matrix} \right\} \right]$$

Значения параметра x определяют:

A — в первом байте каждой логической записи содержится управляющий знак в коде ДКОИ;

B — логические записи объединены в блоки;

F — логические записи фиксированной длины;

M — в первом байте каждой логической записи содержится управляющий знак в машинном коде;

T — переходящие записи;

U — записи неопределенной длины;

V — записи переменной длины.

SYNAD=адрес (см. 7.2.1).

7.2.4. Операнды макрокоманды DCB для BSAM

В макрокоманде DCB для BSAM используются следующие операнды:

BFALN=x (см. 7.2.1).

BFTEK=R — определяет запрос на интерфейс логической записи для записей формата VS при создании набора данных с прямой организацией. При обработке записей формата VS входного набора данных с ключами определяет смещение на длину ключа в буфере сегментов без ключей.

BLKSIZE=n (см. 7.2.2).

BUFCB=адрес (см. 7.2.1).

BUFL=n (см. 7.2.3).

BUFNO=n (см. 7.2.3).

BUFOFF=n, где n — длина префикса блока. Префикс блока используется при обработке записей набора данных в коде КОИ-8. Параметр n принимает значения от 0 до 99 или L. BUFOFF=L задается при обработке записей формата D (записи переменной длины в коде КОИ-8). При обработке выходного набора данных с записями формата F, FB или U в качестве значения параметра n следует указывать 0.

Соответствующее поле в блоке DCB должно быть заполнено до выдачи макрокоманды OPEN.

DDNAME=имя (см. 7.2.1).

DEVD=x — тип периферийного устройства обрабатываемого набора данных; x может принимать одно из следующих значений: DA, TA, PT, PR, PC, RD.

Значения параметра x определяют:

DA — устройство прямого доступа;

TA — накопитель на магнитной ленте;

PT — перфоленточное устройство ввода или вывода;

PR — печатающее устройство;

PC — перфокарточное устройство вывода;

RD — перфокарточное устройство ввода.

Любое из перечисленных в указанном выше порядке значений параметра x определяет и типы устройств, следующих за ним. Если обрабатываемый набор данных вводится с помощью системной программы ввода или выводится с помощью системной программы вывода, то операнд DEVD следует либо опустить, либо задать DEVD=DA. Если операнд DEVD опущен, то полагается DEVD=DA. Операнд DEVD может быть задан только в макрокоманде DCB.

Если DEVD=DA, то можно дополнительно задать операнд KEYLEN.

Если DEVD=TA, то дополнительно можно задать операнд DEN=x, определяющий плотность записи на магнитной ленте; x может принимать значения 0, 2 или 3, которые соответственно определяют следующие плотности записи: 0—8 бит/мм; 2—32 бит/мм, 3—63 бит/мм.

Если $DEVD=PT$, то дополнительно можно задать операнды $CODE=x$, $PARI=x$ и $EORC=p$.

$CODE=x$ — определяет преобразование входных данных из кода, определенного параметром x , в код ДКОИ, а при выводе — из кода ДКОИ в код, определенный этим параметром. Для перфоленточного устройства ввода этот параметр принимает значения I, F, B, C, A, T, K, L, M, P, N, U, для перфоленточного устройства вывода — A, K, P, U, N. Перечисленные значения параметра x определяют следующие коды: I—IBM BCD, F—FRIDEN, B—BURROUGHS, C—NCR, A—КОИ-7 с добавлением до четного числа пробивок, T—TELETYPE, K—КОИ-7 (без контрольной пробивки), L—SE5L, M—S528, P—КОИ-7 (с добавлением до нечетного числа пробивок), N — без преобразования, U — код пользователя. Если операнд $CODE$ опущен, то полагается $CODE=N$.

$PARI=x$ — определяет контроль данных для устройств EC-7902R и EC-7902P.

Параметр x принимает значения E или U;

E — контроль на четное число пробивок;

U — ввод данных с контролем на нечетное число пробивок и вывод данных без контроля.

Если операнд $PARI=x$ опущен, то ввод и вывод данных осуществляются без контроля.

$EORC=pp$ — код символа конца блока при обработке записей формата U для устройства EC-7902P, pp — любое шестнадцатеричное число от 00 до FF.

Если $DEVD=PR$, то дополнительно можно задать операнд $PRTSP=x$.

$PRTSP=x$ — интервал между строками при выводе на печать; x принимает значения 0, 1, 2 или 3 строки. Этот операнд можно задавать только в том случае, если в операнде $RECFM$ не указан параметр A или M. Если последнее условие выполнено и операнд $PRTSP$ опущен, то величина интервала между строками принимается равной 1.

Если $DEVD=PC$ или RD , то дополнительно можно задать операнды $MODE$ и $STACK$.

$$MODE = \begin{Bmatrix} C \\ E \end{Bmatrix} \begin{Bmatrix} O \\ R \end{Bmatrix}$$

Параметры C, E, O, R определяют:

C — ввод и вывод образа перфокарт;

E — ввод и вывод данных в коде ДКОИ;

O — обработка пометок (для EC-6015);

R — ввод перфокарт с исключением колонок (для EC-6015).

Если операнд $MODE$ опущен, то полагается $MODE=E$.

$STACK=x$ — номер приемного кармана для перфокарт; x принимает значения 1 или 2. Если этот операнд опущен, то полагается $STACK=1$.

$DSORG=PS$ или PSU — определяет последовательную организацию обрабатываемого набора данных. Параметр U указывает на непеременяемость набора данных.

EODAD=адрес — адрес программы, получающей управление при достижении конца набора данных. Если RECFM=FS или FBS, то программа обработки условия конца набора данных получает управление также при считывании укороченного блока. При обработке многотомного набора данных в режиме UPDAT (задается макрокомандой OPEN) программа обработки условия конца набора данных должна выдать макрокоманду FEOV, если требуется переключение томов.

EXLST=адрес (см. 7.2.1);

KEYLEN=n (см. 7.2.2).

LRECL=n — длина логической записи в байтах. Если RECFM=FB, то значение параметра n операнда LRECL должно быть кратно значению параметра операнда BLKSIZE. Если RECFM=F, то значение параметра n операнда LRECL должно быть равно значению параметра операнда BLKSIZE. Если RECFM=VBS, то длина логической записи может превышать размер блока. Если RECFM=VS или VBS и длина логической записи превышает 32756 байт, то следует задать LRECL=X. Если RECFM=V или VB, то в качестве значения параметра n операнда LRECL следует указать максимальную длину. Если операнд LRECL опущен, то используется операнд BLKSIZE.

MACRF=x — макрокоманды управления операциями ввода-вывода для обработки последовательного набора данных. Параметр x принимает следующие значения: RC, RP, WC, WP, WL, (R[C], W[C]), (R[P], W[P]).

Значения параметра x определяют:

C — CNTRL;

L — создание набора данных с прямой организацией;

P — POINT (подразумевается NOTE);

R — READ;

W — WRITE.

NCP=n (см. 7.2.1).

OPTCD=x[T] — необязательные средства, предоставляемые управляющей программой. Соответствующее поле в блоке DSB должно быть заполнено до выдачи макрокоманды OPEN. Параметр x принимает одно из значений: C, W, WC, Z, ZC, Q, U или UC.

Значения параметра x определяют:

C — планирование цепочками. В этом случае нельзя задавать BFTEK=R;

W — проверка правильности операции записи (только для устройств прямого доступа);

Z — выполнение укороченной процедуры исправления ошибки (только при вводе с магнитной ленты);

Q — преобразование записей входного набора данных на магнитной ленте из кода КОИ-8 в код ДКОИ либо преобразование записей выходного набора данных на магнитной ленте из кода ДКОИ в код КОИ-8;

U — отмена подавления признака ошибки в данных для печатающих устройств со сменным набором символов.

Подпараметр Т определяет сохранение итоговых данных пользователя.

$RECFM=x_1[x_2][x_3]$ — формат записей набора данных. Параметр x_1 принимает значения U, V, F или D:

U — записи неопределенной длины;

V — записи переменной длины;

F — записи фиксированной длины;

D — записи переменной длины в коде КОИ-8.

Параметры x_2 и x_3 принимают значения A, B, M, S или T.

A — в первом байте каждой логической записи содержится управляющий знак в коде ДКОИ;

B — логические записи объединены в блоки;

M — в первом байте каждой логической записи содержится управляющий знак в машинном коде;

S — стандартные блоки, если $x_1=F$; расширенные записи переменной длины, если $x_1=V$;

T — переходящие записи. В этом случае нельзя задавать планирование цепочками ($OPTCD=C$).

Если параметр x_1 принимает значение U, то $x_2=T$, а $x_3=A$ или M. Если параметр x_1 принимает значение V или F, то $x_2=B$, S, T, BS или BT, а $x_3=A$ или M. Если x_1 принимает значение D, то $x_2=B$, а $x_3=A$. Во всех случаях параметры x_2 и x_3 или один из них могут быть опущены.

Формат $RECFM=V$ не используется для набора данных в коде КОИ-8 и перфокарточного устройства ввода.

SYNAD=адрес (см. 7.2.1).

7.2.5. Операнды макрокоманды DCB для QISAM

В макрокоманде DCB для QISAM используются следующие операнды:

BFALN= x (см. 7.2.1).

BLKSIZE= n (см. 7.2.2). Операнд BLKSIZE задается только при создании набора данных с индексно-последовательной организацией.

BUFCB=адрес (см. 7.2.1).

BUFL= n (см. 7.2.1).

BUFNO= n (см. 7.2.1).

CYLOFL= n — число дорожек на каждом цилиндре основной области, резервируемых для области переполнения цилиндра.

DDNAME=имя (см. 7.2.1).

DSORG=IS или ISU — индексно-последовательная организация набора данных. Параметр ISU указывает на неперемещаемость набора данных. Этот параметр можно задавать только при создании набора данных.

EODAD=адрес — адрес программы, получающей управление при достижении конца набора данных в процессе выполнения операций ввода с помощью макрокоманды GET.

KEYLEN= n (см. 7.2.2).

$LRECL=n$ — длина логических записей создаваемого набора данных с индексно-последовательной организацией (см. 7.2.4).

$MACRF=x$ — макрокоманды управления операциями ввода-вывода для обработки набора данных. Параметр x принимает следующие значения:

$(PM);(PL); (GM[S[\left\{\begin{smallmatrix} K \\ I \end{smallmatrix}\right\}]] (GL[S[\left\{\begin{smallmatrix} K \\ I \end{smallmatrix}\right\}]]L,PU))$

Значения параметра x определяют:

GL — GET (подразумевается RELSE и ESETL). Адрес буфера предоставляется управляющей программой (режим указания);

GM — GET (подразумевается RELSE и ESETL). Запись пересылается в рабочую область (режим пересылки);

I — поиск записи по физическому адресу;

K — поиск по ключу или записи;

PL — PUT в режиме указания (создание набора данных);

PM — PUT в режиме пересылки (создание набора данных);

PU — PUTX;

S — SETL.

$NTM=n$ — число дорожек, занимаемых индексом цилиндра, для которых образуется элемент в главном индексе самого низкого уровня. Этот операнд задается, если используется главный индекс ($OPTCD=M$). Если операнд NTM опущен, то операнд $OPTCD=M$ игнорируется, и главный индекс не создается.

$OPTCD=x$ — необходимые средства, предоставляемые управляющей программой. Соответствующее поле в блоке DCB должно быть заполнено до выдачи макрокоманды OPEN. В качестве значения параметра x может быть задана любая комбинация, составленная из символов W, M, Y, I, L, R, U, которые определяют:

I — использование независимой области переполнения для записей переполнения;

L — фактическое удаление записей, отмеченных как удаленные, если требуется место для новых записей. Для заблокированных записей это средство не используется, если задан операнд $RKP=0$;

M — создание главного индекса;

R — занесение в блок DCB статистической информации (поля RORG1, RORG2 и RORG3);

U — запись элементов индекса дорожки группой. Эта возможность допустима только при обработке записей фиксированной длины;

W — проверку правильности каждой операции записи;

Y — использование области переполнения цилиндра для записей переполнения.

$RECFM=x$ — формат записей выходного набора данных.

Параметр x принимает следующие значения: F, FB, V или VB, которые определяют:

F — запись фиксированной длины;

FB — заблокированные записи фиксированной длины;

V — записи переменной длины;

VB — заблокированные записи переменной длины.

Если этот операнд опущен, то полагается $RECFM=V$.

$RKP=n$ — позиция первого байта ключа относительно начала логической записи выходного набора данных. При $RECFM=V$ или VB значение параметра n должно быть больше или равно 4, если в значение параметра операнда $OPTCD$ не включен символ L , и больше или равно 5, если включен. Если операнд RKP опущен, то полагается $RKP=0$.

$SYNAD$ — адрес (см. 7.2.1).

7.2.6. Операнды макрокоманды DCB для QSAM

В макрокоманде DCB для QSAM используются следующие операнды:

$BFALN=x$ (см. 7.2.1).

$BFTEK=x$ — определяет тип буферизации.

Параметр x принимает значения S , E или A , которые определяют:

S — простую буферизацию;

E — обменную буферизацию;

A — запрос на интерфейс логической записи для записей формата VS при обработке набора данных в режиме указания.

Если операнд $BFTEK$ опущен, то полагается $BFTEK=S$.

$BLKSIZE=n$ (см. 7.2.3).

$BUFCB$ — адрес (см. 7.2.1).

$BUFL=n$ (см. 7.2.1).

$BUFNO=n$ (см. 7.2.3).

Для QSAM последние три операнда задаются также в том случае, если буферный пул создается с помощью макрокоманды $BUILDRCB$.

$BUFOFF=n$ (см. 7.2.4).

$DDNAME$ — имя (см. 7.2.1).

$DEV D=x$ (см. 7.2.4).

$DSORG=PS$ или PSU (см. 7.2.4).

$EODAD$ — адрес (см. 7.2.5).

$EROPT=x$ — действия, предпринимаемые управляющей программой, если возникла постоянная ошибка ввода-вывода и не задан операнд $SYNAD$ либо в программе, адрес которой определяется операндом $SYNAD$, была выдана макрокоманда $RETURN$.

Параметр x принимает значения ACC , SKP , ABE , которые позволяют:

ACC — принять последний обрабатываемый блок;

SKP — пропустить последний обрабатываемый блок;

ABE — закончить задачу аварийно.

Выбор необходимого значения зависит от значений операндов макрокоманды $OPEN$ для обрабатываемого набора данных:

ACC — если значения операндов макрокоманды OPEN—INPUT, OUTPUT (только вывод на печать), RDBACK или UPDAT; SKP, если значения операндов макрокоманды OPEN—INPUT, RDBACK или UPDAT; ABE — если значения операндов макрокоманды OPEN—INPUT, OUTPUT, RDBACK или UPDAT.

EXLST=адрес (см. 7.2.1).

LRECL=n (см. 7.2.4).

MACRF=x — макрокоманды и средства, используемые для обработки набора данных. Параметр x принимает значения (Gx₁[C]); (Px₁[C]) или (Gx₁[C], Px₁[C]), где x₁ в свою очередь принимает значения M, L, T или D.

Значения параметров определяют:

C — CNTRL;

D — режим данных (только для записей форматов VS или VBS);

G — GET;

L — адрес записи предоставляется управляющей программой (режим указания);

M — запись пересылается в рабочую область программы (режим пересылки);

P — PUT или PUTX;

T — режим подстановки.

OPTCD=x (см. 7.2.4).

RECFM=x₁[x₂][x₃] (см. 7.2.4). Кроме перечисленных в 7.2.4, для QSAM возможны комбинации VST и VBST.

SYNAD=адрес (см. 7.2.1).

7.3. Макрокоманды организации обработки данных

7.3.1. Макрокоманда CHECK

В операционной системе ОС ЕС обработка данных выполняется базисными методами доступа BDAM, BISAM, BPAM, BSAM и методами доступа с очередями QISAM, QSAM. Для синхронизации операций ввода-вывода с выполнением программы, использующей макрокоманды базисных методов доступа, служит макрокоманда CHECK или WAIT.

Макрокоманда CHECK переводит выполняемую программу в состояние ожидания до тех пор, пока не завершится операция ввода-вывода, и, если операция завершается успешно, возвращает управление следующей за ней команде. В противном случае управление получает программа обработки ошибок, адрес которой указан в операнде SYNAD макрокоманды DCB. Формат макрокоманды:

CHECK адрес-decb[,DSORG=код]

адрес-decb — адрес блока DECB, образованного соответствующей макрокомандой READ или WRITE;

DSORG=код — определяет метод доступа;

параметр «код» принимает значения: IS—BISAM; ALL—BDAM; BISAM, BPAM или BSAM.

Если операнд DSORG опущен, то предполагается, что обработка данных выполняется с помощью BDAM, BPAM или BSAM.

Макрокоманда WAIT описана в 5.2.1.

Синхронизация операций ввода-вывода с выполнением программы, использующей макрокоманды методов доступа с очередями, выполняется автоматически.

7.3.2. Макрокоманда CLOSE

Макрокоманда CLOSE выполняет заключительные операции обработки наборов данных и устанавливает тома, на которых они находятся, в положения, определенные программистом с помощью операндов макрокоманды и операторов DD. При этом макрокоманда CLOSE создает метки выходного набора данных, восстанавливает в исходное состояние поля блоков DCB, освобождает все области основной памяти, полученные при выдаче макрокоманды OPEN, кроме области буферного пула. Для освобождения последней области после макрокоманды CLOSE следует выдать макрокоманду FREEPOL. Это следует сделать обязательно, если предполагается повторное открытие DCB с другим размером блока логической записи. В одной макрокоманде CLOSE можно указать любое количество блоков DCB.

Существуют три формы записи макрокоманды CLOSE: стандартная, описательная и исполнительная. Формат макрокоманды в стандартной форме:

[метка] CLOSE (адрес-dcb,[дисп],...)[,TYPE=T]

где адрес-dcb — адрес блока управления данными;

дисп — определяет установку тома обрабатываемого набора данных. Этот операнд может принимать значения REREAD, LEAVE, REWIND, DISP.

Значения операнда определяют:

REREAD — установка тома в начало обрабатываемого набора данных;

LEAVE — установка тома в логический конец набора данных;

REWIND — установка тома в точку загрузки;

DISP — установка тома в положение, указываемое значением параметра DISP оператора DD, причем PASS соответствует LEAVE, DELETE—REWIND, а KEEP, CATLG, UNCATLG вызывают перемотку и разгрузку тома.

Если операнд «дисп» опущен, то при TYPE=T подразумевается LEAVE, в противном случае — DISP.

TYPE=T применяется только для BSAM. Этот операнд вызывает образование меток набора данных и установку тома в соответствии со значением операнда «дисп». Однако поля блока DCB не восстанавливаются, за исключением нулевого разряда поля

DCBOFLGS, который сбрасывается. Операнд TYPE=T позволяет продолжить обработку набора данных без повторной выдачи макрокоманды OPEN. Этот операнд нельзя указывать при выдаче макрокоманды CLOSE в программе, адрес которой определяется операндом SYNAD макрокоманды DCB.

В формат макрокоманды CLOSE в описательной форме добавляется операнд MF=L. В качестве значения операнда адрес-dcb может быть задан любой адрес, допустимый в адресной константе типа A.

В формат макрокоманды CLOSE в исполнительной форме добавляется операнд MF=(E, адрес-списка-параметров). В качестве значений операндов, определяющих адреса областей основной памяти, может быть задан любой адрес, допустимый в командах типа RX, или любой из регистров 2—12, предварительно загруженный нужным адресом. Для указания адреса списка параметров можно использовать также регистр 1.

7.3.3. Макрокоманда DCBD

Макрокоманда DCBD позволяет обращаться к полям блока DCB по символическим адресам. Расширением этой макрокоманды является фиктивная программная секция, представляющая собой образ блока DCB. Имя каждого поля в этой программной секции состоит из двух частей: DCB и имени соответствующего операнда макрокоманды DCB. Формат макрокоманды:

[метка] DCBD [DSORG=(тип-организации)][,DEVD=(тип-устройства)]

где DSORG=тип-организации — типы блоков DCB в зависимости от организации обрабатываемых наборов данных. В качестве значений параметра операнда DSORG может быть указано одно или разделенные запятыми несколько из следующих значений: BS, DA, IS, LR, PO, PS, QS.

Названные значения параметра определяют:

- BS — блок DCB для обработки последовательного набора данных посредством BSAM;
- DA — блок DCB для обработки набора данных посредством BDAM;
- IS — блок DCB для обработки индексно-последовательного набора данных;
- LR — только поле длины логической записи (DCBLRECL);
- PO — блок DCB для обработки библиотечного набора данных;
- PS — блок DCB для обработки последовательного набора данных посредством QSAM или BSAM;
- QS — блок DCB для обработки последовательного набора данных посредством QSAM.

Если операнд DSORG опущен, то операнд DEVD игнорируется и в расширении макрокоманды DCBD будут определены только поля основного сегмента блока DCB;

DEVD=тип-устройства — типы периферийных устройств, на которых может находиться обрабатываемый набор данных. В качестве значений параметра операнда DEVD может быть указано одно или разделенные запятыми несколько из следующих значений: DA, PC, PR, PT, RD, TA.

Названные значения параметра определяют:

DA — устройство прямого доступа;

PC — перфокарточное устройство вывода;

PR — печатающее устройство;

PT — перфоленточное устройство ввода или вывода;

RD — перфокарточное устройство ввода;

TA — накопитель на магнитной ленте.

Если операнд DEVD опущен, а в качестве значения операнда DSORG указано BS, QS или PS, то в расширении макрокоманды будут определены поля для всех перечисленных типов периферийных устройств.

7.3.4. Макрокоманда FEOV

Макрокоманда FEOV используется при обработке наборов данных посредством BSAM или QSAM для принудительного завершения обработки текущего тома и автоматического переключения на новый том. Если макрокоманда FEOV выдана для последнего тома входного набора данных, то управление получает программа, адрес которой указан в операнде EODAD макрокоманды DCB. При обработке набора данных посредством BSAM перед выдачей макрокоманды FEOV следует проверить на завершение все ранее инициированные операции ввода-вывода. Формат макрокоманды:

[метка] FEOV адрес-dcb[,дисп]

где адрес-dcb — адрес блока DCB обрабатываемого набора данных;

дисп — диспозиция тома магнитной ленты по завершении обработки.

Операнд «дисп» принимает значения REWIND или LEAVE, которые определяют:

REWIND — установку тома в точку загрузки;

LEAVE — установку тома в позицию логического конца набора данных.

7.3.5. Макрокоманда OPEN

Макрокоманда OPEN завершает заполнение полей блока DCB, устанавливает соответствие заданных параметров, загружает в основную память программы управления операциями ввода-вывода, выделяет буферные области, осуществляет установку тома для обработки набора данных и определяет диспозицию текущего тома при переключении томов. В одной макрокоманде OPEN можно указать любое количество блоков DCB и параметров обработки наборов данных.

Существуют три формы записи макрокоманды OPEN: стандартная, описательная и исполнительная. Формат макрокоманды в стандартной форме:

[метка] OPEN (адрес-dcb,[(режим,дисп)],...)

где адрес-dcb — адрес блока DCB обрабатываемого набора данных;

режим — режим обработки набора данных;

дисп — диспозиция текущего тома при переключении томов.

При создании набора данных значением операнда «режим» может быть только OUTPUT или OUTIN. Если операнд «режим» опущен, то подразумевается INPUT. Если операнд «дисп» опущен, то подразумевается DISP. Если необходимо указать операнд «дисп», то следует также определить операнд «режим». Оба операнда игнорируются при обработке набора данных посредством QISAM в режиме сканирования и BISAM. Возможные значения операндов перечислены в табл. 7.1.

Таблица 7.1

Значения операндов «режим» и «дисп» макрокоманды OPEN

Метод доступа	Тип периферийного устройства					
	магнитная лента		устройство прямого доступа		другие типы периферийных устройств	
	режим	дисп	режим	дисп	режим	дисп
QSAM	INPUT OUTPUT RDBACK	REREAD LEAVE DISP	INPUT OUTPUT UPDAT	REREAD LEAVE DISP	INPUT OUTPUT	—
BSAM	INPUT OUTPUT INOUT OUTIN RDBACK	REREAD LEAVE DISP	INPUT OUTPUT INOUT OUTIN UPDAT	REREUD LEAVE DISP	INPUT OUTPUT	—
QISAM (режим загрузки)	—	—	OUTPUT	—	—	—
BPAM, BDAM	—	—	INPUT OUTPUT UPDAT	—	—	—

Указанные в таблице значения операнда «режим» определяют:
INPUT — входной набор данных;

INOUT — обработка набора данных сначала в режиме INPUT, а затем без повторной выдачи макрокоманды OPEN в режиме OUTPUT. Если в операторе DD задан операнд LABEL=(„IN), то набор данных обрабатывается в режиме INPUT;

OUTPUT — выходной набор данных. Для BDAM режим OUTPUT эквивалентен режиму UPDAT;

OUTIN — обработка набора данных сначала в режиме OUTPUT, а затем без повторной выдачи макрокоманды OPEN в режиме INPUT. Если в операторе DD задан операнд LABEL = („OUT), то набор данных обрабатывается в режиме OUTPUT;

RDBACK — установка тома входного набора данных для обработки в обратном направлении;

UPDAT — режим обновления набора данных на месте.

Приведенные в таблице значения операнда «дисп» определяют:

LEAVE — при переключении томов сохранение положения текущего тома;

REREAD — установка текущего тома в исходное положение;

DISP — установка тома в положение, определяемое операндом DISP оператора DD:

PASS — установка тома в положение конца набора данных;

DELETE — перемотка текущего тома;

KEEP, CATLG или UNCATLG — перемотка и разгрузка текущего тома.

В формат макрокоманды OPEN в описательной форме добавляется операнд MF=L. В качестве значений операнда адрес-dcb может быть задан любой адрес, допустимый в адресной константе типа A.

В формат макрокоманды OPEN в исполнительной форме добавляется операнд MF=(E, адрес-списка-параметров). В качестве значений операндов, определяющих адреса областей основной памяти, в этом случае может быть задан любой адрес, допустимый в командах типа RX, или любой из регистров 2—12, предварительно загруженный нужным адресом. Для указания адреса списка параметров можно использовать также регистр 1.

7.4. Макрокоманды управления буферами

Буферами в операционной системе ОС ЕС ЭВМ называются области основной памяти, используемые во время выполнения операций обмена данными между периферийными устройствами и основной памятью. Программа пользователя ЭВМ может получить буферы автоматически в составе буферного пула с помощью макрокоманд BUILD, BUILDRCDD, GETPOOL и динамически с помощью макрокоманды READ.

Автоматически буферы выделяются программе макрокомандой OPEN. Операнды BUFNO, BUFL, BFALN и BUFCB макрокоманды DCB должны быть закодированы в соответствии с требованиями методов обработки данных. При обработке данных методами доступа с очередями операнд BUFCB должен быть опущен, BUFNO, BFALN, BUFL — необязательны. Однако если, последний операнд опущен, то длина буфера принимается для QSAM равной значению параметра операнда BLKSIZE, а для QISAM независимо от значения BUFL — сумме значений параметров операндов BLKSIZE и KEYLEN.

При обработке данных базисными методами доступа операнд BUFCSB должен быть опущен, BUFNO — необязателен для всех методов доступа, кроме BSAM, BFALN — необязателен, BUFL для BSAM и BDAM — необязателен, для BISAM — игнорируется.

Для всех методов доступа, если операнд BUFNO отсутствует, выделяется не менее двух буферов.

Если буферный пул строится с помощью макрокоманд BUILD и BUILDRCDB, то операнды BUFL и BFALN макрокоманды DCB игнорируются, операнд BUFNO определяет число буферов, выделяемых программе из этого буферного пула, а операнд BUFCSB — адрес буферного пула.

7.4.1. Макрокоманда BUILD

Макрокоманда BUILD строит буферный пул в области основной памяти, резервируемой программистом. Отдельные буферы из буферного пула можно получить затем с помощью макрокоманды GETBUF. Макрокоманду BUILD можно использовать с любым методом доступа. Формат макрокоманды:

[метка] BUILD адрес {число, длина-буфера }
(0)

где адрес — адрес области основной памяти, выровненный по границе слова;

число — требуемое число буферов (до 255);

длина-буфера — длина каждого буфера в байтах (до 32 760). Значение операнда должно быть кратно 4, а для QSAM — не меньше размера блока, определенного в макрокоманде DCB;

(0) — общий регистр 0. Перед выдачей макрокоманды в разряды 0—15 регистра 0 следует загрузить число буферов, а в разряды 16—31 — длину каждого буфера. Размер требуемой области основной памяти вычисляется по формуле: размер области = $8 + \text{длина-буфера} \times \text{число-буферов}$, где первые восемь байт используются макрокомандой для построения блока управления буферным пулом.

7.4.2. Макрокоманда BUILDRCDB

Макрокоманда BUILDRCDB строит буферный пул и область логической записи в основной памяти, резервируемой программистом. Эта макрокоманда позволяет при обработке расширенных записей методом доступа QSAM иметь дело с логическими записями, а не сегментами. Существуют три формы записи макрокоманды BUILDRCDB: стандартная, описательная и исполнительная. Формат макрокоманды в стандартной форме:

[метка] BUILDRCDB адрес, число, длина-буфера, адрес, [длина-области-записи]

где адрес — адрес области основной памяти, используемой в качестве буферного пула, выровненный по границе слова;

число — требуемое число буферов (до 255);

длина-буфера — длина каждого буфера в байтах (до 32 760). Значение параметра должно быть кратно 4;

адрес — адрес области логической записи, выровненный по границе двойного слова;

длина-области-записи — размер области записи. Значение параметра должно быть не меньше максимального значения LRECL плюс 32. Если этот параметр опущен, то предполагается, что программист перед использованием области логической записи самостоятельно заполнил 32-байтовую область управляющей информацией.

Размер области основной памяти, требуемой для буферного пула, вычисляется по формуле: $12 + \text{длина-буфера} \times \text{число-буферов}$, где первые 12 байт используются макрокомандой для построения блока управления буферным пулом.

В формат макрокоманды BUILDRCД в описательной форме добавляется операнд $MF=L$. В качестве значений операндов, определяющих адреса областей основной памяти, может быть задан любой адрес, допустимый в адресной константе типа А. Значениями операндов «длина» и «число» может быть любое абсолютное выражение, допустимое в языке Ассемблер.

В формат макрокоманды BUILDRCД в исполнительной форме добавляется операнд $MF=(E, \text{адрес-списка-параметров})$.

В качестве значений операндов, определяющих адреса областей основной памяти, в этом случае может быть задан любой адрес, допустимый в командах типа RX, или любой из регистров 2—12, предварительно загруженный нужным адресом. Для указания адреса списка параметров можно использовать также регистр 1. Требования к значениям операндов «длина» и «число» те же, что и для описательной формы макрокоманды.

7.4.3. Макрокоманда GETPOOL

Макрокоманда GETPOOL строит буферный пул в области основной памяти, выделяемой управляющей программой. Макрокоманду можно использовать с любым методом доступа. Формат макрокоманды:

GETPOOL адрес-dcb, { число-буферов, длина-буфера }
(0)

где адрес-dcb — адрес блока DCB обрабатываемого набора данных;

число-буферов — требуемое число буферов (до 255);

длина-буфера — длина каждого буфера в байтах (до 32 760). Значение операнда должно быть кратно 8 и для QSAM — не меньше размера блока, определенного в макрокоманде DCB;

(0) — общий регистр 0. При указании (0) перед выдачей макрокоманды следует в разряды 0—15 регистра загрузить число буферов, а в разряды 16—31 — длину каждого буфера.

Макрокоманду GETPOOL следует выдавать в любой момент до завершения выполнения программы модификации блока DCB во время выполнения макрокоманды OPEN.

7.4.4. Макрокоманда GETBUF

Макрокоманда GETBUF позволяет запросить свободный буфер из буферного пула. Формат макрокоманды:

[метка] GETBUF адрес-dcb, регистр

адрес-dcb — адрес блока DCB, указывающего на буферный пул, из которого запрашивается буфер;

регистр — общий регистр (2—12), в котором будет находиться адрес буфера при возврате управления в программу или 0, если доступного буфера не оказалось.

Буфер, полученный с помощью макрокоманды GETBUF, должен быть возвращен в буферный пул с помощью макрокоманды FREEBUF.

7.4.5. Макрокоманда FREEBUF

Макрокоманда FREEBUF возвращает в буферный пул буфер, полученный с помощью макрокоманды GETBUF. Формат макрокоманды:

[метка] FREEBUF адрес-dcb, регистр

адрес-dcb — адрес блока DCB, указывающего на буферный пул, в который следует возвратить буфер;

регистр — общий регистр (2—12), содержащий адрес буфера, возвращаемого в буферный пул.

Макрокоманды GETBUF и FREEBUF можно использовать при обработке наборов данных методами доступа BDAM, BISAM, BRAM и BSAM.

7.4.6. Макрокоманда FREEPOOL

Макрокоманда FREEPOOL освобождает буферный пул, ранее выделенный автоматически или с помощью макрокоманды GETPOOL. Формат макрокоманды:

[метка] FREEPOOL адрес-dcb

где адрес-dcb — адрес блока DCB обрабатываемого набора данных.

7.4.7. Макрокоманда FREEDBUF

Макрокоманда FREEDBUF возвращает в буферный пул буфер, полученный с помощью динамической буферизации. Формат макрокоманды:

[метка] FREEDBUF адрес-decb, $\left\{ \begin{matrix} K \\ D \end{matrix} \right\}$, адрес-dcb

где адрес-decb — адрес блока DECB, построенного макрокомандой READ, получившей буфер;

K — метод доступа BISAM;

D — метод доступа BDAM;
адрес-dcb — адрес блока DCB обрабатываемого набора данных.

7.4.8. Макрокоманда RELSE

Макрокоманда RELSE заставляет следующую за ней макрокоманду GET проигнорировать записи, оставшиеся в текущем буфере, и выбрать первую логическую запись из другого буфера. Эту макрокоманду можно использовать с методами доступа QISAM и QSAM. Формат макрокоманды:

[метка] RELSE адрес-dcb

где адрес-dcb — адрес блока DCB обрабатываемого входного набора данных.

7.4.9. Макрокоманда TRUNC

Макрокоманда TRUNC при обработке сблокированных записей фиксированной или переменной длины осуществляет запись в выходной набор данных содержимого текущего буфера до его полного заполнения и заставляет следующую за ней макрокоманду PUT или PUTX перейти к заполнению другого буфера. Эту макрокоманду можно использовать с методом доступа QSAM. Формат макрокоманды:

[метка] TRUNC адрес-dcb

где адрес-dcb — адрес блока DCB обрабатываемого выходного набора данных.

7.5. Макрокоманды управления периферийными устройствами

7.5.1. Макрокоманда BSP

Макрокоманда BSP позволяет вернуться на один блок при обработке данных на магнитной ленте или устройстве прямого доступа. Эту макрокоманду можно использовать с методом доступа BSAM. Формат макрокоманды:

[метка] BSP адрес-dcb

адрес-dcb — адрес блока DCB обрабатываемого набора данных.

Возврат на блок для магнитной ленты всегда производится в направлении к точке загрузки.

Если набор данных на устройстве прямого доступа образован с помощью переходящих записей, то макрокоманду BSP выдавать нельзя.

При возврате управления макрокоманда BSP устанавливает в регистре 15 следующие коды возврата:

00 — нормальное выполнение;

04 — возврата на блок не произошло (возможно, была обнаружена ленточная марка или начало первого экстенда);

08 — возврата на блок не произошло; макрокоманда BSP была выдана для набора данных SYSIN или SYSOUT.

7.5.2. Макрокоманда CNTRL

Макрокоманда CNTRL выполняет операции управления накопителями на магнитной ленте при обработке данных методом доступа BSAM, перфокарточными устройствами ввода и печатающими устройствами при обработке данных методами доступа BSAM, QSAM. Формат макрокоманды:

[метка] CNTRL адрес-dcb,cc

где адрес-dcb — адрес блока DCB обрабатываемого набора данных;

cc — одна из следующих функций управления:

SS,п — приемный карман на перфокарточном устройстве ввода; п принимает значения 1, 2;

SP,п — количество строк, пропускаемых при печати; п принимает значения 1, 2, 3;

SK,п — прогон листа до указанного кода на перфоленте управления печатающим устройством; п принимает значения 1—12;

BSM — движение магнитной ленты в обратном направлении за ленточную марку и шаг вперед через нее. При этом поле DCBBLKCT DCB устанавливается в нуль;

FSM — движение магнитной ленты вперед через ленточную марку и затем назад за нее. При этом поле DCBBLKCT DCB устанавливается в нуль;

BSR [,п] — движение магнитной ленты в обратном направлении на заданное число блоков. Если параметр п опущен, то его значение принимается равным 1;

FSR [,п] — движение магнитной ленты вперед на заданное число блоков; п — число блоков от 1 до 32 767. Если параметр п опущен, то его значение принимается равным 1.

Макрокоманда CNTRL прекращает выполнение двух последних функций, если обнаруживает ленточную марку. В этом случае при возврате управления в регистр 15 помещается число блоков, на которое не удалось продвинуться. При нормальном завершении выполнения этих функций в регистр 15 помещается нуль.

При обработке данных методом доступа BSAM перед выдачей макрокоманды CNTRL следует проверить на завершение все ранее инициированные операции ввода-вывода.

7.5.3. Макрокоманда NOTE

Макрокоманда NOTE определяет и заносит в общий регистр 1 относительный адрес последнего прочитанного или записанного блока в наборе данных, находящемся на устройстве прямого доступа или магнитной ленте. Для устройства прямого доступа относительный адрес определяется в формате TTRZ, где TT — двубайтовый номер дорожки относительно начала набора данных, R — однобайтовый номер блока на этой дорожке и Z — байт, установленный в 0. Для магнитной ленты в регистр 1 помещается в двоичном коде номер блока относительно начала набора данных. Формат макрокоманды:

[метка] NOTE адрес-dcb

где адрес-dcb — адрес блока DCB обрабатываемого набора данных.

Макрокоманду NOTE можно использовать после завершения всех операций ввода-вывода для заданного в макрокоманде блока DCB при обработке данных методами доступа BPAM и BSAM. Однако при выполнении перечисленных условий эту макрокоманду не следует использовать, если обрабатываемый набор данных находится на томе магнитной ленты без стандартных меток и в соответствующей макрокоманде OPEN задан параметр RDBACK или в операторе DD, определяющем обрабатываемый набор данных, задан операнд DISP с параметром MOD.

7.5.4. Макрокоманда POINT

Макрокоманда POINT устанавливает том магнитной ленты или прямого доступа в положение, позволяющее выполнить операцию чтения или записи заданного в макрокоманде блока. Макрокоманду POINT можно использовать при обработке данных методами доступа BPAM и BSAM после завершения всех операций ввода-вывода для заданного в макрокоманде блока DCB. Формат макрокоманды:

[метка] POINT адрес-dcb,адрес

где адрес-dcb — адрес блока DCB обрабатываемого набора данных;

адрес — адрес слова, содержащего относительный адрес блока, который необходимо считать или записать.

Для магнитной ленты при чтении в обратном направлении следует указывать номер блока на 1 больше требуемого. Номер блока задается в двоичном коде. Макрокоманду POINT не следует использовать, если обрабатываемый набор данных находится на томе магнитной ленты без стандартных меток и в соответствующей макрокоманде OPEN задан параметр RDBACK или в операторе DD, определяющем обрабатываемый набор данных, задан операнд DISP с параметром MOD.

Для устройства с прямым доступом относительный адрес блока задается в формате TTRZ, где TT — двубайтовый номер дорож-

ки относительно начала набора данных, R — однобайтовый номер блока на этой дорожке, Z — указатель выборки, равный 0, если обработку следует начать с указанного в макрокоманде блока, и равный 1, если со следующего.

7.5.5. Макрокоманда PRTOV

Макрокоманда PRTOV обнаруживает так называемое условие переполнения, возникающее после вывода на печатающее устройство строки, следующей за строкой, которой соответствует код дорожки 9 или 12 на перфоленте управления печатающим устройством. Эту макрокоманду можно использовать при обработке данных методами доступа BSAM и QSAM. Формат макрокоманды:

[метка] PRTOV адрес-dcb,код[,адрес]

где адрес-dcb — адрес блока DCB обрабатываемого набора данных;

код — значение проверяемого кода дорожки (9 или 12);

адрес — адрес программы обработки условия переполнения.

При получении управления программой обработки условия переполнения регистры 0 и 1 содержат данные работы макрокоманды PRTOV, содержимое регистров 2—13 то же, что и до выдачи макрокоманды, в регистре 14 — адрес возврата, а в регистре 15 — адрес точки входа в программу обработки условия переполнения.

Если программа обработки условия переполнения не указана, то перед выполнением следующей макрокоманды PUT или WRITE происходит продвижение бумаги до кодовой пробивки на перфоленте, соответствующей дорожке 1.

7.5.6. Макрокоманда SETPRT

Макрокоманда SETPRТ позволяет загрузить буфер печатающего устройства со сменным набором символов новым образом набора символов, а печатающее устройство с управлением форматом страниц — новым образом формата страницы. Эту команду можно использовать при обработке наборов данных методами доступа BSAM и QSAM.

Существуют три формы записи макрокоманды SETPRT: стандартная, описательная и исполнительная. Формат макрокоманды в стандартной форме:

[метка] SETPRT адрес-dcb { UCS=x[,FCB=y][,OPTCD=z]
{ FCB=y[,OPTCD=z]
{ OPTCD=z

где адрес-dcb — адрес блока DCB распечатываемого набора данных;

UCS=x — образ набора символов, загружаемого в буфер печатающего устройства. Параметр x представляет собой выражение (идент [x₁]), где идент — идентификатор образа набора символов, а подпараметр x₁ принимает одно из следующих значений:

- F [OLD] — загрузка образа набора символов в режиме свертки;
- F [OLD], V [ERIFY] — загрузка образа набора символов в режиме свертки с последующей распечаткой;
- V [ERIFY] — распечатка образа набора символов;

FCB=y — образ формата страниц, загружаемого в буфер печатающего устройства. Параметр y представляет собой выражение (идент [y₁]), где идент — идентификатор образа формата страниц, а подпараметр y₁ принимает одно из следующих значений:

- V [ERIFY] — распечатка образа формата страницы;
- A [LIGN] — выравнивание форматов с помощью макрокоманды WTOR,

OPTCD=z — новое значение поля DCBOPTCD в блоке DCB; параметр z представляет собой выражение (z₁[z₂]), где подпараметр z₁ принимает одно из следующих значений:

- B — подавление ошибки в данных при распечатке;
- U — запрещает подавление ошибки в данных при распечатке;

а подпараметр z₂:

- F [OLD] — распечатка данных в режиме свертки;
- U [NFOLD] — распечатка данных без свертки.

В формат макрокоманды SETPRT в описательной форме добавляется операнд MF=L. В качестве значения операнда адрес-dcb может быть указан любой адрес, допустимый в адресной константе типа A.

В формат макрокоманды SETPRT в исполнительной форме добавляется операнд MF=(E, адрес-списка-параметров). В качестве значения операнда адрес-dcb может быть указан любой адрес, допустимый в командах типа RX, или любой из регистров 2—12, предварительно загруженный нужным адресом. Для указания адреса списка параметров можно использовать также регистр 1.

При возврате управления макрокоманда SETPRT в битах 16—23 регистра 15 устанавливает признаки, определяющие результат загрузки буфера формата страниц, а в битах 24—31 — признаки, определяющие результат загрузки буфера набора символов и характеризующие общие ситуации, возникающие при выполнении макрокоманды:

00; 00 — нормальное выполнение;

04; 04 — загрузка сменного набора символов или формата страниц отменена оператором ОС ЕС ЭВМ;

08; 08 — постоянная ошибка ввода-вывода при поиске раздела в библиотеке SYS1.IMAGELIB;

0C; 0C — обе попытки загрузить буфер печатающего устройства закончились неудачно из-за постоянной ошибки ввода-вывода;

10; 10 — распечатка образа набора символов и формата страниц не выполнена из-за постоянной ошибки ввода-вывода;

14; 14 — загрузка сменного набора символов или формата страниц отменена оператором ЭВМ после анализа распечатки;

00; 18 — макрокоманда не выполнена. Возможно не открыт блок DCB; неправильные параметры заданы в макрокоманде DCB; неправильные параметры в SETPRT; недопустимое печатающее устройство;

00; 1С — произошла постоянная ошибка ввода-вывода при выполнении предыдущей операции записи либо при выполнении печатающим устройством команд канала ЗАПРЕТИТЬ ИНДИКАЦИЮ ОШИБКИ В ДАННЫХ или РАЗРЕШИТЬ ИНДИКАЦИЮ ОШИБКИ В ДАННЫХ;

00; 20 — для выполнения макрокоманды SETPRT недостаточно основной памяти.

00; 24 — ошибка при открытии набора данных SYS1.IMAGELIB.

7.6. Макрокоманды управления операциями ввода-вывода BDAM

7.6.1. Макрокоманда READ (BDAM)

Макрокоманда READ инициирует выполнение операции считывания блока из набора данных в заданную область основной памяти. Завершение операции считывания устанавливается макрокомандами CHECK или WAIT. Существуют три формы записи макрокоманды READ: стандартная, описательная и исполнительная. Формат макрокоманды в стандартной форме:

[метка] READ имя-decb,тип,адрес-dcb,адрес-области,длина,
адрес-ключа,адрес-блока,адрес-поля

где имя-decb — имя блока DECB, создаваемого макрокомандой READ;

тип — тип операции чтения; принимает одно из следующих значений:

- DI — поиск записи по идентификатору блока; считываются ключи и данные;
- DK — поиск записи по ключу; считываются только данные;
- DIF — обратная связь при поиске записи по идентификатору;
- DIX — монопольное управление и обратная связь при поиске записи по идентификатору;
- DKF — обратная связь при поиске записи по ключу;
- DKX — монопольное управление блоком и обратная связь при поиске записи по ключу;
- R — добавляется к любому значению операнда «тип», кроме RU, и определяет занесение адреса следующей логической записи в поле, определенное значением операнда «адрес-поля»;
- RU — добавляется к любому значению операнда «тип», кроме R, и указывает занесение адреса следующей логи-

ческой записи или адреса записи R0 в поле, заданное значением операнда адрес-поля. Выбор определяется тем, какая из записей будет встречена первой;

адрес-dcb — адрес блока DCB обрабатываемого набора данных;

адрес-области — адрес области основной памяти для размещения блока данных; 'S' в качестве значения этого операнда определяет динамическое получение буфера, адрес которого помещается в блок DECB;

длина — размер считываемого блока данных в байтах (до 32 760); 'S' в качестве значения этого операнда указывает, что информацию о размере считываемого блока следует брать из блока DCB;

адрес-ключа — при поиске записи по ключу в область, определяемую значением этого операнда, следует поместить ключ, иначе в эту область помещается ключ считываемой записи. Если необходимо считать ключ, а значение операнда «адрес-области» — 'S', то и в качестве значения операнда «адрес-ключа» следует указать 'S'. В этом случае ключ и данные будут считаны в динамически выделенный буфер последовательно. Если ключ записи считывать нет необходимости, то в качестве значения операнда «адрес-ключа» следует указать 0;

адрес-блока — адрес считываемого блока в соответствии с типом адресации, определенным в поле DCBOPTCD DCB. Если определена обратная связь, то в эту же область в той же форме операционной системой помещается адрес считанного блока;

адрес-поля — адрес поля, в которое операционная система помещает адрес записи, следующей за считанной, если к значению операнда «тип» добавлены R или RU. При определении операнда адрес-поля в качестве значения операнда «длина» следует указать 'S'.

Для последовательного считывания неблокированных расширенных записей переменной длины из набора данных с прямой организацией используется макрокоманда READ в следующем формате:

[метка] READ имя-dcb,тип,адрес-dcb,адрес-области

где имя-dcb — имя блока DECB, создаваемого макрокомандой READ;

тип — тип операции чтения, принимает значение SF;

адрес-dcb — адрес блока DCB обрабатываемого набора данных;

адрес-области — адрес области для размещения сегмента логической записи. Все сегменты, кроме первого, размещаются в области со смещением на длину ключа.

Формат макрокоманды READ в описательной форме идентифицирует операнд MF=L, следующий за последним операндом макрокоманды в стандартной форме. В качестве значений операндов, определяющих адреса областей основной памяти, может быть задан любой адрес, допустимый в адресной константе типа A. Значение

нием операнда длина может быть любое абсолютное выражение, допустимое в языке Ассемблер.

Формат макрокоманды READ в исполнительный форме идентифицирует операнд MF=E, следующий за последним операндом макрокоманды в стандартной форме. Первый операнд макрокоманды READ в исполнительный форме — адрес-dcb, значением которого является адрес построенного ранее блока DECB. Этот операнд заменяет первый операнд макрокоманды READ в стандартной и описательной формах. В качестве значений операндов макрокоманды READ в исполнительный форме, определяющих адреса областей основной памяти, может быть задан любой адрес, допустимый в командах типа RX или любой из регистров 2—12, предварительно загруженный нужным адресом. Значением операнда «длина» может быть любое абсолютное выражение, допустимое в языке Ассемблер, или предварительно загруженный любой из регистров 2—12.

7.6.2. Макрокоманда RELEX

Макрокоманда RELEX отменяет монопольное управление блоком, запрошенным макрокомандой READ со значением параметра «тип» — DIX или DKX. Формат макрокоманды:

[метка] RELEX D,адрес-dcb,адрес-блока

где D — определяет прямой доступ;

адрес-dcb — адрес блока DCB в ранее выданной макрокоманде READ;

адрес-блока — значение этого операнда в ранее выданной макрокоманде READ.

При возврате управления макрокоманда RELEX устанавливает в регистре 15 следующие значения кода возврата:

00 — нормальное выполнение;

04 — блок не был под монопольным управлением;

08 — блок не входит в границы набора данных.

7.6.3. Макрокоманда WRITE (BDAM)

Макрокоманда WRITE инициирует выполнение операции записи блока из заданной области основной памяти в набор данных. Завершение операции записи устанавливается макрокомандой CHECK или WAIT. Существуют три формы записи макрокоманды WRITE: стандартная, описательная и исполнительная.

Для добавления блока к создаваемому набору данных с прямой организацией макрокоманда WRITE в стандартной форме имеет следующий формат:

[метка] WRITE имя-dcb,тип,адрес-dcb,адрес-области,
[длина][адрес-поля]

где имя-dcb — имя блока DECB, создаваемого макрокомандой WRITE;

тип — тип операции записи, принимает одно из следующих значений:

- SF — определяет добавление к набору данных нового блока;
- SD — определяет добавление к набору данных, состоящему из блоков фиксированной длины с ключами, фиктивного блока;
- SZ — определяет занесение на дорожку записи емкости дорожки (R0) для набора данных, состоящего из блоков переменной или неопределенной длины;
- SFR — определяет обратную связь при добавлении к набору данных нового блока; указывается только в том случае, если в макрокоманде DCB заданы операнды BFTEK=R и RECFM=VS;

адрес-dcb — адрес блока DCB создаваемого набора данных;

адрес-области — адрес области основной памяти, в которой находится блок для записи в набор данных. При записи данных с ключами ключ должен предшествовать данным. Если в качестве значения операнда «тип» указано SZ, то адрес области может быть опущен. Если в качестве значения операнда «тип» указано SD, то размер этой области задается на единицу больше размера ключа;

длина — размер блока (не более 32 760), добавляемого в набор данных с записями неопределенной длины. Для того чтобы длина добавляемого блока определялась из DCB, вместо размера блока следует указать 'S'. Для других форматов данных этот операнд не используется и может быть опущен;

адрес-поля — адрес поля, в которое операционная система помещает относительный адрес следующей записи, если в качестве значения операнда «тип» указано SFR.

При возврате управления макрокоманда WRITE устанавливает в регистре 15 следующие значения кода возврата:

00 — в набор данных с записями фиксированной, переменной или неопределенной длины будет добавлена запись, если в качестве значения операнда «тип» задано SF, SFR или SD и в наборе данных достаточно места для размещения записи. Если в качестве значения операнда «тип» задано SZ, то будет сделана запись емкости дорожки. В этом случае код 00 указывает также на то, что в наборе данных имеется по крайней мере еще одна дорожка;

04 — в набор данных с записями фиксированной длины будет добавлена запись данных и запись емкости дорожки. Для набора данных с записями переменной или неопределенной длины этот код возврата означает, что блок не будет записан. В этом случае следует сделать запись емкости дорожки, а затем повторно выдать макрокоманду WRITE для записи блока;

08 — в набор данных с записями фиксированной длины будет добавлена запись данных, а затем сделана запись емкости дорожки. В набор данных с записями переменной или неопределенной длины, если в качестве значения операнда тип задано SZ, будет

сделана запись емкости дорожки. В обоих случаях следующую запись можно будет добавить, если задано вторичное распределение для набора данных;

ОС — запись не будет добавлена в набор данных. В этом случае следует выдать макрокоманду CHECK и затем повторно выдать макрокоманду WRITE.

При создании набора данных с расширенными записями переменной длины (RECFM=VS и BFTEK=R в макрокоманде DCB) запись емкости дорожки делается автоматически, если для записи данных на текущей дорожке не хватает места, запись данных является последней на данной дорожке или запись не помещается на текущем томе. В последнем случае на всех оставшихся дорожках первого тома будут сделаны записи емкости дорожек, а запись данных размещена на новом томе.

Для добавления или обновления блока в существующем наборе данных с прямой организацией макрокоманда WRITE в стандартной форме имеет следующий формат:

[метка] WRITE имя-decb,тип,адрес-dcb,адрес-области,длина,
адрес-ключа,адрес-блока

где имя-decb — имя блока DECB, создаваемого макрокомандой WRITE;

тип — тип операции записи; принимает одно из следующих значений:

DA — добавление нового блока на первое доступное место, начиная с адреса, заданного операндом, и адрес-блока;

DAF — то же, что и DA. При возврате управления в поле, адрес которого задан операндом адрес-блока, помещается адрес нового блока;

DI — то же, что и DI. При возврате управления в поле, адрес-блока»;

DIF — то же, что и DI. При возврате управления в поле, адрес которого задан операндом «адрес-блока», помещается адрес нового блока;

DIX — то же, что и DI. Кроме этого, отменяется монопольное управление блоком, заданное в предыдущей макрокоманде READ;

DK — поиск по ключу, заданному операндом «адрес-ключа», и добавление блока на первое доступное место, начиная с адреса, заданного операндом «адрес-блока»;

DKF — то же, что и DK. При возврате управления в поле, адрес которого задан операндом «адрес-блока», помещается адрес нового блока;

DKX — то же, что и DK. Кроме этого, отменяется монопольное управление блоком, заданное в предыдущей макрокоманде READ;

адрес-dcb — адрес блока DECB обрабатываемого набора данных;

адрес-области — адрес области основной памяти, в которой на-

ходится блок для записи в набор данных. Если в соответствующей макрокоманде READ в качестве значения операнда адрес-области было указано 'S', то и в макрокоманде WRITE можно указать 'S'. Однако в этом случае адрес динамически выделенного буфера следует перенести из блока DECB, построенного макрокомандой READ, в блок DECB, построенный макрокомандой WRITE. После выполнения макрокоманды WRITE область динамически выделенного буфера освобождается;

длина — размер блока, записываемого в набор данных (не более 32 760 байт). Для того, чтобы размер блока определялся из блока DCB, следует в качестве значения этого операнда указать 'S';

адрес-ключа — адрес области основной памяти, куда помещается ключ записываемого блока. Если набор данных состоит из записей без ключей, то в качестве значения этого операнда следует указать 0. Если задана динамическая буферизация, то в качестве значения этого операнда следует указать 'S';

адрес-блока — адрес области основной памяти, содержащей адрес блока на устройстве прямого доступа в форме, соответствующей типу адресации, определенному в поле DCBOPTCD блока DCB.

Требования к формату макрокоманды WRITE в описательной и исполнительской форме аналогичны описанным выше для макрокоманды READ.

7.7. Макрокоманды управления операциями ввода-вывода BISAM

7.7.1. Макрокоманда READ (BISAM)

Макрокоманда READ инициирует выполнение операции считывания из набора данных блока, содержащего логическую запись с заданным ключом. По завершении операции указанный блок размещается в определенной области основной памяти, а адрес логической записи в блоке помещается в блок DECB. Завершение операции считывания устанавливается макрокомандами CHECK или WAIT. Формат макрокоманды в стандартной форме:

[метка] READ имя-decb,тип,адрес-dcb,адрес-области,длина,адрес-ключа

имя-decb — имя блока DECB, создаваемого макрокомандой READ;

тип — тип операции чтения, который принимает одно из следующих значений: K — прямая выборка, KU — прямая выборка с последующим исправлением;

адрес-dcb — адрес блока DCB обрабатываемого набора данных;

адрес-области — адрес области основной памяти для размещения блока данных; 'S' в качестве значения этого операнда опре-

деляет динамическое получение буфера, адрес которого помещается в блок DECB;

длина — размер считываемого блока в байтах (до 32 760); 'S' в качестве значения этого операнда указывает, что информация о размере считываемого блока должна браться из области счета блока на устройстве прямого доступа. Значение 'S' указывается при обработке заблокированных записей;

адрес-ключа — адрес области основной памяти, куда помещается ключ считываемой логической записи.

Требования к формату макрокоманды READ в описательной и исполнительной формах аналогичны описанным выше в 7.6.1.

7.7.2. Макрокоманда WRITE (BISAM)

Макрокоманда WRITE позволяет добавить или заменить логическую запись или блок в существующем индексно-последовательном наборе данных. Завершение операции записи устанавливается макрокомандой CHECK или WAIT. Формат макрокоманды в стандартной форме:

[метка] WRITE имя-decb, тип, адрес-dcb, адрес-области, длина,
адрес-ключа

где имя-decb — имя блока DECB, создаваемого макрокомандой WRITE;

тип — тип операции записи, принимает одно из следующих значений:

К — обновление логической записи или блока;

KN — включение новой записи или обновление записи переменной длины с изменением длины.

Чтобы обновить запись, считанную с помощью макрокоманды READ, в которой в качестве значения операнда «тип» было задано KU, следует выдать макрокоманду WRITE в исполнительной форме, указав адрес блока DECB, построенного макрокомандой READ;

адрес-dcb — адрес блока DCB обрабатываемого набора данных;

адрес-области — адрес области основной памяти, в которой находится блок для записи в набор данных. Если в соответствующей макрокоманде READ в качестве значения этого операнда указано 'S', то и в макрокоманде WRITE можно указать 'S'. Однако в этом случае адрес динамически выделенного буфера следует из блока DECB, построенного макрокомандой READ, перенести в блок DECB, построенный макрокомандой WRITE. После выполнения макрокоманды WRITE область динамически выделенного буфера освобождается;

длина — размер блока, записываемого в набор данных (не более 32 760 байт). При добавлении новых записей или замене записей в блоке вместо действительного размера блока следует указывать 'S'.

Требования к формату макрокоманды WRITE в описательной и исполнительной формах аналогичны описанным выше в 7.6.1.

7.8. Макрокоманды управления операциями ввода-вывода BPAM

Макрокоманды метода доступа BPAM (BLDL, FIND, STOW) в сочетании с макрокомандами метода доступа BSAM (POINT, READ, WRITE и CHECK) вносят изменения в существующие разделы набора данных, удаляют или добавляют новые. Если нужно обработать только справочник или отдельный раздел библиотечного набора данных, то можно обойтись только макрокомандами методов доступа QSAM или BSAM, указав в блоке DCB, DSORG=PS. Для справочника в блоке DCB или операторе DD необходимо указать следующие характеристики формата записей и размера записей и блоков: LRECL=256, BLKSIZE=256, RECFM=F; для обработки отдельного раздела следует в операторе DD после имени набора данных указать в скобках имя раздела.

7.8.1. Макрокоманда BLDL

Макрокоманда BLDL позволяет занести в заранее построенный программистом список информацию о заданных в списке разделах библиотечного набора данных. Списку должно предшествовать поле длиной 4 байта с информацией в двоичном коде. В первые два байта этого поля следует поместить число элементов в списке (до 255), а во вторые два байта — длину каждого элемента списка (14—76 байт). Элементы списка должны следовать в алфавитно-цифровом порядке имен разделов. Каждый элемент списка должен начинаться с имени раздела или дополнительного имени, которые дополняются справа, если необходимо, пробелами до 8 байт. Макрокоманда BLDL помещает в этот список после имени раздела его относительный адрес (3 байта), номер набора данных, если обрабатываются сцепленные наборы данных (1 байт), два байта признаков и информацию из поля данных пользователя. Первый байт признаков определяет тип библиотечного набора данных: 0 — личная библиотека; 1 — общая библиотека; 2 — библиотека задания или пункта задания. Нулевой разряд второго байта признаков устанавливается в 1, если в список попадает информация для дополнительного имени. Разряды 1 и 2 этого байта указывают количество 4-байтовых полей в поле данных пользователя, содержащих относительные адреса элементов раздела, а разряды с 3 по 7 — количество полуслов, составляющих поле данных пользователя. Формат макрокоманды:

[метка] BLDL адрес-dcb,адрес-области

где адрес-dcb — адрес блока DCB личной библиотеки или 0, если выбирается информация из общей библиотеки, библиотеки задания или пункта задания;

адрес-области — адрес списка в основной памяти, выровненного по границе полуслова.

При возврате управления макрокоманда BLDL устанавливает в регистре 15 один из следующих кодов возврата:

00 — нормальное выполнение;

04 — список не может быть заполнен, возможно, из-за неправильно заданных начальных параметров. Для каждого элемента, который не был найден, номер записи на дорожке в поле относительного адреса устанавливается в нуль;

08 — постоянная ошибка ввода-вывода при обращении к справочнику библиотечного набора данных.

7.8.2. Макрокоманда FIND

Макрокоманда FIND определяет адрес заданного раздела и помещает его в блок DCB обрабатываемого библиотечного набора данных, что позволяет начать выборку записей этого раздела следующей макрокомандой READ. Формат макрокоманды:

[метка] FIND адрес-dcb,адрес-области,тип

где адрес-dcb — адрес блока DCB обрабатываемого набора данных;

адрес-области — адрес имени раздела, если значение операнда «тип» — С, и относительный адрес раздела в области списка, построенного, например, с помощью макрокоманды BLDL, если значение операнда «тип» — D.

При возврате управления макрокоманда FIND устанавливает в регистре 15 один из следующих кодов возврата:

00 — нормальное выполнение;

04 — имя раздела в справочнике не найдено;

08 — постоянная ошибка ввода-вывода при обращении к справочнику библиотечного набора данных.

Перечисленные коды возврата устанавливаются, если заданы операнды «адрес-области» и D. Если заданы операнды «адрес-области» и С, то код возврата всегда устанавливается в 0. Ошибка в этом случае обнаруживается только при выдаче макрокоманды READ.

7.8.3. Макрокоманда STOW

Макрокоманда STOW позволяет добавить, изменить или исключить элемент справочника библиотечного набора данных. Формат макрокоманды:

[метка] STOW адрес-dcb,адрес-области[тип]

где адрес-dcb — адрес блока DCB обрабатываемого набора данных;

адрес-области — адрес области основной памяти, в которой содержится информация, необходимая для внесения изменений в справочник. Формат области определяется значением операнда «тип»;

тип — тип операции со справочником; принимает одно из следующих значений:

A — добавление элемента в справочник;

C — изменение имени раздела в справочнике;

D — исключение элемента из справочника;

R — замена элемента в справочнике.

В первом и последнем случаях (A или R) формат области, задаваемой операндом «адрес-области», совпадает с форматом элемента справочника. В первые 8 байт области помещается имя раздела или дополнительное имя, дополненное справа, если необходимо до 8 байт, относительный адрес раздела (3 байта), байт признаков и поле данных пользователя (до 62 байт). Поле относительного адреса раздела заполняется программистом только в том случае, если нулевой разряд байта признаков устанавливается в 1. Нулевой разряд байта признаков устанавливается в 1, если определяется элемент справочника для дополнительного имени. Разряды 1 и 2 этого байта указывают количество 4-байтовых полей в поле данных пользователя, содержащих относительные адреса элементов раздела, а разряды 3—7 — количество полуслов, составляющих поле данных пользователя. Если в качестве значения операнда «тип» задано A или R, то макрокоманда STOW, кроме внесения изменений в справочник, завершает обработку текущего раздела формированием записи конца раздела.

Для замены имени раздела в элементе справочника в первые 8 байт области, задаваемой значением операнда «адрес-области», помещается старое имя раздела или старое дополнительное имя, а во вторые 8 байт — новое имя раздела или новое дополнительное имя. Оба имени дополняются справа, если необходимо, пробелами до 8 байт.

Для исключения элемента из справочника в область, задаваемую значением операнда «адрес-области», помещается имя раздела или дополнительное имя, дополненное справа, если необходимо, пробелами до 8 байт.

При возврате управления макрокоманда STOW устанавливает в регистре 15 один из следующих кодов возврата:

00 — нормальное выполнение (тип операции — A, C, D, R);

04 — заданное имя содержится в справочнике (тип операции — A, C);

08 — заданное имя не найдено в справочнике (тип операции D, C, R);

0C — нет места в справочнике для добавления или изменения элемента (тип операции — A, R);

10 — постоянная ошибка ввода-вывода при обращении к справочнику;

14 — блок DCB для обрабатываемого набора данных не открыт или открыт некорректно;

18 — ненормальное завершение макрокоманды GETMAIN, выданной при выполнении макрокоманды STOW.

7.9. Макрокоманды управления операциями ввода-вывода BSAM

7.9.1. Макрокоманда READ (BSAM)

Макрокоманда READ инициирует выполнение операции считывания блока из последовательного или библиотечного набора данных в заданную область основной памяти. Завершение операции считывания устанавливается макрокомандой CHECK. Формат макрокоманды в стандартной форме:

[метка] READ имя-decb,тип,адрес-dcb,адрес-области,длина

где имя-decb — имя блока DECB, создаваемого макрокомандой READ;

тип — тип операции чтения, принимает одно из следующих значений:

SF — выборка блока в прямом направлении;

SB — выборка блока в обратном направлении.

Последний тип операции чтения применяется только для наборов данных на магнитной ленте, состоящих из записей фиксированной или неопределенной длины;

адрес-dcb — адрес блока DCB обрабатываемого набора данных;

адрес-области — адрес области основной памяти для размещения блока данных.

Если значение операнда «тип» — SB, то в качестве значения операнда «адрес-области» следует задать адрес последнего байта области для размещения блока;

длина — размер считываемого блока данных в байтах (до 32760 байт). Информация о размере считываемого блока будет браться из поля BLKSIZE блока DCB, если в качестве значения этого операнда указать 'S'. Операнд «длина» игнорируется при обработке записей фиксированной и переменной длины и может быть опущен.

Требования к формату макрокоманды READ в описательной и исполнительной формах аналогичны описанным в 7.6.1.

7.9.2. Макрокоманда WRITE (BSAM)

Макрокоманда WRITE инициирует выполнение операции записи блока из заданной области основной памяти в последовательный или библиотечный набор данных. Завершение операции и результат ее выполнения устанавливаются макрокомандой CHECK. Формат макрокоманды в стандартной форме:

[метка] WRITE имя-decb,SF,адрес-dcb,адрес-области[длина]

где имя-decb — имя блока DECB, создаваемого макрокомандой WRITE;

адрес-dcb — адрес блока DCB обрабатываемого набора данных;

адрес-области — адрес области основной памяти, в которой находится блок для записи в набор данных. При записи данных с ключами ключ должен предшествовать данным;

длина — размер блока, записываемого в набор данных (до 32 760 байт). Информация о размере блока будет браться из поля BLKSIZE блока DCB, если в качестве значения этого операнда указать 'S'. Операнд «длина» игнорируется и может быть опущен при обработке записей фиксированной и переменной длины.

Требования к формату макрокоманды WRITE в описательной и исполнительной формах аналогичны описанным выше в 7.6.1.

7.10. Макрокоманды управления операциями ввода-вывода QISAM

7.10.1. Макрокоманда ESETL

Макрокоманда ESETL позволяет прекратить последовательную выборку записей из индексно-последовательного набора данных и освободить буферы, связанные с блоком DCB. Формат макрокоманды:

[метка] ESETL адрес-dcb

где адрес-dcb — адрес блока DCB обрабатываемого набора данных.

7.10.2. Макрокоманда GET (QISAM)

Макрокоманда GET осуществляет последовательную выборку логических записей из индексно-последовательного набора данных и размещение их в заданной области основной памяти. Формат макрокоманды:

[метка] GET адрес-dcb[,адрес-области]

где адрес-dcb — адрес блока DCB обрабатываемого набора данных;

адрес-области — адрес области основной памяти для размещения логической записи. Этот операнд задается в том случае, если в макрокоманде DCB операнд MACRF=GM (режим пересылки). Если при этом условии операнд «адрес-области» опущен, то предполагается, что адрес области содержится в регистре 0. При возврате управления адрес области в регистре 0 сохраняется, а в регистр 1 помещается адрес блока DCB.

Если в макрокоманде DCB задан операнд MACRF=GL (режим указания), то операнд «адрес-области» задавать не следует. В этом случае при возврате управления в регистр 1 помещается адрес логической записи в буфере ввода, выделенном автоматически или с помощью одной из макрокоманд управления буферами.

7.10.3. Макрокоманда PUT (QISAM)

Макрокоманда PUT помещает запись из области основной памяти в индексно-последовательный набор данных. Формат макрокоманды:

[метка] PUT адрес-dcb[,адрес-области]

где адрес-dcb — адрес блока DCB обрабатываемого набора данных;

адрес-области — адрес области основной памяти, в которой находится логическая запись. Этот операнд задается, если в макрокоманде DCB указан операнд MACRF=PM (режим пересылки). В этом случае логическая запись пересылается в буфер вывода, после заполнения которого вместе с остальными записями блока переносится в набор данных.

Если в макрокоманде DCB указан операнд MACRF=PL (режим указания), то операнд «адрес-области» задавать не следует. В этом случае при возврате управления в регистр 1 помещается адрес позиции в буфере вывода для формирования там логической записи. Адрес следующей свободной области в буфере вывода вычисляется исходя из содержимого поля LRECL блока DCB.

7.10.4. Макрокоманда PUTX (QISAM)

Макрокоманда PUTX позволяет заменить логическую запись в индексно-последовательном наборе данных. Предварительно запись должна быть считана макрокомандой GET в режиме указания.

Формат макрокоманды:

[метка] PUTX адрес-dcb

где адрес-dcb — адрес блока DCB входного набора данных.

7.10.5. Макрокоманда SETL

Макрокоманда SETL позволяет начать обработку индексно-последовательного набора данных с любой заданной логической записи или с заданного адреса на устройстве прямого доступа. Непосредственно выборка записей осуществляется с помощью макрокоманды GET. Формат макрокоманды:

[метка] SETL адрес-dcb,тип,адрес-области

адрес-dcb — адрес блока DCB обрабатываемого набора данных;

тип — определяет запись, с которой следует начать обработку; принимает одно из следующих значений:

- К — запись с заданным ключом;
- КН — первая запись с заданным или большим значением ключа;
- КС — первая запись с заданным префиксом ключа.

Если эта запись помечена как удаленная, то первая — не помеченная;

KD — запись с заданным ключом с выборкой только полей данных (для несблокированных записей);

KDH — первая запись с заданным или большим значением ключа с выборкой только полей данных (только для несблокированных записей);

I — запись с заданным адресом на устройстве прямого доступа;

ID — запись с заданным адресом на устройстве прямого доступа с выборкой только полей данных (для несблокированных записей);

B — первая запись набора данных;

BD — первая запись набора данных с выборкой только полей данных (для несблокированных записей);

адрес-области — адрес области основной памяти, содержащей ключ, префикс ключа или физический адрес записи в форме MBVCCNHR. Этот операнд не следует задавать, если в качестве значения операнда «тип» задано B или BD.

7.11. Макрокоманды управления операциями ввода-вывода QSAM

7.11.1. Макрокоманда GET (QSAM)

Макрокоманда GET осуществляет последовательную выборку логических записей из набора данных с последовательной организацией и размещает их в заданной области основной памяти. Формат макрокоманды:

[метка] GET адрес-dcb[,адрес-области]

где адрес-dcb — адрес блока DCB обрабатываемого набора данных;

адрес-области — адрес области основной памяти для размещения логической записи. Этот операнд задается в случаях, когда в качестве значения параметра операнда MACRF макрокоманды DCB указано GM, GD или GT. Если при этом операнд «адрес-области» опущен, то предполагается, что адрес области основной памяти содержится в регистре 0.

При всех значениях параметров операнда MACRF, при возврате управления макрокоманда GET помещает в регистр 1 адрес логической записи или сегмента расширенной записи переменной длины, подлежащей обработке. Расширенные записи переменной длины, размер которых больше 32 756 байт, могут быть обработаны, только если MACRF=GL.

7.11.2. Макрокоманда PUT (QSAM)

Макрокоманда PUT помещает запись из области основной памяти в последовательный набор данных. Формат макрокоманды:

[метка] PUT адрес-dcb[,адрес-области]

адрес-dcb — адрес блока DCB обрабатываемого набора данных;

адрес-области — адрес области основной памяти, в которую помещается логическая запись для последующего вывода. Этот операнд задается в случаях, когда в качестве значения параметра операнда MACRF макрокоманды DCB указано PM, PD или PT. Если при этом операнд «адрес-области» опущен, то предполагается, что адрес области основной памяти содержится в регистре 0. Если MACRF=PD, то перед выдачей макрокоманды PUT в поле DCB LRECL блока DCB следует поместить полную длину данных.

При обработке расширенных записей переменной длины в случае, если значение операнда BFTEK макрокоманды DCB равно A или была выдана макрокоманда BUILDRCД, макрокоманда PUT при возврате управления помещает в регистр 1 адрес области в буфере вывода, в которую следует поместить сегмент записи, а в регистр 0 — размер остатка буфера.

7.11.3. Макрокоманда PUTX (QSAM)

Макрокоманда PUTX позволяет заменить логическую запись в существующем последовательном наборе данных или поместить логическую запись в создаваемый последовательный набор данных. В первом случае запись должна быть считана макрокомандой GET в режиме указания. Во втором случае возможно создание набора данных с записями формата VS или VBS. Входной набор данных может состоять из записей этого формата, если задан операнд BFTEK=A в макрокоманде DCB или была выдана макрокоманда BUILDRCД. Формат макрокоманды:

[метка] PUTX адрес-dcb1 [адрес-dcb2]

адрес-dcb1 — адрес блока DCB для входного набора данных;

адрес-dcb2 — адрес блока DCB для выходного набора данных. Этот операнд задается только тогда, когда макрокоманда PUTX используется для создания набора данных. Если при этом второй операнд опущен, то предполагается, что необходимый адрес содержится в регистре 0.

7.12. Макрокоманды обработки ошибок ввода-вывода

7.12.1. Макрокоманда SYNADAF

Макрокоманда SYNADAF выполняет анализ ошибок ввода-вывода и на основе результатов анализа формирует сообщение. Эта макрокоманда включается в программу обработки ошибок ввода-вывода, адрес которой указывается в операнде SYNAD макрокоманды DCB. Необходимые параметры для выполнения макрокоманды SYNADAF содержатся в регистрах 0 и 1 при входе в программу. Формат макрокоманды:

[метка] SYNADAF ACSMETH=метод[,PARM1=значение1]
[,PARM2=значение2]

ACSMETH=метод — метод доступа. В качестве значений параметра указываются BDAM, BPAM, BSAM, QSAM, BISAM, EXCP, QISAM;

PARM1=значение1 — информация, зависящая от метода доступа. Если операнд ACSMETH определяет BDAM, BPAM, BSAM, BISAM или QSAM, то PARM1 определяет регистр, содержащий информацию, находившуюся в регистре 1 при входе в программу обработки ошибок ввода-вывода. Для QISAM операнд PARM1 определяет адрес блока DCB. Для EXCP этот операнд определяет адрес блока IOB. Если операнд PARM1 опущен, то полагается PARM1=(1).

PARM2=значение2 — определяет дополнительную информацию, зависящую от метода доступа. Если операнд ACSMETH определяет BDAM, BISAM, BPAM, BSAM, QISAM или QSAM, то PARM2 определяет регистр, содержащий информацию, находившуюся в регистре 0 при входе в программу обработки ошибок ввода-вывода. Если ACSMETH=EXCP, то операнд PARM2 должен быть опущен. В остальных случаях, если операнд PARM2 опущен, предполагается PARM2=(0).

При возврате управления макрокоманда SYNADAF помещает в регистр 0, 1 соответственно код возврата и адрес буфера, содержащего сообщение.

Значения кода возврата:

00 — нормальное выполнение. В байтах 8—13 буфера сообщения находятся пробелы;

04 — нормальное выполнение. В байтах 8—13 буфера сообщения находятся двоичные данные;

08 — ненормальное выполнение. Байты 50—127 буфера сообщения заполнены звездочками. В байтах 8—13 находятся либо пробелы, либо двоичные данные.

Формат буфера сообщения:

Размер поля
в байтах

Содержимое

0—4	Счетчик длины блока в форме LLbb, где LL=128 и bb=000
5—7	Счетчик длины записи в форме IIbb, где II=124 и bb=000
8—12	Адрес буфера ввода и пробелы, если задана операция вывода
12—14	Количество считываемых байт для операции ввода и пробелы, если задана операция вывода
14—50	Пробелы
50—57	Имя задания
58	Запятая (,)
59—66	Имя пункта задания
67	Запятая (,)
68—71	Адрес периферийного устройства
72—73	Тип периферийного устройства; вывода; TA — для накопителя на магнитной ленте (UR для устройств ввода-вывода; DA — для устройств прямого доступа)
74	Запятая (,)
75—82	Имя оператора DD
83	Запятая (,)
84—89	Выполняемая операция ввода-вывода
90	Запятая (,)

91—105	Описание ошибки
106	Запятая (,)
107—128	Для устройства прямого доступа: 107—120 — физический адрес в форме BBCCNHR; 121 — запятая (,); 122—127 — метод доступа. Для накопителя на магнитной ленте: 107—113 — носительный номер блока в десятичном; 114 — запятая (,); 115—119 — метод доступа; 120—127 — пробелы. Остальные периферийные устройства: 107—120 — звездочки; 121 — запятая (,); 122—127 — метод доступа
128—131	Значение параметра операнда PARM2
132—135	Значение параметра операнда PARM1

7.12.2. Макрокоманда SYNADARLS

Макрокоманда SYNADARLS освобождает области основной памяти, запрошенные макрокомандой SYNADAF для сохранения регистров, буфера сообщения и области сохранения параметров. При выдаче макрокоманды SYNADARS регистр 13 должен содержать обеспечиваемый макрокомандой SYNADAF адрес области сохранения регистров. Формат макрокоманды:

[метка] SYNADARLS

При возврате управления макрокоманда SYNADARLS помещает в регистр 0 один из следующих кодов возврата:

00 — нормальное выполнение;

08 — ненормальное выполнение, возможно из-за того, что регистр 13 не содержит адрес области сохранения регистров, обеспечиваемой макрокомандой SYNADAF, или же из-за нарушения связи областей сохранения.

7.12.3. Содержимое регистров при входе в программу SYNAD

При получении управления программой SYNAD (BISAM) регистры 0—15 содержат следующую информацию:

0 — адрес первого байта уточненного состояния в блоке IOB;

1 — адрес блока DECB;

2 — 13 — то же, что до выдачи макрокоманды;

14 — адрес возврата;

15 — адрес программы SYNAD.

При получении управления программой SYNAD (QISAM) регистры 0—15 содержат следующую информацию:

0 — адрес буфера, содержащего последнюю обработанную запись;

1 — адрес блока DCB;

2—13 — то же, что до выдачи макрокоманды;

14 — адрес возврата;

15 — адрес программы SYNAD.

При получении управления программой SYNAD (BDAM, BRAM, BSAM, QSAM) регистры 0—15 содержат следующую информацию:

0 — в разрядах 0—7 адрес программы канала относительно

адреса IOB (только для QSAM). В разрядах 8—31 содержится адрес блока DECB для BDAM, BPAM и BSAM. Для QSAM в этих разрядах содержится адрес блока IOB;

1 — в разрядах 0—7 устанавливается идентификатор операции, при выполнении которой возникла ошибка ввода-вывода: разряд 0 устанавливается в 1, если ошибка произошла во время выполнения операции ввода. Разряд 1 устанавливается в 1, если ошибка произошла во время выполнения операции вывода. Разряд 2 устанавливается в 1, если ошибка произошла во время выполнения макрокоманд BSP, CNTRL или POINT (только для BPAM и BSAM). Разряд 3 устанавливается в 1, если ошибка произошла во время обновления существующей записи или в случае ошибки, не препятствующей выполнению операции чтения записи. Разряд 3 устанавливается в 0, если ошибка произошла во время образования новой записи или в случае ошибки, препятствующей выполнению операции чтения. Разряд 4 устанавливается в 1, если ошибка в запросе. В этом случае индикаторы состояния, адресуемые в блоке DECB, не устанавливаются (только для BDAM, BPAM, BSAM). Разряд 5 устанавливается в 1, если был найден неправильный знак при преобразовании данных на перфоленте (только для BSAM и QSAM). Разряд 6 устанавливается в единицу, если произошла аппаратурная ошибка (только для BDAM). Разряд 7 устанавливается в 1, если не найдено места для записи (только для BDAM). В разрядах 8—31 содержится адрес блока DCB;

2—13 — то же, что до выдачи макрокоманды;

14 — адрес возврата;

15 — адрес программы SYNAD.

Редактор связей

8.1. Функции Редактора связей

Редактор связей выполняет следующие функции:

- преобразует объектные модули, которые составляют результат работы трансляторов и Ассемблера, в загрузочные модули, пригодные для загрузки и выполнения в основной памяти; помещает загрузочный модуль в библиотечный набор данных в качестве поименованного раздела, откуда он может быть выбран в основную память для выполнения;

- объединяет в один загрузочный модуль поступающие на основной ввод объектные модули и поступающие на дополнительный ввод объектные и загрузочные модули;

- разрешает внешние ссылки между отдельными модулями;

- включает модули из автоматически вызываемой библиотеки для разрешения внешних ссылок, оставшихся неразрешенными после обработки основного и дополнительного ввода;

- обрабатывает управляющие операторы, поступающие на основной и дополнительный ввод;

- редактирует модули путем замены, исключения и перегруппировки программных секций;

- создает программы оверлейной структуры;

- создает в одном пункте задания несколько загрузочных модулей;

- присваивает загрузочному модулю заданные характеристики;

- резервирует память для общих программных секций (Фортран, Ассемблер) и статических внешних областей (ПЛ/1);

- вычисляет полную длину и присваивает смещение всем псевдореграмм;

- обеспечивает информацию об индексе состояния модуля в справочнике библиотеки загрузочных модулей;

- присваивает загрузочному модулю код функционирования (только в издании 6.1);

- позволяет выравнивать программные секции и общие области по границам в 2 или 4 Кбайта (только в издании 6.1).

8.2. Оператор EXEC для Редактора связей

Оператор EXEC для Редактора связей определяет имя вызываемого Редактора связей (параметр PGM), режимы работы Редактора связей и характеристики загрузочных модулей (параметр PARM), размер раздела основной памяти (параметр REGION), а также может содержать другие параметры, необходимые в каждом конкретном случае.

8.3. Имя Редактора связей

Редактор связей имеет следующие уровни: F44, F64, F88 и F128, которые указываются при генерации операционной системы. Число в обозначении уровня определяет минимальный объем основной памяти (в Кбайтах) для Редактора связей. С повышением уровня возрастает скорость работы Редактора связей. В системе одновременно может быть несколько различных уровней Редактора связей, которым присваиваются собственные имена. Кроме того, имеются дополнительные имена, которые соответствуют максимальному уровню Редактора связей, существующему в конкретном сгенерированном варианте операционной системы. Указанные ниже имена следует использовать в параметре PGM оператора EXEC для вызова Редактора связей соответствующего уровня.

Уровень F64 в издании 4.1 не поддерживается. Остальные уровни этого издания имеют следующие имена: F44—IEWLF440; F88—IEWLF880; F128—IEWLF128. Кроме того, допускаются дополнительные имена IEWL и LINKEDIT, соответствующие максимальному уровню Редактора связей.

Уровни Редактора связей в издании 6.1 имеют следующие имена: F44—IEWLF440, F64—HEWLF064, F88—HEWLF088, F128—HEWLF128.

Дополнительные имена IEWL, HEWL и LINKEDIT соответствуют максимальному уровню Редактора связей. При этом имя HEWL не распространяется на Редактор связей уровня F44.

8.4. Характеристики загрузочных модулей и режимы работы Редактора связей

Характеристики загрузочных модулей и режимы работы Редактора связей могут быть заданы путем указания параметров Редактора связей в поле PARM оператора EXEC, вызывающего Редактор связей, в следующем формате:

PARM='параметр-Редактора-связей,...'

где параметр-Редактора-связей может принимать значения, рассматриваемые ниже.

8.4.1. Характеристики загрузочных модулей

RENT — создаваемые загрузочные модули реентерабельные. Характеристика реентерабельности записывается Редактором связей в элемент оглавления библиотеки. При этом Редактор связей не проверяет фактическую реентерабельность загрузочного модуля.

Все программные секции и загрузочные модули, составляющие вновь создаваемый Редактором связей реентерабельный загрузочный модуль, также должны быть реентерабельными. Если на вход Редактора связей поступает не реентерабельный загрузочный модуль, заданная характеристика реентерабельности не принимается.

Реентерабельному модулю автоматически присваивается характеристика повторной используемости (REUS). Указание параметра RENT несовместимо с указанием параметров OVLY и REUS. Если не указаны параметр RENT, а также параметр REUS, то загрузочный модуль не допускает повторного использования и его копия должна каждый раз разгружаться в основную память до начала его выполнения.

REUS — создаваемые загрузочные модули повторно используемые. Характеристика повторной используемости записывается Редактором связей в элемент оглавления библиотеки. При этом Редактор связей не проверяет фактическую возможность повторного использования загрузочного модуля.

Все программные секции и загрузочные модули, составляющие вновь создаваемый Редактором связей повторный модуль, должны быть повторно используемыми или реентерабельными. Если на вход Редактора связей поступает не повторно используемый и не реентерабельный загрузочный модуль, то заданная характеристика не принимается.

Повторно используемый модуль не является реентерабельным. Указание параметра REUS несовместимо с указанием параметров OVLY, RENT и SCTR. Если не указаны параметр REUS, а также параметр RENT, то загрузочный модуль не допускает повторного использования. Копия такого модуля должна каждый раз загружаться в основную память до начала его выполнения.

Наряду с характеристикой реентерабельности при указании параметра RENT автоматически присваивается характеристика повторной используемости без указания параметра REUS.

REFR — создаваемые загрузочные модули обновляемые. Характеристика обновляемости записывается Редактором связей в элемент оглавления библиотеки. При этом Редактор связей не проверяет фактическую возможность обновляемости загрузочного модуля.

Копию обновляемого загрузочного модуля в основной памяти можно заменить на новую с помощью программных средств восстановления и диагностики, при восстановлении работоспособности программы после порчи копии из-за сбоев и ошибок в основ-

ной памяти. Копия такого модуля не должна модифицироваться в момент выполнения. Все программные секции и загрузочные модули, составляющие вновь создаваемый Редактором связей обновляемый загрузочный модуль, также должны быть обновляемыми. Если на вход Редактора связей поступает не обновляемый загрузочный модуль, то заданная характеристика обновляемости не принимается. Характеристика обновляемости не означает характеристики реентерабельности.

Характеристика обновляемости не определяется указанием параметра RENT. Указание параметра REFR несовместимо с указанием параметра OVLY. Отсутствие параметра REFR означает, что загрузочный модуль не является обновляемым.

OVLY — создаваемые загрузочные модули имеют оверлейную структуру, задаваемую управляющими операторами OVERLAY. Если задана характеристика оверлейной структуры, а управляющие операторы OVERLAY не заданы, то эта характеристика не принимается. Такой случай рассматривается как восстанавливаемая ошибка с кодом серьезности 2, когда при задании параметра LET загрузочный модуль отмечается как выполнимый.

Если характеристика оверлейной структуры не задана, но заданы управляющие операторы OVERLAY и INSERT, то загрузочный модуль не имеет оверлейную структуру. Такой случай также рассматривается как восстанавливаемая ошибка с кодом серьезности 2. Загрузочный модуль оверлейной структуры не может быть реентерабельным (повторно используемым) и обновляемым или иметь формат взброс. Указание параметра OVLY не совместимо с указанием параметров RENT, REUS, REFR, SCTR. Отсутствие параметра OVLY означает, что создаваемый загрузочный модуль не имеет оверлейной структуры.

SCTR — создаваемые загрузочные модули имеют формат взброс. Такой модуль загружается управляющей программой в несмежные области основной памяти в пределах выделенного раздела (загрузка взброс). Загрузочные модули, которые загружаются в смежные области основной памяти, имеют блочный формат (блочная загрузка). Загрузочные модули с форматом взброс пригодны для загрузки взброс и блочной загрузки. Если управляющая программа не обладает свойством загрузки взброс, то загрузочные модули с форматом взброс загружаются блоками. Указание параметра SCTR не совместимо с указанием параметров OVLY и REUS. Отсутствие параметра SCTR означает, что загрузочные модули имеют блочный формат.

OL — создаваемые загрузочные модули имеют характеристику «только загружаемый» и загружаются в основную память с помощью макрокоманды LOAD. Управление такому модулю передается с помощью команды передачи управления или макрокоманды CALL. Попытка загрузки и передачи управления модулю с характеристикой «только загружаемый» с помощью макрокоманд LINK, XCTL или ATTACH приводит к аварийному завершению активной задачи.

Использование загрузочных модулей с указанной характеристикой приводит к экономии основной памяти и сокращению времени выполнения программы. Поскольку управление передается без управляющей программы, последняя не может контролировать этот процесс и обнаруживать ошибки.

NE — создаваемые загрузочные модули имеют характеристику «нередатируемые». Загрузочный модуль с данной характеристикой не имеет в библиотеке словаря внешних имен и не может быть повторно обработан Редактором связей, что приводит к экономии памяти в библиотеке. Эта характеристика делает непригодной функцию **EXPAND** и предназначена только для управляющей программы. Характеристика «нередатируемый» не принимается, если затребованы план модуля или таблица перекрестных ссылок. Указание параметра **NE** не совместимо с указанием параметров **TEST**, **XREF**, **MAP**.

DC — размер записей текста ограничивается 1024 байтами, что обеспечивает возможность размещения библиотеки загрузочных модулей на магнитных дисках любого типа.

TEST — модули подлежат тестированию с помощью Тестрана (для издания 4.1) или команды **TEST** системы разделения времени (для издания 6.1) и содержат необходимые для этого таблицы имен. Указание параметра **TEST** не совместимо с указанием параметра **NE**.

ALIGN4 (только для издания 6.1) — указывает, что выравнивание программных секций в создаваемом Редактором связей загрузочном модуле (с помощью управляющего оператора **PAGE** или управляющего оператора **ORDER** с операндом **P**) должно производиться по границе 4К. Если данный параметр не указан, то выравнивание выполняется по границе 2К.

AC-n (только для издания 6.1) — определяет, что загрузочному модулю присваивается код санкционирования *n*, где *n* — число не более чем из 8 цифр в диапазоне от 0 до 255. Код санкционирования помещается в элемент оглавления библиотеки, соответствующий загрузочному модулю, и может быть перекрыт управляющим оператором **SETCODE**.

8.4.2. Режим работы Редактора связей

LET — создаваемый загрузочный модуль отмечается как выполнимый, даже если получен код серьезности 2. Ошибками, приводящими к возникновению кода серьезности 2, являются: неразрешенные внешние ссылки, исключаяющие вызовы в оверлейной структуре, ошибки в управляющих операторах Редактора связей, а также отсутствие места в оглавлении библиотеки, в которую помещается создаваемый загрузочный модуль, отсутствие модуля в библиотеке и др. При заданном режиме **LET** отпадает необходимость задавать режим **XCAL**.

NCAL — запрет на автоматический вызов библиотеки для разрешения внешних ссылок, которые остались неразрешенными при

обработке основного ввода. Загрузочный модуль, создаваемый Редактором связей, выполнимый даже при наличии неразрешенных внешних ссылок. При заданном режиме NCAL управляющий оператор LIBRARY и оператор SYSLIB DD использовать не следует.

XCAL — режим исключающего вызова. Редактор связей отмечает создаваемый загрузочный модуль как выполнимый, если между сегментами имеются верные исключающие ссылки. Для каждой такой ссылки выдается предупреждающее сообщение. Для модулей оверлейной структуры необходимо задавать характеристику OVLY.

SIZE = (объем 1, объем 2) — объем основной памяти для использования Редактором связей, а также объем основной памяти для текстового буфера, который предназначен для содержания данных ввода и вывода. Если режим SIZE не задан или не указан один из объемов, то используются значения, определенные при генерации операционной системы.

Объем 1 указывает максимальный объем памяти в байтах, доступный для Редактора связей. Он может быть задан либо в форме p , где p — число байт не более 9999999, либо в форме pK , где p — число Кбайт не более 9999. Минимальное количество основной памяти зависит от уровня Редактора связей (для F44—44K, для F64—64K, для F88—88K, для F128—128K).

Объем 2 указывает максимальный объем основной памяти, используемый в качестве буфера загрузочного модуля из объема основной памяти, заданного объемом 1. Объем 2, как и объем 1, может быть задан в байтах и в Кбайтах. Минимальное значение для объема 2 составляет 6K. Буфер загрузочного модуля используется для вводных текстовых записей, записей промежуточных данных и записей создаваемого загрузочного модуля. Таким образом, объем 2 должен вмещать самую большую запись входного и выходного загрузочных модулей, а также промежуточного набора данных (SYSUT1).

Если объем 1 превышает доступный объем основной памяти, то последний автоматически сокращается для Редактора связей, но не менее объема, предусмотренного при генерации операционной системы. После этого выбирается количество основной памяти для буфера загрузочного модуля как максимально возможное значение в диапазоне от 6K до количества, заданного объемом 2. Величина объема 1 должна быть по возможности большей.

DCBS — задает в параметре DCB размер блока (BLKSIZE) для набора данных, описанного в операторе DD с именем SYSLMOD. Если набор данных новый, то размер блока задается программистом, но не более максимально возможного для устройства, содержащего набор данных. При существующем наборе данных используется максимальное значение из заданного программистом и существующего значения, но не более максимально возможного для устройства, содержащего набор данных.

LIST — запрос на вывод управляющих операторов, обработанных Редактором связей, в формате перфокарт в диагностический набор данных.

MAP — запрос на вывод плана создаваемого загрузочного модуля в диагностический набор данных. Указание параметра MAP не совместимо с указанием параметров XREF и NE.

XREF — запрос на вывод таблицы перекрестных ссылок в диагностический набор данных. Указание параметра XREF не совместимо с указанием параметров MAP и NE.

TERM — запрос на вывод нумерованных сообщений об ошибке и предупреждающих сообщений Редактора связей в набор данных, определенный оператором DD с именем SYSTERM. Сообщения появляются как в указанном, так и в диагностическом наборах данных. Если оператор DD с именем SYSTERM отсутствует, то режим TERM не принимается.

8.5. Размер раздела основной памяти

Размер раздела определяется в режимах MVT и SVS параметром REGION оператора EXEC либо устанавливается при генерации или оператором в режиме MFT. Для Редактора связей размер раздела должен быть по крайней мере на 10 Кбайт больше указанного в параметре SIZE значения объема1.

8.6. Коды возврата Редактора связей

Редактор связей после выполнения выдает код возврата, соответствующий максимальному коду серьезности. Код возврата равен максимальному коду серьезности, умноженному на 4, и помещается в регистр 15. В случае, если Редактор связи является программой пункта задания, код возврата может быть проверен с помощью параметра COND оператора JOB или EXEC.

Коды возврата:

- 00 — нормальное выполнение Редактора связей; загрузочные модули отмечены как выполнимые;
- 04 — выполнение Редактора связей считается успешным, но имели место предупреждающие сообщения; загрузочные модули отмечены как выполнимые;
- 08 — имели место сообщения об ошибках; выполнение загрузочных модулей, созданных Редактором связей, может быть неудачным. Такие модули отмечены как невыполнимые, если не указан режим LET, и как выполнимые в обратном случае;
- 12 — допущена серьезная ошибка; выполнение загрузочных модулей невозможно;
- 16 — возникновение ошибки привело к прекращению выполнения Редактора связей.

8.7. Операторы DD для Редактора связей

Редактор связей в процессе своей работы использует шесть наборов данных, четыре из которых являются обязательными. Эти наборы данных описываются операторами DD с именами: SYSLIN, SYSLIB, SYSUT1, SYSPRINT, SYSLMOD, SYSTEM. Кроме того, могут быть использованы дополнительные наборы данных.

SYSLIN — обязательный. Описывает набор данных основного ввода, содержащий объектные модули и (или) управляющие операторы. Этот набор данных может находиться либо на магнитных дисках, либо на магнитной ленте, либо на перфокарточном устройстве ввода. Имеет последовательную или библиотечную организацию. В последнем случае указывается имя раздела библиотеки.

SYSLIB — необязательный. Требуется при использовании автоматического вызова библиотеки для разрешения внешних ссылок, оставшихся неразрешенными в период обработки основного ввода. Оператор описывает автоматически вызываемую библиотеку, расположенную на магнитных дисках. Имена разделов не указываются. Библиотека может содержать либо объектные и (или) управляющие операторы, либо загрузочные модули. При использовании сцепленных библиотек не допускается смешение библиотек объектных и загрузочных модулей.

SYSUT1 — обязательный. Описывает промежуточный набор данных на дисках, имеющий последовательную структуру.

SYSPRINT — обязательный. Описывает диагностический набор данных, содержащий выводимые сообщения Редактора связей, управляющие операторы, план загрузочного модуля, таблицу перекрестных ссылок. Этот набор данных имеет последовательную структуру и для него указывается печатающее устройство или внешнее запоминающее устройство для промежуточного хранения.

SYSLMOD — обязательный. Описывает библиотеку, в которой размещаются созданные Редактором связей загрузочные модули. Библиотека должна располагаться на магнитных дисках. Имя раздела указывается либо вместе с именем библиотеки в операторе DD, либо с помощью управляющего оператора NAME.

SYSTEM — необязательный. Описывает набор данных, предназначенный для вывода нумерованных сообщений об ошибках и предупреждающих сообщений Редактора связей. Этот набор данных применяется в случае, если указан режим TERM параметра PARM оператора EXEC. Обычно он используется с Системой разделения времени для вывода сообщений на терминал (абонентский пункт).

Дополнительные наборы данных Редактора связей описываются дополнительными операторами DD, имена которых определяются в управляющих операторах INCLUDE и LIBRARY. Эти операторы описывают последовательные (INCLUDE) или библиотечные наборы данных, расположенные на магнитных дисках или магнитной ленте.

8.8. Каталогизированные процедуры для Редактора связей

В операционной системе существуют две каталогизированные процедуры для использования Редактора связей с именами LKED и LKEDG.

Вызов процедур, замена, дополнение и исключение параметров и операторов в них производятся по правилам языка управления заданиями, которые изложены в гл. 4. Ниже рассматриваются указанные процедуры, где символ С размещается в позиции 72. В позициях 73—80 располагаются идентификационные номера карт, которые здесь не приводятся.

Процедура LKED состоит из одного пункта, в котором вырабатывается загрузочный модуль, помещаемый во временный библиотечный набор данных. Последний передается следующим пунктам текущего задания. Ниже приводится текст процедуры LKED:

```
//LKED      EXEC    PGM=IEWL,PARM='XREF,LIST,LET,NCAL',— —C
//          REGION=96K
//SYSPRINT   DD      SYSOUT=A
//SYSLIN     DD      DDNAME=SYSIN
//SYSLMOD    DD      DSNNAME=&&GOSET(GO),SPACE=(1024,      C
//          (50,20,1)),
//          UNIT=SYSDA,DISP=(MOD,PASS)                      C
//SYSUTI     DD      UNIT=SYSDA,SEP=(SYSLMOD,SYSLIN),      C
//          SPACE=(1024,200,20)
```

Процедура LKEDG состоит из двух пунктов, в первом из которых выполняется Редактор связей, вырабатывающий загрузочный модуль, а во втором производится его выполнение. Загрузочный модуль помещается во временный библиотечный набор данных, передаваемый второму пункту. Начало текста процедуры LKEDG совпадает с текстом процедуры LKED, за ним следует оператор

```
//GO EXEC PGM=*.LKED.SYSLMOD,COND=(4,LT,LKED)
```

Процедура LKEDG предусматривает, что созданный загрузочный модуль выполняется, если Редактор связей имеет код возврата 0 или 4. При коде возврата 8 загрузочный модуль создается как выполнимый, но не выполняется. Для его выполнения необходимо перекрыть параметр COND пункта GO процедуры LKEDG, установив значение COND=(8, LT, LKED).

8.9. Динамический вызов Редактора связей

Редактор связей может быть вызван из программы динамически по макрокомандам LINK, ATTACH, LOAD или XCTL.

Формат макрокоманд приведен в гл. 5. В качестве имени программы в этих макрокомандах необходимо употреблять основные или дополнительные имена Редактора связей, рассмотренные выше. Операнд PARAM в макрокомандах LINK и ATTACH для вызова Редактора связей имеет следующий формат:

```
PARAM=(адрес-1 [,адрес-2])
```

где адрес-1 — адрес списка, содержащего параметры Редактора связей, определяющие режимы его работы и характеристики загрузочных модулей; должен начинаться на границе полуслова. Два старших байта содержат количество байт в списке. При отсутствии параметров адрес-1 должен быть указан обязательно. Список имеет свободную форму. Параметры записываются так же, как в поле PARM оператора EXEC, вызывающего Редактор связей;

адрес-2 — адрес списка имен операторов DD для наборов данных, применяемых в период работы Редактора связей; должен начинаться на границе полуслова. Два старших байта содержат количество байт в списке. При использовании стандартных имен указание адреса-2 можно опустить. Список имеет свободную форму и предназначен для указания имен операторов DD, которые применяются вместо стандартных.

Остальные операнды макрокоманд используются в соответствии с их назначением.

8.10. Управляющие операторы Редактора связей

8.10.1. Включение дополнительного ввода Редактора связей

Оператор **INCLUDE** определяет наборы данных для дополнительного ввода Редактора связей. Наборы данных могут иметь последовательную или библиотечную организацию. Последовательные наборы данных содержат объектные модули и (или) управляющие операторы; библиотечные — либо объектные модули и (или) управляющие операторы, либо загрузочные модули. Операторы **INCLUDE** обрабатываются в порядке их появления. Формат оператора:

INCLUDE имя-оператора-dd [имя-раздела,...],...

где имя-оператора-dd — имя оператора DD, содержащее описание последовательного или библиотечного набора данных. Для библиотечного набора указывается по крайней мере одно имя раздела;

имя-раздела — имя или дополнительное имя раздела библиотечного набора данных.

8.10.2. Дополнительные автоматически вызываемые библиотеки

Оператор **LIBRARY** определяет дополнительные автоматически вызываемые библиотеки, предназначенные для разрешения внешних ссылок в программе. Вызываемая библиотека может содержать либо объектные модули и (или) управляющие операторы, либо загрузочные модули. С помощью оператора **LIBRARY** можно ограничить разрешение внешних ссылок аппаратом автоматического вызова либо на период выполнения текущего пункта задания

Редактора связей, либо всех пунктов Редактора связей. Формат оператора:

$$\text{LIBRARY } \left\{ \begin{array}{l} \text{имя-оператора-dd(имя-раздела,...)} \\ \text{(внешняя-ссылка,...)} \\ \text{*(внешняя-ссылка,...)} \end{array} \right\}, \dots$$

где имя-оператора-dd — имя оператора DD, содержащее описание библиотеки автоматического вызова;

имя-раздела — имя или дополнительное имя раздела библиотеки, который используется для разрешения внешних ссылок;

(внешняя-ссылка,...) — список внешних ссылок, которые не разрешаются с помощью аппарата автоматического вызова при выполнении текущего пункта задания Редактора связей;

*(внешняя-ссылка,...) — список внешних ссылок, которые не разрешаются с помощью аппарата автоматического вызова при выполнении всех пунктов Редактора связей.

8.10.3. Определение имени, дополнительного имени и точки входа загрузочного модуля

Оператор NAME определяет имя загрузочного модуля (имя раздела библиотеки), созданного из модулей с помощью операторов, предшествующих данному оператору. Оператор NAME является ограничителем загрузочного модуля во входном потоке. С его помощью можно указать, что созданный загрузочный модуль должен заменить существующий модуль с тем же именем в библиотеке выходных модулей. Формат оператора:

$$\text{NAME имя-раздела}[(R)]$$

где имя-раздела — имя созданного загрузочного модуля;

(R) — означает, что созданный загрузочный модуль должен заменить существующий модуль с тем же именем в библиотеке выходных модулей, определенной оператором //SYSLMOD DD.

Оператор ALIAS определяет дополнительное имя загрузочного модуля (дополнительное имя раздела библиотеки), а также дополнительные имена точек входа. Для загрузочного модуля (раздела библиотеки) допускается до 16 дополнительных имен. Формат оператора:

$$\text{ALIAS } \left\{ \begin{array}{l} \text{имя} \\ \text{внешнее-имя} \end{array} \right\} \dots$$

где имя — дополнительное имя загрузочного модуля, соответствующее основной точке входа загрузочного модуля;

внешнее-имя — дополнительное имя, которое в модуле является именем программной секции или именем точки входа.

Оператор ENTRY определяет основную точку входа, используемую при вызове модуля по имени. Формат оператора:

$$\text{ENTRY внешнее-имя}$$

где внешнее-имя — имя основной точки входа, которое в модуле является именем программной секции или именем точки входа.

8.10.4. Редактирование модуля

Оператор REPLACE позволяет определить замену или удаление программной секции, удаление имени точки входа. Формат:

REPLACE { имя-1[(имя-2)]
 { имя-точки-входа } } ,...

где имя-1 и имя-2 — имена программных секций. Если применяется только имя-1, то оно определяет имя удаляемой программной секции. Если используются имя-1 и имя-2, то первое из них означает имя замещаемой, второе — имя замещающей программных секций;

имя-точки-входа — удаляемое имя точки входа.

Оператор CHANGE определяет замену внешнего имени на новое имя. Формат оператора:

CHANGE внешнее-имя(имя),...

где внешнее-имя — замещаемое имя, которое может быть либо именем программной секции, либо именем точки входа, либо внешней ссылкой;

имя — новое имя, на которое производится замена.

Оператор ORDER (только в издании 6.1) определяет последовательность, в которой программные секции или поименованные общие области должны быть расположены в загрузочном модуле. Если для загрузочного модуля задано несколько операторов ORDER, то последовательность программных секций должна соответствовать последовательности операторов. Формат оператора:

ORDER имя[(P)],...

где имя — имя программной секции или общей области, которую нужно упорядочить;

(P) — означает, что начало программной секции или общей области с предшествующим именем должно быть выравнено по границе страницы. Выравнивание производится по границе 2K, если не задан режим ALIGN4 в операторе EXEC, и по границе 4K в противном случае.

Оператор PAGE (только в издании 6.1) определяет выравнивание программной секции или поименованной общей области по границе страницы. Формат оператора:

PAGE имя,...

где имя — имя программной секции или общей области, начало которой должно быть выравнено по границе страницы. Выравнивание производится по границе 2K, если не задан режим ALIGN4 в операторе EXEC, и по границе 4K в противном случае.

Оператор EXPAND (только в издании 6.1) позволяет увеличить размер программной секции или поименованной общей области на заданное количество байт. Формат оператора:

EXPAND имя(число),...

где имя — имя программной секции или общей области, размер которой должен быть увеличен;

число — десятичное число не более 4095 байт, на которое увеличивается размер программной секции или общей области.

Оператор IDENTIFY (только в издании 6.1) определяет данные, предназначенные для занесения в запись идентификации программной секции. Формат оператора:

IDENTIFY имя('данные'),...

где имя — имя программной секции;

данные — данные идентификации в коде ДКОИ не более 40 байт, предназначенные для занесения в запись идентификации.

Оператор SETCODE (только в издании 6.1) присваивает загрузочному модулю определенный код санкционирования, который помещается в элемент оглавления библиотеки, соответствующий загрузочному модулю. Формат оператора:

SETCODE AC (код-санкционирования)

где код-санкционирования — десятичное число от 0 до 255, состоящее не более чем из 8 десятичных цифр.

Оператор SETSSI определяет шестнадцатеричный код в индексе состояния загрузочного модуля, находящегося в элементе оглавления библиотеки. Формат оператора:

SETSSI код

где код — шестнадцатеричный код из восьми цифр в индексе состояния загрузочного модуля.

8.10.5. Оверлейные программы

Оператор OVERLAY определяет начало и указывает имя сегмента или области оверлейной программы. Формат оператора:

OVERLAY имя[(REGION)]

где имя — имя, присвоенное началу сегмента;

REGION — указывает начало новой области.

Перед корневым сегментом оператор OVERLAY не используется. Последовательность операторов OVERLAY соответствует порядку сегментов в оверлейной структуре программы сверху вниз, слева направо и область за областью. Применение операторов OVERLAY допускается только в режиме OVLY, указанном в операторе EXEC.

Оператор INSERT определяет перемещение программной секции из входной последовательности в сегмент оверлейной структуры. Формат оператора:

INSERT имя,...

где имя — имя программной секции, перемещаемой в сегмент оверлейной структуры.

Оператор INSERT следует размещать вслед за оператором OVERLAY, определяющим начало сегмента. Если секция перемещается в корневой сегмент, то оператор INSERT размещается перед первым оператором OVERLAY.

Загрузчик

9.1. Функции Загрузчика

Загрузчик выполняет ряд функций Редактора связей и программы выборки. Он преобразует объектные модули, составляющие результат работы трансляторов и Ассемблера, в загрузочный модуль, размещает его в основной памяти и передает ему управление. При этом загрузочный модуль не записывается в библиотечный набор данных. Таким образом Загрузчик осуществляет редактирование и выполнение загрузочного модуля в одном пункте задания. Загрузчик используется в случае, если не требуются специальные режимы Редактора связей и запоминание загрузочного модуля. При этом применение Загрузчика существенно сокращает время редактирования и выполнения загрузочного модуля по сравнению с использованием Редактора связей. Загрузчик является реентерабельной программой и может находиться в общей области. В этом случае одна копия Загрузчика может удовлетворить все запросы на его использование, что приводит к экономии основной памяти. Загрузчик выполняет следующие функции:

- объединяет поступающие на ввод объектные и загрузочные модули в один загрузочный модуль;

- разрешает внешние ссылки между отдельными модулями;

- включает модули из автоматически вызываемой библиотеки и (или) из общей области для разрешения внешних ссылок, оставшихся неразрешенными после обработки основного ввода Загрузчика;

- удаляет дублирующие копии программных модулей и автоматически редактирует связи;

- производит настройку всех адресных констант загрузочного модуля на ту область основной памяти, в которой модуль будет выполняться.

Загрузчик может быть использован в системе разделения времени. В этом случае он вызывается пользователем системы через команду LOADGO, в которой определяются наборы данных и режимы Загрузчика, необходимые при его выполнении.

Загрузчик обрабатывает следующие режимы Редактора связей: NCAL, LET, MAP, SIZE и TERM. Кроме того, он имеет и соб-

ственные режимы. Остальные режимы и управляющие операторы Редактора связей Загрузчик не обрабатывает, но их указание не считается ошибкой.

9.2. Оператор EXEC для Загрузчика

Оператор EXEC определяет имя Загрузчика (параметр PGM), режимы его работы и параметры выполняемой программы — загрузочного модуля (параметр PARM), а также может содержать другие параметры, необходимые в каждом конкретном случае.

Для вызова программы Загрузчика используется либо основное имя IEWLDRGO, либо дополнительное LOADER. Параметр PGM оператора EXEC, вызывающего Загрузчик, должен иметь один из следующих форматов:

PGM=IEWLDRGO или PGM=LOADER

Режимы работы Загрузчика и параметры выполняемой программы могут быть заданы путем указания параметров в поле PARM оператора EXEC, вызывающего Загрузчик, в следующем формате:

PARM='[параметр-Загрузчика],.../[параметр-программы],...'

Параметры Загрузчика отделяются от параметров программы символом `"/`. При отсутствии параметров Загрузчика перед параметром программы ставится символ `"/`. Если параметры Загрузчика и программы отсутствуют, поле PARM можно опустить. При использовании в поле параметров специальных символов параметры заключаются в одиночные кавычки.

Параметры Загрузчика означают следующие режимы работы:
CALL — использование автоматически вызываемой библиотеки, описанной в операторе DD с именем SYSLIB, для разрешения внешних ссылок, оставшихся неразрешенными после обработки основного ввода. При отсутствии оператора DD с именем SYSLIB этот режим игнорируется.

NOCALL (или NCAL) — автоматически вызываемая библиотека для разрешения внешних ссылок не используется. Этот режим автоматически устанавливает режим NORES;

RES — для разрешения внешних ссылок следует использовать общую область. Разрешение внешних ссылок производится в последовательности: сначала осуществляется обработка основного ввода, затем оставшиеся неразрешенными внешние ссылки разрешаются относительно общей области и, наконец, оставшиеся неразрешенными внешние ссылки разрешаются относительно автоматически вызываемой библиотеки. Режим RES устанавливает также режим CALL.

NORES — общая область для разрешения внешних ссылок не используется.

LET — управление передается созданной Загрузчиком программе даже в том случае, если будут обнаружены ошибки с кодом серьезности 2.

NOLET — созданной Загрузчиком программе управление не передается, если при обнаружении ошибок код серьезности 2. NOLET является параметром по умолчанию.

MAP — запрос на план программы, создаваемой Загрузчиком, который должен быть помещен в набор данных описываемый оператором DD с именем SYSLOUT. При отсутствии оператора DD режим MAP игнорируется.

NOMAP — план создаваемой Загрузчиком программы не вырабатывается.

SIZE — объем (в байтах) области основной памяти, которая может использоваться Загрузчиком. По умолчанию предполагается SIZE=100K.

EP — внешнее имя, являющееся точкой входа программы, создаваемой Загрузчиком.

NAME — имя, которое должно быть использовано для идентификации в операционной системе программы, создаваемой Загрузчиком. При отсутствии этого параметра по умолчанию программа получает имя **GO.

PRINT — информационные и диагностические сообщения, вырабатываемые Загрузчиком, записываются в набор данных, определяемый оператором DD с именем SYSLOUT. PRINT является параметром по умолчанию.

NOPRINT — информационные и диагностические сообщения не вырабатываются.

TERM — нумерованные диагностические сообщения должны быть записаны в набор данных, определяемый оператором DD с именем SYSTERM. При отсутствии этого оператора режим TERM игнорируется.

NOTERM — нумерованные диагностические сообщения не записываются в набор данных, определяемый оператором DD с именем SYSTERM. NOTERM является параметром по умолчанию.

Указанные выше параметры по умолчанию могут быть изменены при генерации операционной системы с помощью макрокоманды LOADER.

9.3. Операторы DD для Загрузчика

Загрузчик в процессе работы использует четыре набора данных, из которых обязательным является только один. Наборы данных описываются операторами DD с именами SYSLIN, SYSLIB, SYSLOUT и SYSTERM. Эти имена могут быть изменены при генерации операционной системы с помощью макрокоманды LOADER. Поскольку программа, создаваемая Загрузчиком, выполняется в том же пункте, что и Загрузчик, то операторы DD для этой программы должны размещаться за операторами DD для Загрузчика.

SYSLIN. Оператор DD с именем SYSLIN определяет основной ввод Загрузчика и описывает наборы данных, содержащие как объектные, так и загрузочные модули. Оператором могут быть описаны либо последовательные наборы данных, либо разделы биб-

лиотек. Для включения в основной ввод нескольких модулей применяются сцепленные операторы DD.

SYSLIB. Оператор DD с именем SYSLIB является необязательным. Он требуется в случае, если применяется автоматический вызов библиотеки для разрешения внешних ссылок, оставшихся неразрешенными после обработки основного ввода, и если задается режим RES общей области. Для данного оператора задается режим CALL или RES. Набор данных, определенный этим оператором, должен иметь библиотечную структуру и содержать объектные или загрузочные модули. Допускается использование сцепленных операторов DD, но при этом библиотечные наборы данных должны содержать либо объектные, либо загрузочные модули.

SYSLOUT. Оператор DD с именем SYSLOUT является обязательным. Он используется для вывода сообщений об ошибках и плана программы, создаваемой Загрузчиком при заданных режимах MAP и PRINT. Набор данных, описываемый этим оператором, должен иметь последовательную структуру.

SYSTEM. Оператор DD с именем SYSTEM является обязательным. Он используется для вывода нумерованных диагностических сообщений при заданном режиме TERM. Набор данных, описываемый оператором DD с именем SYSTEM, может использоваться в Системе разделения времени для вывода сообщений на терминал (абонентский пункт) и в режиме пакетной обработки, либо вместо набора данных SYSLOUT, либо в дополнение к нему. Набор данных SYSTEM не открывается до появления диагностических сообщений, поэтому его использование вместо SYSLOUT сокращает время работы Загрузчика.

9.4. Динамический вызов Загрузчика

Загрузчик может быть вызван динамически из программы по макрокомандам LINK, ATTACH, LOAD или XCTL. Формат макрокоманд приведен в гл. 5. При использовании Загрузчика для редактирования и выполнения программ в качестве имени программы в этих макрокомандах применяется основное (IEWLDRGO) или дополнительное (LOADER) имя Загрузчика. Операнд PARAM в макрокомандах LINK и ATTACH для вызова Загрузчика имеет следующий формат:

PARAM=(адрес-1[адрес-2])

где адрес-1 — адрес списка, имеющего параметры Загрузчика и (или) создаваемой им программы. Список должен начинаться на границе полуслова. Два старших байта содержат счетчик байт в списке. Адрес-1 указывается обязательно даже при отсутствии параметров. Список имеет свободную форму. Параметры записываются так же, как в поле PARM оператора EXEC, вызывающего Загрузчик;

адрес-2 — адрес списка имен операторов DD для наборов данных, применяемых при работе Загрузчика. Два старших байта содержат счетчик байт в списке. При использовании стандартных имен указание адреса-2 можно опустить. Список предназначен для

указания имен операторов DD, применяющихся вместо стандартных.

Остальные операнды макрокоманд используются в соответствии с их назначением.

Если Загрузчик предназначен только для редактирования без передачи управления созданной программе, то применяется имя IEWLOADR или IEWLOAD для вызова Загрузчика с помощью макрокоманд LOAD и CALL. В этом случае после редактирования адрес созданной Загрузчиком программы передается в регистре 0.

При имени IEWLOADR в регистре 1 находится адрес двух полных слов: первое содержит начальный адрес созданной Загрузчиком программы, второе — размер этой программы в байтах. Эта информация может быть использована для освобождения с помощью макрокоманды FREEMAIN основной памяти, занятой программой, после завершения ее выполнения. Управление созданной Загрузчиком программе передается макрокомандой CALL.

При имени IEWLOAD регистр 1 содержит адрес поля из 8 байт, в котором находится имя созданной Загрузчиком программы. Это поле используется затем в макрокомандах LINK, ATTACH, LOAD или XCTL для вызова этой программы.

9.5. Коды возврата Загрузчика

Код возврата пункта задания, в котором выполняется Загрузчик, определяется кодом возврата Загрузчика и кодом возврата программы, созданной Загрузчиком.

Код возврата Загрузчика равен максимальному коду серьезности, возникшему при его выполнении, умноженному на 4. Этот код помещается в регистр 15. Коды серьезности Загрузчика имеют следующие значения:

0 — нормальное выполнение Загрузчика;

1 — обнаружены условия, которые могут привести к ошибке при выполнении созданной Загрузчиком программы. Указанная программа получает управление;

2 — обнаружены ошибки, из-за которых выполнение созданной Загрузчиком программы может быть невозможным. Эта программа получает управление при заданном режиме LET и не получает его в обратном случае;

3 — обнаружены ошибки, которые делают невозможным выполнение созданной Загрузчиком программы. Управление этой программе не передается;

4 — обнаружены ошибки, которые привели к прекращению работы Загрузчика.

Код возврата пункта задания, в котором выполняется Загрузчик, определяется следующим образом:

при кодах возврата Загрузчика 0, 4 и 8 (в режиме LET) он равен коду возврата созданной Загрузчиком программы;

при кодах возврата Загрузчика 8 (в режиме NOLET), 12 и 16 он равен коду возврата Загрузчика; при этом созданная Загрузчиком программа не выполняется.

Программа сортировка-объединение

10.1. Функции программы

Программа СОРТИРОВКА-ОБЪЕДИНЕНИЕ (ПСО) входит в состав системных обрабатывающих программ операционной системы ОС ЕС и предназначена для упорядочения наборов данных с последовательной организацией.

ПСО выполняет следующие функции:

сортирует набор данных в определенную последовательность; объединяет от 2 до 16 отсортированных наборов данных в один отсортированный.

10.1.1. Общие сведения

Последовательность, в которую упорядочивается набор данных, зависит от содержания управляющего слова каждой записи набора данных. Управляющие слова определяются в управляющем операторе SORT. Каждое управляющее слово имеет длину до 256 байт и может содержать от 1 до 64 управляющих полей в первых 4092 байтах записи. Управляющие поля имеют следующие форматы: символьный, с фиксированной точкой, нормализованный с плавающей точкой, упакованный десятичный, зонный десятичный, двоичный, специальный.

В зависимости от формата управляющие поля имеют следующую длину: символьный, специальный, с фиксированной точкой и нормализованный с плавающей точкой — от 1 до 256 байт; упакованный десятичный — от 1 до 16 байт; зонный десятичный — от 1 до 32 байт; двоичный — от 1 до 256 байт.

Каждое управляющее поле служит для упорядочения набора данных в порядке возрастания или убывания значения этого поля. Изменяя управляющие поля с помощью программ пользователя ЭВМ, можно влиять на процесс сортировки. Модифицированные управляющие поля не заменяют соответствующие управляющие поля в самой записи, а используются только для сопоставления.

В ПСО предусмотрены специальные выходы для подключения в определенных точках программ пользователя ЭВМ, предназначенных для выполнения дополнительных функций.

Обрабатываемые ПСО входные и выходные наборы данных могут состоять из сблокированных или несблокированных записей фиксированной или переменной длины (RECFM=F, V, FB или

VB). Также могут обрабатываться расширенные (сегментированные) записи (RECFM=VS или VBS). Можно использовать наборы данных, обработка которых возможна с применением последовательного метода доступа с очередями (QSAM). Для ввода и вывода данных применяются любые периферийные устройства, обеспечиваемые последовательным методом доступа с очередями.

Программа СОРТИРОВКА-ОБЪЕДИНЕНИЕ использует четыре метода сортировки: балансовый, многофазный, осцилляторный, перекрестный. Если для промежуточной памяти предназначается накопитель на магнитных лентах, то применяется балансовый, осцилляторный или многофазный метод, если накопители на магнитных дисках, то балансовый или перекрестный метод.

При выполнении программы ПСО могут создаваться записи контрольных точек для последующего повторного пуска задания с одной из этих точек. Для создания контрольных точек в управляющем операторе SORT указывается операнд СКРТ и определяется специальный набор данных.

При определении наборов данных, используемых ПСО, применяются операторы DD со следующими именами:

SORTIN — для входного набора данных, который должен быть отсортирован;

SORTIN01...SORTIN16 — для входных наборов данных, которые должны быть объединены;

SORTWK01...SORTWK32 — для промежуточных наборов данных, используемых при сортировке;

SORTOUT — для выходного набора данных при сортировке или объединении;

SORTMODS — для временного библиотечного набора данных, предназначенного для всех программ пользователя ЭВМ, включенных во входной поток;

SORTCKPT — для набора данных, используемого для записи контрольных точек;

SORTLIB — для библиотечного набора данных, содержащего загрузочные модули ПСО;

SYSIN — для набора данных, содержащего управляющие операторы ПСО;

SYSOUT — для набора данных, используемого в качестве системного выходного набора данных.

Для применения программ пользователя ЭВМ, которые еще не были обработаны Редактором связей, необходимы дополнительные операторы DD со следующими именами:

SYSLIN — для входного набора данных, используемого Редактором связей;

SYSLMOD — для выходного набора данных, применяемого Редактором связей;

SYSUT1 — для рабочего набора данных, используемого Редактором связей;

SYSPRINT — для набора данных, применяемого Редактором связей для диагностических сообщений.

10.1.2. Требования к конфигурации ЭВМ

Технические средства должны обеспечивать функционирование операционной системы ОС ЕС и выполнение системных обрабатывающих программ. В частности, для ПСО минимально требуется:

20 Кбайт основной памяти, при этом собственно для программы необходимо 15 Кбайт, а остаток используется операционной системой. Для эффективного выполнения сортировки требуется не менее 44 Кбайт;

один накопитель на магнитных дисках (НМД);

три накопителя на магнитных лентах (НМЛ).

НМД может быть одновременно системным резидентом. Применяемые НМД определяются типовыми именами 5050, 5061 и 5066, НМЛ — типовыми именами 5010 и 5000. НМЛ и НМД используются для промежуточной памяти (максимально: 32 НМЛ; 6 НМД с типовым именем 5050, 17 НМД с типовыми именами 5061 или 5066). В пункте задания все устройства для промежуточной памяти должны быть одного типа.

Размер основной памяти, отводимой для программы СОРТИРОВКА-ОБЪЕДИНЕНИЕ, обуславливает максимальную длину записей входных и выходных наборов данных, обрабатываемых программой и используемых в качестве промежуточной памяти (табл. 10.1).

Таблица 10.1

Основная память для ПСО, Кбайт	Внешнее устройство для промежуточной памяти									
	НМЛ, макс. длина в байтах		НМД, макс. длина в байтах							
			5050		5061				5066	
					3 рабочие области		6 рабочих областей		6 рабочих областей	
20	1 100	800	1 300	900	1 300	900	—	—	—	—
45	5 500	4 600	3 600	3 600	6 600	5 100	3 500	3 100	3 300	2 900
100	14 900	12 800	3 600	3 600	7 200	7 200	7 200	7 200	12 500	11 000
200	32 700	27 400	3 600	3 600	7 200	7 200	7 200	7 200	13 000	13 000
и более										
	F, V	VS	F, V	VS	F, V	VS	F, V	VS	F, V	VS
Формат записей										

10.2. Управляющие операторы программы

Управляющие операторы ПСО подготавливаются в формате 80-позиционной перфокарты и содержат: поле операции, поле операндов и поле комментариев, разделенные не менее одним пробелом:

Операция	Операнды	Комментарии
----------	----------	-------------

Для полей оператора отводятся позиции с 1-й по 71-ю. Первая позиция перфокарты с управляющим оператором должна иметь пробел. Оператор можно продолжить на несколько перфокарт.

В поле операции указывается тип управляющего оператора (имеется пять допустимых типов): SORT, MERGE, RECORD, MODS и END. Поле операции должно находиться на первой перфокарте, если оператор занимает более одной перфокарты.

Поле операндов содержит параметры для программы СОРТИРОВКА-ОБЪЕДИНЕНИЕ. Это второе поле оператора, которое должно начинаться также на первой перфокарте, если оператор занимает более одной перфокарты. Несколько операндов разделяются запятыми. Возможны следующие форматы операндов:

КЛЮЧЕВОЕ-СЛОВО

КЛЮЧЕВОЕ-СЛОВО=ЗНАЧЕНИЕ

КЛЮЧЕВОЕ-СЛОВО=(ЗНАЧЕНИЕ1, ЗНАЧЕНИЕ2,..., ЗНАЧЕНИЕ_n)

Последовательность операндов в поле операндов может быть любой. Внутри операндов не допускаются пробелы. Значения могут состоять максимум из восьми алфавитно-цифровых символов. Запятые, пробелы, скобки, знаки равенства используются как разграничители и не могут применяться в значениях.

Поле комментариев может содержать любую информацию, необходимую пользователю ЭВМ.

Любой символ, отличный от пробела, в позиции 72 означает, что имеется перфокарта продолжения. Позиции 73—80 могут использоваться для любых целей, например для нумерации перфокарт. На перфокарте продолжения позиции 1—15 должны иметь пробелы. Для каждого типа оператора установлено максимальное число перфокарт продолжения: SORT — 19, MERGE — 19, RECORD — 5, MODS — 19; END — не имеет продолжения.

Тип управляющего оператора может быть включен только один раз для каждого выполнения программы. Управляющие операторы SORT и MERGE не включаются одновременно. Общее число перфокарт с управляющими операторами, включая перфокарты продолжения, должно быть не более 33.

Управляющие операторы имеют следующие предназначения:

SORT — для описания управляющих полей, по которым будет производиться сортировка набора данных. Формат оператора:

$$\text{SORT} \left\{ \begin{array}{l} \text{FIELDS}=(P_1, M_1, F_1, S_1, \dots, P_{64}, M_{64}, F_{64}, S_{64}) \\ \text{FIELDS}=(P_1, M_1, S_1, \dots, P_{64}, M_{64}, S_{64}), \text{FORMAT}=\text{XX} \end{array} \right\} \left\{ \begin{array}{l} \text{[SIZE=Y]} \\ \text{[SKEPREC=Z]} \\ \text{[CKPT]} \end{array} \right.$$

где FIELDS — описание управляющих полей для операции сортировки:

P_i — адрес управляющего поля относительно начала записи ($1 \leq i \leq 64$); M_i — длина i -го управляющего поля; F_i — формат данных в управляющем поле, которое может принимать одно из следующих значений: CH — символьный, ZD — зонный десятичный, PD — упакованный десятичный, FI — с фиксированной точкой, BI — двоичный, FL — с плавающей точкой.

кой, SC — специальный (если все управляющие поля содержат один и тот же тип данных, то этот параметр может быть опущен и использован дополнительный операнд **FORMAT=XX**); S_i — направление упорядочения набора данных по i -му управляющему полю: A — возрастающая последовательность, D — убывающая последовательность, E — модификация управляющего поля с помощью программы пользователя ЭВМ по выходу E61;

SIZE=Y — указывает число записей во входном наборе данных;

SKIPREC=Z — используется для задания пропуска Z записей входного набора данных перед обработкой;

СКРТ — применяется для задания записей контрольных точек.

MERGE — для задания операции объединения. Формат оператора:

$$\text{MERGE} \left\{ \begin{array}{l} \text{FIELDS}=(P_1, M_1, F_1, S_1, \dots, P_{64}, M_{64}, F_{64}, S_{64}) \\ \text{FIELDS}=(P_1, M_1, S_1, \dots, P_{64}, M_{64}, S_{64}), \text{FORMAT}=\text{XX} \end{array} \right\} [\text{SIZE}=\text{Y}]$$

Формат оператора **MERGE** аналогичен формату оператора **SORT**, но имеются следующие отличия: ключевым словом является **MERGE**; не используются операнды **SKIPREC** и **СКРТ**; значение операнда **SIZE=Y** определяет общее число записей во всех входных наборах данных.

RECORD — для описания формата и длины записей, подлежащих сортировке или объединению. Оператор необходим при использовании программы пользователя ЭВМ для изменения длины записей в период выполнения ПСО. Формат оператора:

$$\text{RECORD} \left\{ \begin{array}{l} \text{TYPE}=\text{F}, \text{LENGTH}=(L_1, L_2, L_3) \\ \text{TYPE}=\text{V}, \text{LENGTH}=(L_1, L_2, L_3, L_4, L_5) \end{array} \right\}$$

где **TYPE** — формат записей входного набора данных;

F — запись фиксированной длины;

V — запись переменной длины;

LENGTH — длина обрабатываемых записей:

L_1 — длина записи входного набора данных ($L_1 \geq 18$ байт);

L_2 — длина записи на входе фазы сортировки (L_2 не используется при объединении наборов данных); L_3 — длина записи выходного набора данных. При записи переменной длины L_1, L_2, L_3 означают максимальную длину записей; L_4 — минимальную длину записи входного набора данных; L_5 — наиболее вероятную длину записи входного набора данных.

Для записей фиксированной длины L_4, L_5 не используются.

MODS — для передачи управления на программы пользователя ЭВМ из разных выходов программы СОРТИРОВКА-ОБЪЕДИНЕНИЕ при ее выполнении. Формат оператора:

$$\text{MODS} \langle \text{ВЫХОД} \rangle = (n_1, m_1, s_1 \left[\begin{array}{c} N \\ S \end{array} \right]), \dots, \langle \text{ВЫХОД} \rangle = (n_{17}, m_{17}, s_{17} \left[\begin{array}{c} N \\ S \end{array} \right])$$

где $\langle \text{ВЫХОД} \rangle$ — определяет идентификатор выхода в формате E_{ij} (всего возможно 17 выходов; см. табл. 10.2);

Функции выходов программы СОРТИРОВКА-ОБЪЕДИНЕНИЕ

Функции	Выходы																
	фаза сортировки						фаза промежу- точного объеди- нения					фаза заключи- тельного объеди- нения					все фазы
	E11	E15	E16	E17	E18	E19	E21	E25	E27	E28	E29	E31	E35	E37	E38	E39	E61
Открытие наборов данных	×						×					×					
Закрытие наборов данных				×					×					×			
Добавление записей		×											×				
Изменение записей		×						×					×				
Удаление записей		×						×					×				
Итоговая обработка записей								×					×				
Чтение входного набора данных (ПСО в качестве подпрограммы)		×															
Запись выходного набора данных (ПСО в качестве подпрограммы)													×				
Подавление контроля последовательности													×				
Выход по служебной ссылке	×						×					×					
Обработка при пре- вышении N_{max}			×														
Обработка ошибок чтения					×				×						×		
Обработка конца на- бора данных					×												
Проверка меток пользователя ЭВМ					×												
Обработка ошибок записи						×				×						×	
Модификация управ- ляющих полей записи																	×

n — имя программы пользователя ЭВМ (имя раздела библиотеки, если программа находится в библиотеке);

m — размер основной памяти (точный или приблизительный, в байтах), который занимает программа пользователя ЭВМ;

s — имя оператора DD, который определяет библиотечный набор данных, содержащий программу пользователя ЭВМ, или SYSIN, если программа пользователя ЭВМ находится во входном потоке;

$\begin{Bmatrix} N \\ S \end{Bmatrix}$ — параметры, определяющие требования к редактированию

программ пользователя ЭВМ:

N — не требуется повторное редактирование;

8 — требуется повторное редактирование (такие программы могут подключаться только к выходам E11, E21 и E31).
END — для указания конца всех управляющих операторов для конкретного задания. Формат оператора:

END

Оператор не имеет операндов.

10.3. Выходы на программы пользователя ЭВМ

Все имена выходов имеют форму Eij, где i и j — цифры. Цифра i соответствует фазе, к которой относится выход:

- 1 — фаза сортировки;
- 2 — фаза промежуточного объединения;
- 3 — фаза заключительного объединения.

Цифра j отражает тип функции, которая может быть выполнена программой пользователя ЭВМ, подключаемой на этом выходе. Для изменения управляющих полей используется программа пользователя ЭВМ, подключаемая на выходе E61 компонента исполнения каждой из трех фаз. В табл. 10.2 приведены функции всех выходов. При этом для каждого выхода указаны только типичные функции.

Связь с программами пользователя ЭВМ основывается на использовании общих регистров согласно следующим соглашениям:

R1 — используется для передачи адреса списка параметров вызываемой программе;

R13 — содержит адрес области, в которой программа пользователя ЭВМ может сохранить содержимое любых регистров;

R14 — содержит адрес точки возврата в программу СОРТИРОВКА-ОБЪЕДИНЕНИЕ;

R15 — содержит адрес программы пользователя ЭВМ. Используется для кода возврата, передаваемого в программу СОРТИРОВКА-ОБЪЕДИНЕНИЕ.

10.4. Вызов программы СОРТИРОВКА-ОБЪЕДИНЕНИЕ

Программа может вызываться как основная с помощью оператора EХЕС, который с необходимыми операторами DD и управляющими операторами программы СОРТИРОВКА-ОБЪЕДИНЕНИЕ должен следовать во входном потоке, или как подпрограмма с помощью макрокоманд АТТАСН, LINK или ХСТЛ в какой-либо выполняемой программе. Информация об управляющих операторах для программы СОРТИРОВКА-ОБЪЕДИНЕНИЕ должна находиться в списке параметров вызывающей программы.

10.5. Операторы языка управления заданиями для программы СОРТИРОВКА-ОБЪЕДИНЕНИЕ

Во входном потоке операционной системы задание на выполнение ПСО должно включать операторы языка управления заданиями и управляющие операторы ПСО. Операторы языка управления заданиями содержат операторы JOB, EXEC и связанные с ними операторы DD.

Оператор JOB предназначен для идентификации задания и указания дополнительной информации (по усмотрению программиста) в соответствии с правилами языка управления заданиями.

Оператор EXEC определяет имя программы СОРТИРОВКА-ОБЪЕДИНЕНИЕ (параметр PGM) или каталогизированную процедуру для ее вызова, а также может содержать дополнительные параметры, необходимые для каждого конкретного случая.

Для вызова ПСО следует использовать либо имя IERRCO00, либо дополнительное имя SORT, а при использовании каталогизированных процедур — указывать процедуры SORT или SORTD.

Каталогизированная процедура SORTD применяется в случае, когда программы пользователя ЭВМ не используются или они предварительно обработаны Редактором связей. Процедура SORTD содержит следующие операторы:

```
//SORT      EXEC      PGM=IERRCO00,REGION=40K
//SYSOUT    DD        SYSOUT=A
//SORTLIB   DD        DSNAME=SYS1.SORTLIB,DISP=SHR
```

Каталогизированная процедура SORT применяется при использовании программ пользователя ЭВМ, требующих обработки Редактором связей. Процедура SORT содержит следующие операторы:

```
//SORT      EXEC      PGM=IERRCO00,REGION=98K
//SYSOUT    DD        SYSOUT=A
//SYSPRINT  DD        DUMMY
//SYSLMOD   DD        DSNAME=&GOSET,UNIT=SYSDA,SPACE=(3600,(20,20,1))
//SYSLIN    DD        DSNAME=&LOADSET,UNIT=SYSDA,SPACE=(80,(10,10))
//SORTLIB   DD        DSNAME=SYS1.SORTLIB,DISP=SHR
//SYSUT1    DD        DSNAME=&SYSUT1,SPACE=(1000,(60,20)),
//           UNIT=(SYSDA,SEP=(SORTLIB,SYSLMOD,SYSLIN))
```

Формат поля PARM оператора EXEC:

$$\text{PARM} = \left[\text{CORE} = \left\{ \begin{matrix} \times \times \times \times \times \times \\ \text{MAX} \end{matrix} \right\} \right], \text{MSG} = \left\{ \begin{matrix} \text{NO} \\ \text{CC} \\ \text{CP} \\ \text{AC} \\ \text{AP} \end{matrix} \right\} \left[\left\{ \begin{matrix} \text{BALN} \\ \text{OSCL} \\ \text{POLY} \\ \text{CRCX} \end{matrix} \right\} \right]$$

где CORE = $\times \times \times \times \times \times$ — указание в байтах значения размера основной памяти, которое на время выполнения задания используется вместо значения, определенного для ПСО при генерации операционной системы;

CORE=MAX — запрос максимально возможного размера основной памяти (5 Кбайт резервируется для операционной системы);

MSG — задание режима вывода сообщений;

NO — сообщение не выводится на печать;

CC — на консоль оператора ЭВМ выводятся только аварийные сообщения;

CP — на печатающее устройство выводится только аварийное сообщение;

AC — все сообщения выводятся на консоль оператора ЭВМ;

AP — все сообщения выводятся на печатающее устройство;

BALN, OSCL, POLY, CRCX — используются для определения соответственно балансового, осцилляторного, многофазного и перекрестного методов.

10.6. Динамический вызов программы СОРТИРОВКА-ОБЪЕДИНЕНИЕ

Программа ПСО может вызываться как подпрограмма с помощью макрокоманд **ATTACH, LINK** или **XCTL** в программах, написанных на языке Ассемблера. При использовании динамического вызова функции программы **СОРТИРОВКА-ОБЪЕДИНЕНИЕ** ограничиваются только выполнением сортировки. Ограничения имеются при выполнении операции сортировки: допускаются только два выхода из ПСО (**E15** и **E35**) для программ пользователя ЭВМ; при использовании макрокоманды **ATTACH** записи контрольных точек не создаются.

Вызывающая программа должна иметь все данные, которые для ПСО обычно содержатся в операторе **EXEC** и управляющих операторах ПСО. Данные передаются в виде списка параметров и списка адресов. Кроме того, для ПСО предусмотрена область сохранения регистров. Управляющие операторы ПСО определяются как литеральные константы. Операторы **DD** для ПСО должны быть во входном потоке в пункте задания, в котором динамически вызывается ПСО. В пункт задания необходимо включать операторы **DD** со следующими именами: **SORTLIB, SORTIN, SORTWK01—SORTWK32, SYSOUT, SORTOUT**.

В макрокомандах **ATTACH, LINK** или **XCTL** указываются имя точки входа ПСО (**SORT** или **IERRCO00**) и адрес списка параметров в программе. Перед вызовом ПСО адрес списка параметров загружается в регистр 1, а адрес области сохранения регистров — в регистр 13.

Список параметров состоит из одного полного слова. В старшем байте содержится шестнадцатеричное значение **X'80'**. Остальные три байта содержат адрес списка адресов для ПСО. Формат списка параметров и списка адресов:

Список параметров:

X'80'	Адрес списка адресов
-------	----------------------

Список адресов ↓ Адрес, на который указывает список параметров

		Длина списка адресов
Начальный адрес управляющего оператора SORT		
Конечный адрес управляющего оператора SORT		
Начальный адрес управляющего оператора RECORD		
Конечный адрес управляющего оператора RECORD		
Адрес программы пользователя ЭВМ для выхода E15 или нули		
Адрес программы пользователя ЭВМ для выхода E35 или нули		
Символы для замещения в именах операторов DD символов SORT		
Значение размера основной памяти для ПСО		
Метод сортировки		
X'FF'	Не используется	Режим вывода сообщений на печать

Длина списка адресов от 24 до 40 байт.

10.7. Коды возврата

После завершения работы ПСО возвращает операционной системе или вызывающей программе код возврата в регистре 15. Следующий пункт задания или вызывающая программа могут опрашивать этот код. Возможны следующие коды:

0 — успешное выполнение ПСО;

16 — выполнение ПСО прекращено из-за ошибок.

Программное обеспечение устройств машинной графики

11.1. Графический метод доступа

Графический метод доступа включает макрокоманды и стандартные программы для работы с алфавитно-цифровыми и графическими дисплеями ЕС ЭВМ. В ОС ЕС издания 4.1 обеспечивается работа алфавитно-цифрового дисплея ЕС-7906 и графического дисплея ЕС-7064. В издании 6.1 дополнительно обеспечиваются алфавитно-цифровые дисплеи ЕС-7061 и ЕС-7063, а также групповой комплекс графических дисплеев ЕС-7905.

11.1.1. Макрокоманда DCB

Эта макрокоманда строит блок управления данными (DCB) для одного или группы дисплеев, включая печатающее устройство для алфавитно-цифровых дисплеев. Формат макрокоманды:

```
[метка]   DCB   DSORG=GS;MACRF=код[,GTYPE={BASIC|EXPRESS}]
              [,DDNAME=имя-dd][,EXLST=адрес-списка]
              [,GNCP=число-блоков][,POLST=адрес-списка-опроса]
```

где MACRF=код — типы используемых макрокоманд: R — GREAD или GREADR; W — WRITE; C — GCNTRL. Допустимы восемь следующих комбинаций: R, W, RC, WC, (R, W), (RC, W), (R, WC), (RC, WC);

GTYPE=BASIC|EXPRESS — метод обработки сигналов внимания: BASIC — базисный, EXPRESS — специальный;

DDNAME=имя-dd — имя оператора DD, определяющего дисплей или группу дисплеев;

EXLST=адрес-списка — адрес списка программ выхода. По умолчанию равен нулю;

GNCP=число-блоков — число макрокоманд ввода-вывода (от 1 до 99), которые могут быть одновременно выданы для данного блока DCB (до макрокоманды WAIT). По умолчанию равно 1;

POLST=адрес-списка-опроса — адрес области, подготовленной программистом для специального метода обработки сигналов внимания. Первый элемент содержит адрес последнего элемента и

вначале должен быть установлен в 0. Последующие элементы содержат адреса блоков DCB. Таким образом, длина области должна равняться $(n+1)$ словам, где n — число блоков DCB.

Фиктивная секция, описывающая поля блока DCB для графических устройств, задается следующим образом:

DCBD DSORG=GS

11.1.2. Макрокоманда OPEN

Макрокоманда инициализирует один или несколько блоков. Формат макрокоманды:

[метка] OPEN (адрес-dcb[,адрес-dcb]...)
[MF={L|(E,имя-списка-параметров)|(E,(1))}]

где адрес dcb — адрес DCB;

имя-списка-параметров — адрес списка параметров, образованного макрокомандой OPEN в списковой форме;

(E,(1)) — адрес списка параметров, содержащийся в регистре 1.

11.1.3. Макрокоманда GREAD

Макрокоманда считывает данные из буфера дисплея. Формат макрокоманды:

[метка] GREAD адрес-dcb,тип,адрес-dcb, $\left\{ \begin{array}{l} \text{адрес-осбр} \\ \text{длина,адрес-области,} \\ \text{адрес-буфера} \\ \text{[длина],адрес-области} \\ \text{[устройство]} \end{array} \right\}$
[старт-адрес][MF={L|E}]

где адрес-dcb — адрес блока управления событием данных;
тип — тип операции считывания.

Алфавитно-цифровые дисплеи:

DSM — считать ручной ввод;

DSMW — считать ручной ввод, не снимая блокировки клавиатуры;

SMI — считать ручной ввод, не стирая стартового символа;

SMIW — считать ручной ввод, не стирая стартового символа и не снимая блокировки клавиатуры;

DSB — считать буфер индикатора;

DSBW — считать буфер индикатора, не снимая блокировки клавиатуры;

SMA — считать ручной ввод с полным адресом (только EC-7061, EC-7063);

SMAW — считать ручной ввод с полным адресом, не снимая блокировки клавиатуры (только EC-7061, EC-7063).

Полный адрес для типов SMA и SMAW передается перед данными и занимает 2 байта: первый байт — номер строки, второй — номер колонки начала сообщения.

Графические дисплеи:

BUF — считывание буфера. Перезапись прекращается и не возобновляется. Операнд «старт-адрес» не используется.

CUR — считывание до курсора. Передача заканчивается, если счетчик байтов канала равен 0 или если встречается курсор. В последнем случае вместо кода знака в позиции курсора передается X'1A', в DECB+16 записывается указатель неправильной длины (X'43'), в DECB+17 (3 байта) — выровненный вправо остаточный счетчик из слова состояния канала. Операнд «старт-адрес» не используется.

FND — нахождение курсора. Аналогично CUR, но данные в память не передаются. Кроме того, автоматически выдается команда **УТОЧНИТЬ СОСТОЯНИЕ**, которая передает 4 байта уточненного состояния в область, указываемую операндом «адрес-области».

STR — считывание и старт. То же, что и BUF, но перезапись возобновляется начиная с ячейки по адресу, указываемому операндом «старт-адрес».

адрес-dcd — адрес блока DCB для данного дисплея;

адрес-осбр (только для графических дисплеев) — адрес указателя блока управления выводом ОСБР (см. 11.5.6). Данные считываются из буфера начиная с ячейки, адрес которой содержится в поле BLP блока OACB, и помещаются в GDOA по адресу, указанному в поле CRSA блока OACB. Число передаваемых байтов данных равно разности между значениями полей OLP и CRSA блока OACB;

длина — число считываемых или (для типа FND) проверяемых байтов данных;

адрес-области — адрес области основной памяти, куда помещаются данные;

адрес-буфера (только для графических дисплеев) — адрес полуслова, содержащего буферный адрес считываемых данных;

устройство (только для алфавитно-цифровых дисплеев) — относительный адрес дисплея (от 1 до N) в группе, описываемой одним блоком DCB. Если это значение указывается в регистровой форме, то содержимое регистра при выполнении должно быть на единицу меньше;

старт-адрес (только для графических дисплеев) — адрес полуслова, содержащего буферный адрес приказа, с которого возобновляется перезапись. Если опущен для типа STR, то принимается равным значению «адрес-буфера» или поля BLP в OACB.

Коды возврата (десятичные):

04 — превышены пределы отведенного буфера или неправильный буферный адрес (для графических дисплеев);

08 — отрицательная или неправильная длина, полученная из OACB (для графических дисплеев);

16 — отсутствует адрес DCB;

20 — неправильный адрес DCB;

- 24 — отсутствует адрес ОСВР (для графических дисплеев);
- 28 — не указан тип операции;
- 32 — не указан адрес буфера (для графических дисплеев);
- 36 — не указан адрес области.

Индикация завершения для алфавитно-цифровых дисплеев:

DECВ+16=0, ЕСВ=7F	Нормальное завершение операции
DECВ+16=45, ЕСВ=41	Постоянная ошибка
DECВ+16=43, ЕСВ=7F	
DECВ+17=0	Запись большой длины
DECВ+17≠начальное значение счетчика CSW	Запись короткой длины
DECВ+17=начальное значение счетчика CSW	Отсутствие на экране стартового символа или символа конца сообщения (при чтении ручного ввода)

11.1.4. Макрокоманда GWRITE

Макрокоманда записывает данные в буфер дисплея или печатающего устройства комплекса алфавитно-цифровых дисплеев. Формат макрокоманды:

[метка] GWRITE адрес-decb,тип,адрес-dcb, $\left. \begin{array}{l} \text{адрес-осбр} \\ \text{длина,адрес-области,} \\ \text{адрес-буфера} \\ \text{[длина],адрес-области} \\ \text{[,устройство]} \end{array} \right\}$
 [,старт-адрес] [,MF={L|E}]

где адрес-decb — адрес блока управления событием данных;
 тип — тип операции записи.

Алфавитно-цифровые дисплеи:

DSB — писать в буфер индикатора или печатающего устройства;

DSBW — писать в буфер индикатора, не снимая блокировки клавиатуры;

EBW — стереть и писать в буфер индикатора;

EBWW — стереть и писать в буфер индикатора, не снимая блокировки клавиатуры;

LNE — писать по адресу строки;

LNEW — писать по адресу строки, не снимая блокировки клавиатуры;

EWL — стереть и писать по адресу строки;

EWLW — стереть и писать по адресу строки, не снимая блокировки клавиатуры;

LPA — писать по полному адресу в буфер дисплеев ЕС-7061, ЕС-7063;

LPAW — писать по полному адресу в буфер дисплеев ЕС-7061, ЕС-7063, не снимая блокировки клавиатуры;

DNP — стереть и писать в буфер дисплеев EC-7061, EC-7063, за исключением защищенного поля;

DNPW — стереть и писать в буфер дисплеев EC-7061, EC-7063, за исключением защищенного поля, не снимая блокировки клавиатуры.

Графические дисплеи:

BUF — писать в буфер индикатора. Перезапись прекращается и не возобновляется. Операнд «старт-адрес» не используется;

STR — писать и возобновить перезапись с буферной ячейки, адрес которой указывается содержимым поля «старт-адрес»;

адрес-dcb — адрес блока DCB для данного дисплея;

адрес-осбр (только для графических дисплеев) — адрес указателя блока управления выводом ОСБР (см. 11.5.6). Данные передаются в буфер по адресу, содержащемуся в поле BLP области ОАСВ. Определение начального адреса и числа байт данных производится с помощью полей SLOA и OLP области GDOA;

длина — число передаваемых байтов данных (включая знак новой строки и знак управления адресом строки для алфавитно-цифровых дисплеев);

адрес-области — адрес ячейки основной памяти, из которой начинают передаваться данные;

адрес-буфера (только для графических дисплеев) — адрес полуслова, содержащего буферный адрес, с которого начинают записываться данные;

устройство (только для алфавитно-цифровых дисплеев) — см. 11.1.3.

Коды возврата (десятичные):

Те же, что и для макрокоманды GREAD (см. 11.1.3).

Формат первого байта данных для операций с адресацией строки (для алфавитно-цифровых дисплеев):

Номер строки	Шестнадцатеричный код	Номер строки	Шестнадцатеричный код
	EC-7906		EC-7061, EC-7063
1	F0	7	F6
2	F1	8	F7
3	F2	9	F8
4	F3	10	F9
5	F4	11	FA
6	F5	12	FB
			EC-7061, EC-7063
			06
			07
			08
			09
			0A
			0B

Второй байт полного адреса для EC-7061, EC-7063 — это шестнадцатеричный адрес колонки (от 00 до 4F).

11.1.5. Макрокоманда GCNTRL

Макрокоманда используется для стирания буфера индикатора (алфавитно-цифровые дисплеи), для подачи звукового сигнала, остановки или запуска перезаписи изображения, включения нужных индикаторов на функциональной клавиатуре или для уста-

новки и удаления курсора (графические дисплеи). Формат макрокоманды:

[метка] GCNTRL адрес-decb,тип,адрес-dcb, $\left\{ \begin{array}{l} \text{адрес-данных} \\ \text{адрес-буфера} \\ \text{старт-адрес} \\ \text{[устройство]} \end{array} \right\} [MF = \{L|E\}]$

где адрес-decb — адрес блока управления событием данных;
тип — тип операции.

Алфавитно-цифровые дисплеи:

ERS — стереть буфер индикатора;

ERSW — стереть буфер индикатора, не снимая блокировки клавиатуры;

ENP — стереть буфер дисплеев ЕС-7061, ЕС-7063, за исключением защищенных полей;

ENPW — стереть буфер дисплеев ЕС-7061, ЕС-7063, за исключением защищенных полей, не снимая блокировки клавиатуры.

Графические дисплеи:

ALM — подача звукового сигнала. Четвертый операнд не нужен;

HLT — установка регистра адреса буфера и останов. Необходим операнд «адрес-буфера». Перезапись прекращается;

IND — установка индикаторов функциональной клавиатуры. Необходим операнд «адрес-данных»;

INS — установить курсор. Перезапись прекращается. Необходим операнд «адрес-буфера»;

RMV — удаление курсора. Перезапись прекращается. Необходим операнд «адрес-буфера»;

STR — установка регистра адреса буфера и пуск. Необходим операнд «старт-адрес»;

адрес-dcb — адрес блока DCB для данного дисплея;

адрес-данных (только для графических дисплеев) — адрес полного слова, содержащего маску установки индикаторов функциональной клавиатуры: 1 — включен, 0 — выключен;

адрес-буфера (только для графических дисплеев) — адрес полуслова, содержащего буферный адрес;

старт-адрес (только для графических дисплеев) — адрес полуслова, содержащего буферный адрес первого выполняемого приказа;

устройство (только для алфавитно-цифровых дисплеев) — см. 11.1.3.

Коды возврата (десятичные):

16 — отсутствует адрес DCB;

20 — неправильный адрес DCB;

24 — не указан адрес ОСВР (для графических дисплеев);

28 — не указан тип операции;

32 — не указан адрес буфера (для графических дисплеев);

36 — не указан адрес области.

11.1.6. Макрокоманда GREADR

Макрокоманда (только для графических дисплеев) идентифицирует источник сигнала внимания. Формат макрокоманды:

[метка] GREADR адрес-dcb, тип, адрес-dcb, адрес-области [MF={L|E}]

где адрес-dcb — адрес блока управления событием данных;
тип — тип операции:

MIP — считывание ручного ввода. В основную память считываются три байта. Первый байт содержит код запроса, указывающий функциональную клавиатуру (X'40') или клавишу КОНЕЦ (X'40') или Отмена (X'90'); второй байт — код клавиши функциональной клавиатуры; третий байт — идентификатор шаблона для функциональной клавиатуры. При нажатии клавиш ОТМЕНА или КОНЕЦ байты 2 и 3 содержат нули;

XYP — считывание регистров координат X и Y. Передаются 4 байта данных (по 2 для каждого регистра). Этой операции должна предшествовать макрокоманда GCNTRL с кодом типа HLT;

адрес-dcb — блок DCB для данного дисплея;

адрес-области — адрес области в основной памяти.

Коды возврата (десятичные):

Те же, что и для макрокоманды GCNTRL (см. п. 11.1.5).

11.1.7. Макрокоманда CLOSE

Макрокоманда отключает один или несколько блоков DCB. Формат макрокоманды:

[метка] CLOSE (адрес-dcb[, адрес-dcb]...) [MF={L|(E, имя-списка-параметров)|E,(1)}}]

где адрес-dcb — адрес закрываемого блока DCB;
имя-списка-параметров — см. 11.1.2.

11.1.8. Базисный метод обработки сигналов внимания

Макрокоманда SAEC. Создает блок управления сигналами внимания (GACB), указывающий программу обработки сигналов внимания. Формат макрокоманды:

[метка] SAEC [EP={адрес-точки-входа|0}] [DCB=адрес-dcb]
[COMAREA=адрес-области-связи]
[ATTNTYP={{(A|S|R), код}|NULL|SAVE|RESTORE}]
[PFKMSK={{(A|S|R), {целое-число|целое-число—
целое-число},...}|NULL|SAVE|RESTORE}]
[RESTART={YES|NO}] [MF={L|(E, адрес-gacb)}]

где EP=адрес-точки-входа — точка входа в программу обработки сигналов внимания. Если указано 0 (или операнд EP опущен), то все сигналы внимания будут помещаться в очередь к этой про-

грамме, пока она не будет активизирована макрокомандой ATTNTYP;

DCB=адрес-dcb — адрес блока DCB для данного дисплея;

COMAREA=адрес-области-связи — адрес области связи на границе полного слова длиной не менее 16 байт, которая служит для передачи информации о сигнале внимания программе обработки сигналов внимания;

ATTNTYP={ (A|S|R, код) |NULL|SAVE|RESTORE} — типы сигналов внимания (кроме сигналов от функциональной клавиатуры графических дисплеев), которые будут вызывать передачу управления программе обработки сигналов внимания: A, S, R — указывают, что последующая информация будет прибавляться, вычитаться или замещать существующее поле ATTNTYP в GACB; код — указывает типы сигналов внимания:

7066

7061 — от алфавитно-цифровых дисплеев

7063

LP — от светового пера

EOS — по концу последовательности приказов

AE — по асинхронной ошибке

END — от клавиши КОНЕЦ

} для графических дисплеев

NULL (только для графических дисплеев) — указывает, что никакие типы сигналов внимания не обслуживаются программой обработки внимания; SAVE и RESTORE — поле ATTNTYP в GACB должно соответственно запоминаться или замещаться маской, сохраненной предыдущей макрокомандой SAEC с операндом ATTNTYP=SAVE;

PFKMSK={ (A|S|R), {целое-число|целое-число — целое-число}, ...) |NULL|SAVE|RESTORE} (только для графических дисплеев) — клавиши функциональной клавиатуры (ФК), сигналы внимания от которых будут обслуживаться программой обработки сигналов внимания; A, S и R имеют то же назначение, что и для поля ATTNTYP (см. выше); целое-число — номер клавиши ФК (от 0 до 31); последовательность клавиш задается указанием первого и последнего номеров через дефис (например, 5-9); NULL, SAVE и RESTORE — имеют то же значение, что и для поля ATTNTYP (см. выше);

RESTART={YES|NO} (только для графических дисплеев): YES — указывает, что перезапись возобновляется после сигнала, вызванного обнаружением светового пера; NO — перезапись не возобновляется;

адрес-gacb — адрес ранее построенного блока GACB.

При получении управления программой обработки сигналов внимания регистр 1 содержит адрес GACB. Блок GACB, построенный макрокомандой SAEC, состоит из 14 слов и имеет следующий формат:

Адрес	Значение
+0	Адрес области связи COMAREA
+4	Адрес DCB
+8	PFKMSK, биты 0—31 соответствуют клавишам 0—31 (для алфавитно-цифровых дисплеев не используется)
+12	Маска ATTNTYP
+16	Адрес точки входа в программу обработки сигналов внимания

Позиции поля ATTNTYP (для графических дисплеев)

Биты 0—26	Резервируются
27	Асинхронная ошибка
29	Конец последовательности приказов
30	Световое перо
31	Клавиша КОНЕЦ

Формат области COMAREA

Адрес	Длина	Значение
+0	1	Идентификатор устройства (от 1 до 16), от которого пришел сигнал внимания. Если DCB описывает один индикатор, то значение этого поля равно 0 (для графических дисплеев это поле резервируется)
+1	1	Идентификатор шаблона
+2	1	Номер клавиши
+3	1	Тип сигнала: 01—клавиша КОНЕЦ 02—функциональная клавиатура 03—световое перо 04—конец последовательности приказов 06—асинхронная ошибка 07—сигнал от алфавитно-цифрового дисплея
+4	4	Данные уточненного состояния (для алфавитно-цифровых дисплеев это поле резервируется)
+8	2	X—координата } для алфавитно-цифровых диспле- Y—координата } ев эти поля резервируются
+12	4	Резервируется
+16		Область, определяемая пользователем

Макрокоманда SPAR. Включает программу обработки сигналов внимания. Формат макрокоманды:

[метка] SPAR (адрес-gacb[,адрес-gacb]...) [,PARTY=целое-число]
[,MF={L|(E, адрес-списка-gacb))}]

где адрес-gacb — адрес GACB, построенный макрокомандой SAEC;

PARTY=целое-число — приоритет (от 0 до 127) программы обработки сигналов внимания. По умолчанию принимается равным 0;
адрес-списка-gacb — адрес списка блоков GACB.

Коды возврата (десятичные):

- 04 — отсутствует адрес DCB;
- 08 — GACB включен ранее;
- 12 — нет основной памяти;

20 — макрокоманда SPAR выдана не в той задаче, в которой был открыт DCB.

Макрокоманда DAR. Выключает программу обработки сигналов внимания. Формат макрокоманды:

[метка] DAR (адрес-gacb [,адрес-gacb]...
[MF={L|CE, адрес-списка-gacb}}])

где адрес-gacb — адрес GACB, построенный макрокомандой SAEC;

адрес-списка-gacb — адрес списка блоков GACB.

Коды возврата (десятичные):

04 — отсутствует адрес DCB;

08 — программа не включена макрокомандой SPAR;

12 — программа активна;

20 — макрокоманда DAR выдана не в той задаче, которая выдала макрокоманду SPAR для данного GACB.

Макрокоманда ATTNINQ. Выбирает из очереди отдельный сигнал внимания, ожидает поступление сигнала внимания или очищает очередь. Формат макрокоманды:

[метка] ATTNINQ адрес-gacb[,MODE={R|W|CLEAR|(C,адрес-перехода),
TYP={ANY|код}}] [UNIT={целое-число|ALL}]
[MF={L|(E,адрес-списка-параметров}})]

где адрес-gacb — адрес блока GACB;

MODE — режим запроса:

R — режим отказа. Программа, выдавшая ATTNINQ, не получает управления, пока не появится сигнал внимания, тип которого указан в GACB;

W — режим ожидания. Программа находится в режиме ожидания до появления информации о сигнале внимания, тип которого указан в GACB. Допускается передача управления программе более высокого приоритета;

CLEAR — режим очистки. Все сигналы в очереди для данной программы обработки внимания гасятся;

C, адрес-перехода — условный режим. Выполняется переход по указанному адресу, если в очереди имеется сигнал, тип которого соответствует операнду TYP (графические дисплеи) или поступает от устройства, заданного в операнде UNIT (алфавитно-цифровые дисплеи);

TYP — используется только при условном режиме для графических дисплеев. ANY — определяет любой тип сигналов.

Код может быть END, EOS, LP, AE (см. описание макрокоманды SAEC) или PFKnn (где nn указывает номер клавиши ФК) и не всегда совпадает с типом, указанным в GACB;

UNIT={целое-число|ALL} (только алфавитно-цифровые дисплеи) — целое десятичное число в диапазоне от 1 до 16, указывающее устройство, от которого ожидается сигнал внимания, если одному блоку DCB соответствует несколько индикаторов. ALL задает прием сигналов от всех устройств:

адрес-списка-параметров — адрес списка, построенного другой макрокомандой ANALYZ,

(п) — номер регистра, содержащего адрес списка параметров.

Область выходных данных готовится пользователем, имеет длину 8 байт и следующий формат:

Смещение	Длина	Значение
0	4	Данные уточненного состояния для графического дисплея. Для алфавитно-цифровых дисплеев эта область резервируется.
4	1	Индекс устройства (для алфавитно-цифровых дисплеев)
5	3	Адрес DCB

Коды возврата (десятичные):

04 — сигналов внимания нет;

08 — сигнал внимания есть, но нет программы обработки этого сигнала;

12 — отсутствует операнд «адрес-списка-опроса» или «список-программ-внимания»;

16 — отсутствует адрес выходных данных.

Макрокоманда GSERV. Восстанавливает GCB. Формат макрокоманды:

[метка] GSERV {адрес-элемента-списка, бит-gcb|(1)}

где адрес-элемента-списка — адрес 4-байтного поля в списке опроса, которое имеет формат, описанный выше;

бит-gcb — код для сброса битов GCB:

1 — сбрасывается разряд 7 (ввод с клавиатуры);

2 — сбрасывается разряд 6 (световое перо, асинхронная ошибка или конец последовательности приказов);

3 — сбрасываются разряды 6 и 7.

Для алфавитно-цифровых дисплеев можно указывать только код 1. Если указывается регистр, то в старший байт регистра нужно загрузить код, а в младшие три байта — адрес элемента списка.

11.1.10. Управление буфером для графических дисплеев

Макрокоманда ASGNBFR. Выделяет буфер. Формат макрокоманды:

[метка] ASGNBFR адрес-dcb, адрес-счетчика-байтов
[,MF={L|(E,адрес-списка-параметров)}]

где адрес-dcb — адрес блока DCB для данного дисплея;

адрес-счетчика-байтов — адрес полуслова, в котором содержится размер запроса в байтах. Если этот размер не кратен 256, то берется ближайшее большее число, кратное 256;

адрес-списка-параметров — адрес списка, построенного другой макрокомандой ASGNBFR.

16 — отведенная для таблицы память мала;
20 — неправильный адрес таблицы или код запроса в списке параметров.

11.2. Пакет графических подпрограмм для языков Фортран, Кобол и ПЛ/1

11.2.1. Общие сведения

Подпрограммы пакета графических подпрограмм (ПГП) создают изображение на графических дисплеях ЕС-7064 и ЕС-7905 и вызываются из программ, написанных на языке высокого уровня (Фортран, Кобол или ПЛ/1) с помощью оператора CALL.

Существуют следующие ограничения при вызове подпрограмм ПГП:

функции определения состояния (ITPC, ITBP, PTBP, ITST) не используются для запроса информации о выполнении подпрограмм ПГП (Кобол, ПЛ/1);

входные данные для подпрограмм должны быть двоичными числами с фиксированной или плавающей точкой длиной в полное слово (Кобол, ПЛ/1);

аргументы в операторе CALL могут быть заданы для Кобола в виде констант, целых констант, действительных констант, переменных, целых переменных, действительных переменных, имен массивов; для ПЛ/1 — в виде скалярных констант, переменных или выражений;

для ПЛ/1, если используется структура, нужно указывать составное имя ее элемента; строки знаков могут использоваться только в качестве аргумента «текст» подпрограмм PTEXT и PLSTR и аргумента «область» подпрограммы GSPRD, при этом аргумент «счетчик» должен быть равным текущей длине строки;

для ПЛ/1 все массивы, указанные в аргументах, должны быть индексированы.

Подпрограммы ПГП присоединяются Редактором связей:

для Фортрана — с помощью оператора

```
INCLUDE SYSLIB (IHDGSP03)
```

для Кобола — с помощью операторов

```
INCLUDE SYSLIB (IHDGSP01)
```

```
INCLUDE SYSLIB (IHDGSP02)
```

```
INCLUDE SYSLIB (IHDGSP03)
```

для ПЛ/1 — с помощью операторов

```
INCLUDE SYSLIB (IHEGSP01)
```

```
INCLUDE SYSLIB (IHEGSP02)
```

```
INCLUDE SYSLIB (IHEGSP03)
```

11.2.2. Общие аргументы подпрограмм ПГП

Нгд — целая переменная, идентифицирующая набор графических данных.

Номер — константа или переменная, идентифицирующая элемент, последовательность или буферную подпрограмму. Задается программистом.

Ключ — целая переменная, идентифицирующая элемент, последовательность или буферную подпрограмму. Задается ПГП.

Выключатель — целая константа или переменная, задающая действие:

- 1 — включить данный элемент;
- 2 — выключить данный элемент;
- 3 — подставить данный элемент вместо ранее созданного.

11.2.3. Подпрограммы инициализации и завершения

Подпрограмма INGSP. Инициализирует ПГП. Формат:

CALL INGSP (пгп, нуль-переменная)

где пгп — целая переменная, значение которой присваивает ПГП, чтобы идентифицировать данное обращение;

нуль-переменная — целая переменная, используемая вместо опущенных аргументов или определяющая условие аварийного завершения. Задается программистом. Если абсолютное значение нуль-переменной больше, чем код возврата, то выполнение продолжается. В противном случае происходит аварийное завершение программы, причем выдается дамп, если значение нуль-переменной отрицательное.

В графической программе первой всегда должна вызываться подпрограмма INGSP.

Подпрограмма INDEV. Инициализирует дисплей. Формат:

CALL INDEV (пгп, индекс, устройство [буфер][режим])

где пгп — целая переменная, значение которой должно совпадать с аргументом пгп при вызове подпрограммы INGSP;

индекс — целая константа или переменная, значением которой является двузначное десятичное число, большее семи (XX), входящее в имя оператора DD(FTXXF001), описывающего дисплей;

устройство — целая переменная (см. 11.2.2), значение которой присваивает подпрограмма INDEV;

буфер — целая константа или переменная, задающая длину в байтах (128 или число, кратное 256) области вывода графических данных. По умолчанию предполагается 256 байт;

режим (только для EC-7905) — целая константа или переменная, указывающая, что яркость воспроизведения графических и текстовых элементов будет изменяться в соответствии с параметром «яркость» в подпрограммах PLINE, PPNT, PSGMT и PTEXT:

- 1 — все элементы будут воспроизводиться с одинаковой яркостью;
- 2 — элементы могут воспроизводиться с обычной или повышенной яркостью.

Подпрограмма INGDS. Инициализирует набор графических данных. Формат:

CALL INGDS (устройство, нгд[,буфер][,идентификация]
[нгд1, ...нгд 49])

где устройство — см. выше;

нгд — целая переменная (см. 11.2.2), значение которой присваивает подпрограмма INGDS;

буфер — целая константа или переменная, задающая длину (в байтах) области вывода: если опущен, то принимается равным значению при обращении к INDEV;

идентификация — целая константа или переменная, задающая способ идентификации: 1 — стандартный, 2 — пользовательский;

нгд1 — нгд49 — целые переменные для эквивалентных наборов графических данных.

Подпрограмма SPEC. Указывает состояние программы. Формат:

CALL SPEC (пгп, код,подпрограмма [,подпрограмма]...)

где пгп — см. подпрограмму INGSP. Значение этого аргумента должно совпадать со значением, присвоенным подпрограммой INGSP;

код — целая константа или переменная со следующими значениями: 1 — загружаемая подпрограмма, 2 — вызываемая подпрограмма;

подпрограмма — целая константа или переменная, идентифицирующая подпрограмму в соответствии со следующей таблицей:

Имя	Номер	Имя	Номер	Имя	Номер	Имя	Номер	Имя	Номер
INDEV	1	SALRM	16	PTEXT*	29	FSMOD	42	RTBP*	55
TMDEV	2	GSPRD*	17	STPOS	30	STEOS	43	ITST*	56
INGDS	3	RCURS*	18	MVPOS	31	ORGDS	44	ENBIN*	63
TMGDS	4	ICURS*	19	BGSEQ	32	LOCNP	45	MODIN*	64
GRATL	5	SDATM	20	BGSUB	33	BGTRK	46	DSBIN	65
ENATL	6	SGRAM	21	ENSEQ	34	RDTRK	47	RDINP*	66
ENATN	7	SDATL	22	ENSUB	35	ENTRK	48		
DSATN	8	SGDSL	23	LKSUB	36	DFSTR	49		
MPATL	9	SSCIS	24	INCL	37	PLSTR	50		
MLPEO	10	SCHAM	25	OMIT	38	ORGEN	51		
SLPAT	11	PLINE*	26	EXEC*	39	NVRT	52		
MLITS	12	PPNT*	27	RESET	40	ITRC*	53		
RQATN*	13	PSGMT*	28	IDPOS	41	ITBP*	54		

Примечания: 1. Звездочка означает, что данная подпрограмма по умолчанию является загружаемой.

2. Номера 14 и 15 резервные.

Парой аргументов можно задать серию подпрограмм, если второй аргумент сделать отрицательным (например, 12—15 задает подпрограммы 12, 13, 14 и 15).

Подпрограмма TMGDS. Завершает использование набора графических данных. Формат:

CALL TMGDS (нгд)

где нгд — см. 11.2.2.

Подпрограмма TMDEV. Завершает использование устройства. Формат:

CALL TMDEV (устройство)

где устройство — см. выше.

Подпрограмма TMGSP. Завершает использование ПГП. Формат:

CALL TMGSP (пгп)

где пгп — см. подпрограмму INGPS.

11.2.4. Подпрограммы настройки

Подпрограмма SDATM. Устанавливает режим данных. Формат:

CALL SDATM (нгд, х-режим [,у-режим])

где нгд — см. 11.2.2;

х-, у-режим — целые константы или переменные, задающие тип и форму входных данных для координат X и Y:

- 1 — вещественные в абсолютной форме;
- 2 — вещественные в форме приращений;
- 3 — целые в абсолютной форме;
- 4 — целые в форме приращений.

Если у-режим опущен, то для него принимается значение х-режима.

Подпрограмма SGRAM. Устанавливает графический режим. Формат:

CALL SGRAM (нгд, выход)

где нгд — см. 11.2.2;

выход — целая константа или переменная, задающая тип выходных данных:

- 1 — оптимизированная форма;
- 2 — абсолютная форма;
- 3 — форма приращений.

Подпрограмма SCHAM. Устанавливает режим знаков. Формат:

CALL SCHAM (нгд, знак)

где нгд — см. 11.2.2;

знак — целая константа или переменная, задающая тип воспроизводимых знаков:

- 1 — защищенные знаки основного размера;
- 2 — защищенные знаки увеличенного размера;
- 3 — незащищенные знаки основного размера;
- 4 — незащищенные знаки увеличенного размера.

Подпрограмма SGDSL. Устанавливает границы кадра. Формат:

CALL SGDSL (нгд, кх1, жу1, кх2, ку2[, эх1, эу1, эх2, эу2])

где нгд — см. 11.2.2;

кх1, ку1 — константы или переменные, задающие координаты нижнего левого угла кадра;
кх2, ку2 — константы или переменные, задающие координаты правого верхнего угла кадра;
эх1, эу1 — константы или переменные, задающие координаты нижнего левого угла экрана;
эх2, эу2 — константы или переменные, задающие координаты правого верхнего угла экрана.

Подпрограмма SDATL. Устанавливает границы данных. Формат:

CALL SDATL (нгд,вкх1,вку1,вкх2,вку2)

где нгд — см. 11.2.2;

вкх1, вку1 — константы или переменные, задающие координаты нижнего левого угла области входных данных;

вкх2, вку2 — константы или переменные, задающие координаты верхнего правого угла области входных данных.

Подпрограмма SSCIS. Устанавливает режим усечения. Формат:

CALL SSCIS (нгд,усечение)

где нгд — см. 11.2.2;

усечение — целая константа или переменная, задающая режим усечения:

- +1 — усечение на границах экрана, порождение образа продолжается;
- −1 — усечение на границах экрана, порождение образа прекращается;
- +2 — усечение на границах кадра, порождение образа продолжается;
- −2 — усечение на границах кадра, порождение образа прекращается;
- +3 — усечение не производится.

11.2.5. Подпрограммы порождения образов

Подпрограмма MVPOS. Устанавливает луч в позицию. Формат:

CALL MVPOS (нгд,х,у,[номер],[ключ],[выключатель])

где нгд, номер, ключ и выключатель — см. 11.2.2;

х, у — константы или переменные, задающие координаты X и Y луча.

Подпрограмма STPOS. Устанавливает луч в абсолютную позицию. Формат:

CALL STPOS (нгд,х,у,[номер],[ключ],[выключатель])

где нгд, номер, ключ, выключатель — см. 11.2.2;

х, у — см. выше. Координаты должны быть в абсолютной форме.

Подпрограмма PLINE. Вычерчивает линию. Формат:

CALL PLINE (нгд,х,у,[номер],[ключ],[выключатель],[число]
,[индх],[инду],[Δх],[Δу],[яркость])

где нгд, номер, ключ, выключатель — см. 11.2.2;

х, у — константы, переменные или имена массивов, задающие координаты конечной (конечных) точки (точек) линии (линий);
число — целая константа или переменная, задающая число линий;

индх, инду — целые константы или переменные, индексирующие массивы аргументов X и Y;

Δх, Δу — константы или переменные, задающие приращения к координатам конечных точек;

яркость (только для ЕС-7905) — целая константа или переменная, задающая яркость воспроизведения элемента;

1 — обычная яркость (это значение принимается по умолчанию);

2 — повышенная яркость.

Подпрограмма PPNT. Вычерчивает точку. Формат:

CALL PPNT (нгд,х,у,[номер],[ключ],[выключатель],[число]
,[индх],[инду],[Δх],[Δу],[яркость])

где аргументы — см. выше.

Подпрограмма PSGMT. Вычерчивает отрезок. Формат:

CALL PSGMT (нгд,хн,ун,хк,ук,[номер],[ключ],[выключатель]
,[число],[индхн],[индун],[индхк],[индук],[Δхн],
[Δун],[Δхк],[Δук],[яркость])

где нгд, номер, ключ, выключатель — см. 11.2.2;

хн, ун, хк, ук — константы, переменные или имена массивов, задающие координаты начальных и конечных точек отрезков;

число — целая константа или переменная, задающая число отрезков;

индхн, индун, индхк, индук — целые константы или переменные, индексирующие массивы аргументов хн, ун, хк, ук;

Δхн, Δун, Δхк, Δук — константы или переменные, задающие приращения координат начальных и конечных точек отрезков;

яркость — см. выше.

Подпрограмма PTEXT. Выводит текст. Формат:

CALL PTEXT (нгд,текст,счетчик,[номер],[ключ],[выключатель]
[х],[у],[яркость])

где нгд, номер, ключ, выключатель — см. 11.2.2;

текст — переменная или имя массива, задающие начало текста;
счетчик — целая константа или переменная, задающая число знаков в тексте;

х, у — константы или переменные, задающие координаты центра первого знака текста;

яркость — см. выше.

Подпрограмма STEOS. Образует приказ КОНЕЦ ПОСЛЕДОВАТЕЛЬНОСТИ ПРИКАЗОВ. Формат:

CALL STEOS (нгд,[номер],[ключ],[выключатель])

где нгд, номер, ключ, выключатель — см. 11.2.2.

11.2.6. Подпрограммы группировки элементов

Подпрограмма BGSEQ. Начинает последовательность элементов. Формат:

CALL BGSEQ (нгд,[номер],[ключ],[выключатель])

где нгд, номер, выключатель — см. 11.2.2;

ключ — целая переменная, идентифицирующая данную последовательность, присваивается подпрограммой BGSEQ.

Подпрограмма ENSEQ. Заканчивает последовательность элементов. Формат:

CALL ENSEQ (нгд, ключ)

где нгд, ключ — см. 11.2.2.

11.2.7. Буферные подпрограммы

Подпрограмма BGSUB. Начинает буферную подпрограмму. Формат:

CALL BGSUB (нгд,[номер],[ключ],[выключатель])

где нгд, выключатель — см. 11.2.2;

номер — константа или переменная, идентифицирующая буферную подпрограмму. Задается программистом;

ключ — целая переменная, идентифицирующая буферную подпрограмму. Задается подпрограммой BGSUB. Используется только при вызове подпрограммы RESET.

Подпрограмма ENSUB. Заканчивает буферную подпрограмму. Формат:

CALL ENSUB (нгд [,ключ])

где нгд — см. 11.2.2;

ключ — целая переменная, идентифицирующая созданную буферную подпрограмму. Задается подпрограммой ENSUB.

Подпрограмма LKSUB. Вызывает буферную подпрограмму. Формат:

CALL LKSUB (нгд,[бп-номер],[бп-ключ],[св-номер],[св-ключ]
[выключатель])

где нгд, выключатель — см. 11.2.2;

бп-номер — константа или переменная, идентифицирующая вызываемую буферную подпрограмму;

бп-ключ — целая переменная, идентифицирующая вызываемую буферную подпрограмму;

св-номер — константа или переменная, идентифицирующая вызов буферной подпрограммы. Задается программистом;

св-ключ — целая переменная, идентифицирующая вызов буферной подпрограммы. Задается подпрограммой LKSUB.

11.2.8. Подпрограммы управления образами

Подпрограмма RESET. Удаляет образы. Формат:

CALL RESET(нгд[,номер][,ключ])

где нгд, номер, ключ — см. 11.2.2.

Подпрограмма IDPOS. Указывает позицию луча. Формат:

CALL IDPOS (нгд,х,у[,mx][,my])

где нгд — см. 11.2.2;

х, у — константы или переменные, задающие абсолютную позицию, от которой должны отсчитываться координаты X и Y для новых элементов;

mx, my — константы или переменные, задающие координаты текущего положения луча.

11.2.9. Управление воспроизведением образов

Подпрограмма EXEC. Выполняет последовательность приказов. Формат:

CALL EXEC (нгд)

где нгд — см. 11.2.2.

Подпрограмма INCL. Переводит во включенное состояние. Формат:

CALL INCL (нгд[,номер][,ключ])

где нгд, номер, ключ — см. 11.2.2.

Подпрограмма OMIT. Переводит в выключенное состояние. Формат:

CALL OMIT (нгд[,номер][,ключ])

где нгд, номер, ключ — см. 11.2.2.

Подпрограмма ORGDS. Упорядочивает наборы графических данных. Формат:

CALL ORGDS (нгд1[,нгд2]...)

где нгд1 — нгдп — целые переменные, задающие наборы графических данных.

11.2.10. Подпрограмма ввода с клавиатуры и анализа буферных данных

Подпрограмма ICURS. Вводит курсор. Формат:

CALL ICURS (нгд [,номер], [ключ] [,позиция])

где нгд, номер, ключ — см. 11.2.2;

позиция — целая константа или переменная, задающая позицию знака в текстовом элементе. Позицией первого знака является единица.

Подпрограмма RCURS. Удаляет курсор. Формат:

CALL RCURS (нгд)

где нгд — см. 11.2.2.

Подпрограмма GSPRD. Считывает данные. Формат:

CALL GSPRD (нгд ,область,счетчик,тип,[код],[номер1],
[ключ1],[номер2],[ключ2])

где нгд — см. 11.2.2;

область — переменная или имя массива, куда будут считываться данные;

счетчик — целая константа или переменная, задающая объем считываемых данных. Если счетчик положительный, то считывание заканчивается после того, как считан заданный объем данных или если встретился конец набора графических данных или конец последнего элемента. Если счетчик задан со знаком минус, то считывание заканчивается при тех же условиях, что и при положительном счетчике, и, кроме того, если обнаружен курсор;

тип — целая константа или переменная, задающая тип считываемых данных: 1 — считывается текст; 2 — должны считываться приказы и данные;

код — целая переменная, указывающая причину окончания считывания. Устанавливается подпрограммой GSPRD и имеет следующий смысл:

0 — считаны все данные, заданные аргументом счетчик;

+п — обнаружен конец нгд или элемента, п — число считанных байтов;

—п — встретился курсор, п — число считанных данных;

номер1 — константа или переменная, задающая номер первого считываемого элемента;

ключ1 — целая переменная, задающая ключ первого считываемого элемента;

номер2 — константа или переменная, задающая номер последнего считываемого элемента;

ключ2 — целая переменная, задающая ключ последнего считываемого элемента.

11.2.11. Подпрограммы обработки запросов

Подпрограмма CRATL. Создает уровень запросов. Формат:

CALL CRATL (устройство, уровень, очередь)

где устройство — см. 11.2.3;

уровень — целая переменная, идентифицирующая уровень запросов. Устанавливается подпрограммой CRATL;

очередь — целая переменная, указывающая, что делать с информацией о запросах после того, как она представлена пользователю подпрограммой RQATN: 1 — удалить информацию из очереди после ее использования; 2 — удалить информацию из очереди после ее использования, если уровень для этой очереди в данный момент активен.

Подпрограмма ENATL. Удаляет уровень запросов. Формат:

CALL ENATL (уровень[,условие])

где уровень — целая переменная (см. выше);

условие — целая константа или переменная, задающая следующие действия: 1 — удалить данный уровень и все предыдущие, а следующий уровень сделать активным; 2 — удалить все уровни в стеке до заданного, который сделать активным. По умолчанию принимается равным единице.

Подпрограмма ENATN. Разрешает запросы. Формат:

CALL ENATN(уровень,запрос,запрос...)

где уровень — см. выше.

запрос — целая константа или переменная, задающая тип запроса:

0—31 — клавиши функциональной клавиатуры;

32 — клавиша КОНЕЦ;

34 — световое перо;

35 — приказ КОНЕЦ ПОСЛЕДОВАТЕЛЬНОСТИ ПРИКАЗОВ.

Можно задать промежуток, указав пару аргументов «запрос», причем первый задает начало промежутка, а второй со знаком минус — конец промежутка.

Подпрограмма DSATN. Запрашивает запросы. Формат:

CALL DSATN(уровень,запрос,запрос...)

где уровень — см. выше;

запрос — см. выше.

Подпрограмма SLPAT. Разрешает запросы от светового пера. Формат:

CALL SLPAT(нгд,функция)

где нгд — см. 11.2.2;

функция — целая константа или переменная со следующими значениями: 1 — разрешить запрос от светового пера для данного набора графических данных; 2 — запретить запрос от светового пера для данного набора графических данных.

Подпрограмма RQATN. Получает информацию о запросах. Формат:

CALL RQATN(уровень,тип,функция,информация,запрос,запрос...)

где уровень — см. выше;

тип — целая переменная, идентифицирующая тип запроса. Устанавливается подпрограммой RQATN. Имеет следующие значения:

1 — запрос от клавиши 0 функциональной клавиатуры;

0 — нет запросов данного типа;

1 — 31 — запрос от одной из клавиш функциональной клавиатуры;

32 — запрос от клавиши КОНЕЦ;

34 — запрос от светового пера;

35 — запрос от приказа КОНЕЦ ПОСЛЕДОВАТЕЛЬНОСТИ ПРИКАЗОВ;

функция — целая константа или переменная, определяющая режим работы подпрограммы RQATN: 1 — выбрать из очереди первую информацию о запросах; если ее нет, то вернуться к пользовательской программе; 2 — выбрать из очереди первую информацию о запросах; если ее нет, то ждать получения запроса;

информация — имя массива, идентифицирующее массив размером в 10 слов для дополнительной информации о запросах. Формат массива:

Слово	Значение
1	Идентификатор нгд, содержащего обнаруженный световым пером образ. Идентификатор шаблона (для запросов от функциональной клавиатуры, остальные слова не используются)
2	Значение ключа элемента, образ которого обнаружен световым пером, или нуль, если образ не был идентифицирован
3	Число, обозначающее положение знака в строке, если световое перо указывает на этот знак. Число байт от начала элемента, обнаруженного световым пером, до графического приказа, породившего обнаруженную часть образа, если обнаружение произошло не на знаке и элемент идентифицирован.
4	Число байт от начала набора графических данных до графического приказа, породившего обнаруженную часть образа, если элемент не был идентифицирован
5	Номер обнаруженного элемента Нуль, если элемент не идентифицирован
6	Ключ последовательности, связанной с обнаруженным образом. Нуль, если обнаруженный образ не связан с последовательностью
7	Число байтов от начала последовательности до графического приказа, породившего обнаруженную часть образа. Нуль, если в слове 5 содержится нуль
8 и 9	Номер последовательности, связанной с обнаруженным образом. Нуль, если последовательность не идентифицирована номером или если обнаруженный образ не связан с последовательностью
10	Абсолютные координаты (X и Y) конечной точки обнаруженного отрезка, точки или знака. Нуль, если информация о координатах не запрашивалась Код знака (ДКОИ), на который указывает световое перо. Код записывается в старший байт слова, а младшие заполняются нулями. Нуль, если световое перо не указывает на знак или если такая информация не запрашивалась

запрос — целая константа или переменная, идентифицирующая запрос (см. подпрограмму ENATN).

Подпрограмма MLPEO. Модифицирует информацию о запросах. Формат:

CALL MLPEO(уровень,тип,[информация][,рестарт])

где уровень — см. подпрограмму CRATL;

тип — целая переменная или константа, указывающая тип запроса:

1 — запросы от приказа **КОНЕЦ ПОСЛЕДОВАТЕЛЬНОСТИ ПРИКАЗОВ**;

2 — запросы от светового пера;

информация — целая константа или переменная, задающая вид дополнительной информации: 1 — не запрашивать координаты и коды знаков; 2 — формировать координаты обнаружения или положения луча в момент выполнения приказа **КОНЕЦ ПОСЛЕДОВАТЕЛЬНОСТИ ПРИКАЗОВ**; 3 — формировать код обнаруженного знака; 4 — формировать код и координаты обнаруженного знака. По умолчанию принимается значение 1;

рестарт — целая константа или переменная, задающая режим регенерации: 1 — возобновление регенерации; 2 — прекращение регенерации. По умолчанию принимается значение 1.

Подпрограмма MLITS. Включает световую индикацию функциональной клавиатуры. Формат:

CALL MLITS({устройство|уровень}, состояние[, клавиша...])

где устройство — см. 11.2.3;

уровень — см. подпрограмму CRATL;

состояние — целая константа или переменная, задающая функцию: 1 — не изменять световую индикацию. При этом значении необходимо использовать аргумент «уровень»; 2 — выключить подсветку всех клавиш; 3 — включить подсветку клавиш, от которых разрешены запросы для данного уровня; 4 — включить подсветку клавиш, заданных аргументом «клавиша»;

клавиша — целая константа или переменная со значениями от 0 до 31, задающая подсвечиваемые клавиши. Можно указывать диапазон клавиш парой аргументов, причем первый должен указывать номер первой клавиши диапазона, а второй должен быть отрицательным и указывать номер последней клавиши.

Подпрограмма MPATL. Модифицирует стек уровней запросов. Формат:

CALL MPATL(уровень, смещение[, отсчет])

где уровень — см. подпрограмму CRATL;

смещение — целая константа или переменная, определяющая направление перемещения и число уровней, на которое нужно сдвинуть заданный уровень: $+n$ — сдвиг вверх на n уровней относительно уровня, заданного аргументом «отсчет»; $-n$ — сдвиг вниз;

отсчет — целая переменная, идентифицирующая уровень запросов, относительно которого осуществляется сдвиг. По умолчанию принимается текущий уровень.

Подпрограмма SALRM. Выдает звуковой сигнал. Формат:

CALL SALRM(устройство)

где устройство — см. 11.2.3.

11.2.12. Подпрограммы для светового пера

Подпрограмма LOCPN. Определяет позицию светового пера. Формат:

CALL LOCPN(нгд, х, у)

где нгд — см. 11.2.2;

х, у — целые или вещественные переменные, представляющие позицию светового пера на экране. Устанавливаются подпрограммой LOCPN.

Подпрограмма BGTRK. Начинает трассировку. Формат:

CALL BGTRK(нгд,х,у)

где нгд — см. 11.2.2;

х, у — константы или переменные, задающие абсолютные координаты позиции на экране, с которой начинается трассировка.

Подпрограмма RDTRK. Считывает координаты символа трассировки. Формат:

CALL RDTRK(нгд,х,у)

где нгд — см. 11.2.2,

Подпрограмма ENTRK. Заканчивает трассировку. Формат:

CALL ENTRK(нгд)

где нгд — см. 11.2.2.

Подпрограмма RDTRK. Считывает координаты символа трассировки. Формат:

CALL RDTRK(нгд,х,у)

где нгд — см. 11.2.2;

х, у — переменные, определяющие координаты центра символа трассировки в абсолютной форме. Устанавливаются подпрограммой RDTRK.

11.2.13. Подпрограммы для планшета (только для ЕС-7905)

Подпрограмма ENBIN. Разрешает ввод данных с планшета. Формат:

CALL ENBIN(нгд[число][переключатель])

где нгд — см. 11.2.2;

число — целая константа или переменная (от 1 до 1000), задающая максимальное количество точек, которые могут быть введены с планшета; по умолчанию принимается равным 30;

переключатель — целая константа или переменная, задающая режим высвечивания данных, введенных с планшета:

1 — данные высвечиваются в виде связанных отрезков, соединяющих введенную точку с предыдущей. Это значение принимается по умолчанию;

2 — введенные данные высвечиваются в виде точек;

3 — введенные данные не высвечиваются.

Подпрограмма DSBIN. Запрещает ввод данных с планшета. Формат:

CALL DSBIN(нгд)

где нгд — см. 11.2.2.

Подпрограмма MODIN. Модифицирует введенные данные. Формат:

CALL MODIN(нгд[число])

где нгд — см. 11.2.2;

число — целая константа или переменная, указывающая количество удаляемых точек; удаляются точки, введенные последними, по умолчанию принимается равным 1.

Подпрограмма RDINP. Считывает введенные данные. Формат:

CALL RDINP (нгд, область, счетчик [, код])

где нгд — см. 11.2.2;

область — переменная или имя массива, куда будут помещаться координаты введенных точек (4 байта — координата X, 4 байта — координата Y), точки считываются последовательно, начиная с первой введенной точки;

счетчик — целая константа или переменная, задающая количество считываемых точек;

код — целая переменная, значение которой присваивается подпрограммой RDINP:

0 — число введенных точек равно счетчику;

+n — число введенных точек больше счетчика на величину n;

-n — число введенных точек меньше счетчика на величину n.

11.2.14. Отладочные средства (только для языка Фортран)

Коды возврата:

Код	Значение	Дополнительная информация
0	Подпрограмма выполнена успешно	Нет
1	Имело место усечение	Счетчик входных данных, которые были усечены последними: 1—для STPOS и NVPOS 0—если не было порождено данных То же, что и для кода 1
2	Ошибка масштабирования	1—замещающий элемент больше замещаемого
3	Не хватило памяти	2—нет места в 128-байтном нгд 3—для нгд не хватило буферной памяти в устройстве 4—не хватило основной памяти
4	Ошибка в аргументах	0—ошибка не может быть отнесена к конкретному аргументу 1—n—номер первого ошибочного аргумента
5	Ошибка ввода-вывода	Нет

Функция ITRC. Определяет код возврата. Формат:

ITRC (пгп, код)

где пгп — см. 11.2.3;

код — целая константа или переменная, задающая нижнюю границу проверяемых кодов (1—5) или требование дополнительной информации (9). Функция ITRC получает значение 0, если код возврата меньше заданного кода, и равна значению кода возврата в противном случае или значению дополнительной информации, если код в аргументе равен 9.

Функция ITBP. Определяет целочисленную позицию луча. Формат:

ITBP (нгд, координата)

где нгд — см. 11.2.2;

координата — целая константа: 1 — предполагаемая координата X; 2 — истинная координата X; 3 — предполагаемая координата Y; 4 — истинная координата Y.

Функция RTBP. Определяет вещественную позицию луча. Формат:

RTBP(нгд,координаты)

где нгд, координата — см. выше.

Функция ITST. Определяет состояние. Формат:

ITST(нгд,данные)

где нгд — см. 11.2.2;

данные — целая константа, задающая группу данных: 1 — входные данные координаты X; 2 — входные данные координаты Y; 3 — выходные данные; 4 — текстовые данные.

Значения функции ITST:

Группа	Значение	Смысл
Входные данные	1	Вещественные, абсолютные
	2	Вещественные, приращения
	3	Целые, абсолютные
	4	Целые, приращения
Выходные данные	1	Оптимизированная форма
	2	Абсолютная форма
	3	Приращения
Знаковые данные	1	Защищенные, основного размера
	2	Защищенные, увеличенного размера
	3	Незащищенные, основного размера
	4	Незащищенные, увеличенного размера

11.2.15. Вызов подпрограмм ПГП из программ, написанных на языке Ассемблера

Обращение к подпрограммам ПГП производится с помощью макрокоманды CALL, список параметров которой содержит адреса констант или переменных, являющихся аргументами.

Значения функций ITRC, ITBP, RTBP и ITST передаются в регистрах:

целочисленные значения — в общем регистре 0;

вещественные значения — в плавающем регистре 0.

Программа редактируется при использовании библиотеки SYS1.FORTLIB для автоматического вызова и оператора INCLUDE SYSLIB (INCGSP03).

11.2.16. Растровое описание графических символов

Создание таблицы растров

Таблица растров — это отдельная программная секция на языке Ассемблера, которая после редактирования помещается в биб-

лиотеку SYS1.LINKLIB, библиотеку JOBLIB или STEPLIB с именем модуля GSPnn, где nn — двузначное десятичное число от 02 до 99. Номер 01 имеет стандартная таблица растров ПГП.

Таблица растров состоит из трех частей. Первая часть — 8 байт, зарезервированных для ПГП; вторая часть — список адресов (4-байтных адресных констант) растров. Позиция нужного адреса определяется по формуле:

$$\text{позиция адреса} = (\text{адрес списка}) + 4 \times (\text{код символа})$$

Третья часть — это растры, состоящие из последовательности полуслов, описывающей растр. Каждое полуслово имеет следующий формат:

Разряд	Значение	Описание
0	0	Луч включен
	1	Луч выключен (в первом полуслове луч всегда выключен)
1—7	0—127	Координата X точки в матрице 128×128
8	0	Растр не закончен
	1	Растр закончен
9—15	0—127	Координата Y точки в матрице 128×128

Подпрограмма DFSTR. Модифицирует растр. Формат:

CALL DFSTR(нгд,таблица,растр,символ,число)

где нгд — см. 11.2.2;

таблица — целая константа или переменная, задающая таблицу растров. Принимает значения от 1 до 99;

растр — имя переменной или массива, идентифицирующее область основной памяти, содержащей описание нового растра. Описание состоит из последовательности полных слов, по два слова на каждый отрезок растра:

слово1 — положительное или отрицательное целое значение от 1 до 128, задающее координату X конечной точки отрезка в матрице 128×128. Если значение отрицательное, то луч будет погашен;

слово2 — положительное целое значение от 1 до 128, задающее координату Y конечной точки отрезка в матрице 128×128;

символ — целая константа или переменная, задающая код (от 0 до 255) символа, для которого модифицируется растр;

число — целая константа или переменная, задающая число отрезков (пар слов), образующих новый растр.

Подпрограмма RLSTR. Воспроизводит растр. Формат:

CALL RLSTR(нгд,таблица,текст,счетчик,высота,[ширина],[интервал],[ориентация],[номер],[ключ],[выключатель],[X][Y])

где нгд, номер, ключ, выключатель — см. 11.2.2;

таблица — см. выше;

текст — целая константа или переменная, задающая код воспроизводимого символа (от 0 до 255), или имя массива, содержащего коды символов;

счетчик — целая константа или переменная, задающая количество воспроизводимых графических символов, причем положительное значение указывает, что коды символов упакованы по четыре в каждое слово памяти, а отрицательное — в каждом слове имеется только один код, выровненный вправо;

высота — действительная константа или переменная, задающая (в сантиметрах) высоту матрицы 128×128 (значение должно быть не менее 0,3 см и задаваться с точностью до трех знаков после запятой);

ширина — действительная константа или переменная, задающая (в сантиметрах) ширину матрицы 128×128 (значение должно быть не менее 0,3 см и задаваться с точностью до трех знаков после запятой), по умолчанию ширина принимается равной высоте;

интервал — действительная константа или переменная, задающая (в сантиметрах) расстояние между центрами матриц. По умолчанию принимается равным $\frac{4}{3}$ высоты матрицы;

ориентация — имя массива из четырех вещественных чисел с одинарной точностью, задающих ориентацию воспроизводимых символов. Первые два числа задают синус и косинус угла поворота каждого символа относительно положительного направления оси X, вторые два числа — синус и косинус угла между линией, соединяющей центры матриц, и положительным направлением оси. По умолчанию углы поворота принимаются равными 0;

X, Y — константы или переменные, задающие координаты центра первого графического символа. По умолчанию принимается текущее положение луча.

11.2.17 Преобразование координат — подпрограмма CNVRT

Формат оператора вызова подпрограммы CNVRT:

CALL CNVRT (нгд, преобразование, [хвх], [увх], [хвых] [увых])

где нгд — см. 11.2.2;

преобразование — целая константа или переменная, задающая тип преобразования: 1 — преобразовать целые координаты (в растровых единицах) в координаты программиста в соответствии с последним обращением к подпрограмме SDATM; 2 — преобразовать координаты программиста в целые значения (растровые единицы);

хвх, увх — константы или переменные, задающие преобразуемые координаты;

хвых, увых — переменные, которым присваиваются преобразованные значения.

11.2.18. Прямое порождение графических приказов — подпрограмма ORGEN

Формат оператора вызова подпрограммы ORGEN:

CALL ORGEN (нгд, область, длина, [список], [номер], [ключ] [выключатель])

где нгд, номер, ключ, выключатель — см. 11.2.2;

область — имя переменной или массива, идентифицирующее область основной памяти, содержащей графические приказы и данные. Адресация в области должна осуществляться с нулевого адреса и ссылки к другим графическим элементам, не входящим в состав добавляемых приказов, и обозначаться как 'FFFF';

длина — целая константа или переменная, задающая длину области графических приказов и данных;

список — имя переменной или массива, идентифицирующее список полных слов в основной памяти, каждое из которых содержит ключ графического элемента. К последнему есть ссылка в добавляемых графических приказах.

11.2.19. Преобразование входных-выходных числовых данных — подпрограмма BCNV (для программ на языках Ассемблер и Фортран)

Формат оператора вызова подпрограммы BCNV:

CALL BCNV(ввод,вывод,преобразование,длина[,дробь])

где ввод — имя переменной или массива, идентифицирующее преобразуемые данные;

вывод — имя переменной или массива, куда помещаются преобразованные данные;

преобразование — целая константа или переменная, задающая тип преобразования:

101 — из формата с плавающей запятой в символическое представление (формат E);

102 — из формата с плавающей запятой в символическое представление (формат F);

103 — из формата с фиксированной запятой в символическое представление (формат I);

104 — из шестнадцатеричного формата в символическое представление (формат Z);

201 — из формата E в формат с плавающей запятой;

202 — из формата F в формат с плавающей запятой;

203 — из формата I в формат с фиксированной запятой;

204 — из формата Z в шестнадцатеричный формат;

длина — целая константа или переменная, задающая длину числовых данных в коде ДКОИ, длина данных в форматах с плавающей и фиксированной запятой, а также в шестнадцатеричном формате равна 4 байтам;

дробь — целая константа или переменная, задающая длину дробной части для преобразований, в которых участвует формат с плавающей запятой.

11.3. Базисное программное обеспечение графопостроителей

11.3.1. Общие сведения

Базисное программное обеспечение графопостроителей дает возможность писать программы на языке Фортран для рисования различных графиков, чертежей и рисунков на графопостроителях ЕС-7051, ЕС-7052, ЕС-7053, ЕС-7054. Это обеспечение включает шесть базисных подпрограмм, обращение к которым из программ, написанных на языке Фортран, производится с помощью оператора CALL. Подпрограммы создают графические данные и выводят их на графопостроитель, магнитную ленту или перфоленту.

Подпрограммы объединяются с вызывающей программой Редактором связей. Задание, использующее базисные подпрограммы, выглядит следующим образом:

```
//имя                JOB
//                  EXEC      FORTGCLG
//FORT.SYSIN         DD      *
//                  основная программа
/*
//LKED.SYSIN         DD      *
//      INCLUDE      SYSLIB   (IFFGAA07) } только для ЕС-7054
/*
//GO.SYSIN           DD      *
//                  входные данные
/*
//GO.FTnnF001        DD      UNIT=cuu
```

Здесь nn — десятичное число от 00 до 99, задаваемое в качестве значения параметра «устройство» в обращении к подпрограмме PLOTS (см. ниже), а cuu — адрес графопостроителя, НМЛ или перфоленточного устройства ввода-вывода. Для магнитной ленты оператор DD выглядит следующим образом:

```
//GO.FTnnF001    DD      UNIT=5010,VOL=SER=метка,DSN=имя,
                        DCB=(BLKSIZE=4000,RECFM=U,DEN=0),
                        LABEL=(I,тип),DISP=(NEW,KEEP)
```

где метка — метка тома магнитной ленты;
имя — имя набора данных, выводимого на магнитную ленту;
l — номер файла на магнитной ленте;
тип — тип разметки магнитной ленты (например, SL или NL).

11.3.2. Подпрограмма PLOTS

Подпрограмма PLOTS имеет восемь точек входа.

Точка входа PLOTS — инициализация. Вход PLOTS предназначен для первоначальной инициализации подпрограммы PLOTS и всего базисного программного обеспечения. К нему обращаются

один раз в начале работы графической программы перед обращением к другим подпрограммам. Формат:

CALL PLOTS (буфер,размер,устройство,режим)

где буфер — метка области вывода, в которой должны накапливаться выходные графические данные;

размер — целочисленная константа или переменная, указывающая количество слов в области вывода. Величина этого параметра должна соответствовать величине массива области вывода, определяемой оператором DIMENSION;

устройство — целочисленная константа или переменная, указывающая соответствующий оператор DD (от 00 до 99). По умолчанию или при ошибке принимается равной 10;

режим — целочисленная константа или переменная, указывающая вид графических данных и тип графопостроителя (этот параметр игнорируется для ЕС-7054). Значением параметра является двузначное число mn. Значения m:

- 1 — вывод графических данных в приказах;
- 2 — вывод графических данных в инкрементальных командах через блок преобразования данных (БПД);
- 3 — вывод графических данных в инкрементальных командах через блок управления построителем (БУП) с максимальной скоростью;
- 4 — вывод графических данных в инкрементальных командах через БУП с пониженной скоростью и улучшенным качеством.

Значения n задают тип графопостроителя, для которого данные выводятся на магнитную ленту или перфоленту (в режиме канального подключения n игнорируется):

- 1 — графопостроитель ЕС-7051;
- 2 — графопостроитель ЕС-7052;
- 3 — графопостроитель ЕС-7053.

Точка входа PLOT — вычерчивание отрезков. Вход PLOT предназначен для перемещения пишущего элемента по прямой. Формат:

CALL PLOT(x,y,состояние)

где x, y — действительные константы или переменные, выражающие координаты конца отрезка (см);

состояние — целочисленная константа или переменная со знаком. Имеет следующие значения:

- 2 — пишущий элемент во время движения опущен;
- 3 — пишущий элемент во время движения поднят;
- 2 или —3 — аналогично 2 и 3, но точка конечного положения пишущего элемента становится новым началом координат;

12, 13, —12, —13 — аналогично 2, 3, —2 и —3; кроме того, предписывает преобразование заданных параметров x и y, которое описано ниже для точки входа OFFSET (см. с. 245);

999 — окончание формирования графических данных. Пишущий элемент поднимается;

1 — пишущий элемент остается в прежнем состоянии.

Точка входа FACTOR — масштабирование. Вход FACTOR позволяет увеличить или уменьшить размеры всего чертежа. Формат:

CALL FACTOR (масштаб)

где масштаб — действительная константа или переменная, задающая отношение нового размера чертежа к нормальному размеру.

Точка входа WHERE — самоопределение. Вход WHERE позволяет определить текущие координаты пишущего элемента и масштабный множитель. Формат:

CALL WHERE (гх,гу,масштаб)

где гх, гу — действительные переменные, задающие текущие координаты х и у положения пишущего элемента;

масштаб — действительная переменная, задающая действующий в настоящий момент масштабный множитель (если перед этим обращением к входу FACTOR не было, то принимается 1.0).

Точка входа NEWPEN — выбор пишущего элемента. Формат:

CALL NEWPEN (перо)

где перо — целочисленная константа или переменная, определяющая номер пишущего элемента. Новый элемент перемещается в позицию старого пишущего элемента. Значениями могут быть 1, 2, 3 и (для EC-7054) 4. По умолчанию принимается 1.

Точка входа OFFSET — выборочное преобразование. Вход OFFSET задает преобразование координат х и у положения пишущего элемента в случае обращения к входу PLOT с «состояния» 12, 13, —12, —13. Формат:

CALL OFFSET (xs,ys,xm,ym)

где xs, ys — действительные константы или переменные, задающие сдвиг по координатам х и у;

xm, ym — действительные константы или переменные, задающие масштабный множитель по координатам х, у.

Формулы преобразования:

$$\begin{aligned}x' &= (x - xs) / xm \\ y' &= (y - ys) / ym\end{aligned}$$

Точка входа TYPL — тип линий (только для EC-7051, EC-7053 и EC-7054). Формат:

CALL TYPL (линия)

где линия — целое число, задающее тип вычерчиваемых линий:

1 — сплошная;

2 — пунктирная;

3 — штрих-пунктирная.

Точка входа PCIRC — вычерчивание дуг окружностей (только для EC-7051 и EC-7053). Вход PCIRC вычерчивает дуги окружностей в режиме круговой интерполяции. Формат:

CALL PCIRC (хк,ук,х0,у0,состояние)

где хк и ук — действительные константы или переменные, задающие координаты Х и У конца вычерчиваемой дуги (см);

x_0 и y_0 — действительные константы или переменные, задающие координаты центра вычерчиваемой дуги (см);

состояние — целочисленная константа или переменная, управляющая состоянием пишущего элемента и направлением круговой интерполяции:

32, 33, 31 — аналогично 2, 3, 1 (см. выше), при этом круговая интерполяция осуществляется по часовой стрелке;

42, 43, 41 — аналогично 2, 3, 1 (см. выше), при этом круговая интерполяция осуществляется против часовой стрелки.

11.3.3. Подпрограмма SYMBOL

Подпрограмма SYMBOL вычерчивает символы любого размера под любым углом. Имеются два формата вызова подпрограммы — стандартный и специальный. В табл. 11.1 приводится перечень символов, которые можно вычертить подпрограммой SYMBOL, и их кодов.

Стандартный формат вызова подпрограммы SYMBOL. Формат:

CALL SYMBOL ($x, y, \text{высота}, \text{текст}, \text{угол}, \text{число}$)

где x и y — действительные числа или переменные, задающие координаты нижнего левого угла первого вычерчиваемого символа. Если требуется вычертить символ, начиная с текущей позиции пишущего элемента, параметры x и y задаются равными 999.0;

высота — действительное число или переменная, задающая высоту (см) вычерчиваемого символа. Ширина символа, включая промежуток между символами, равна высоте;

текст — имя области, в которой находится вычерчиваемый текст;

угол — действительное число или переменная, задающие угол в градусах относительно оси x вычерчиваемого текста;

число — целое число или переменная, задающие длину текста.

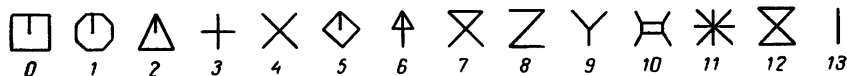
Специальный формат вызова подпрограммы SYMBOL. Рисует один символ, соответствующий значению «экви». Формат:

CALL SYMBOL ($x, y, \text{высота}, \text{экви}, \text{угол}, \text{код}$)

где высота, угол — аналогичны стандартному вызову (см. выше);

x и y — задают геометрический центр вычерчиваемого символа;

экви — целое число или переменная, указывающая вычерчиваемые символы:



код — целочисленная отрицательная константа или переменная, задающая положение пишущего элемента при движении к точке (x, y):

—1 — пишущий элемент поднят;

—2 — пишущий элемент опущен.

Таблица 111

XX	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0					Пробел	&	-	Σ	ц	ц̣	а	а	{	}	\	0
1		су1				Y	/		а	j		ы	A	J		1
2							Ψ	\leq	б	k	s	з	B	K	S	2
3	+					*	χ	\geq	с	р	t	ш	C	L	T	3
4	X				+	Σ	ш	Δ	d	m	u	э	D	M	U	4
5		HC			X		л	Υ	e	n	v	ц	E	N	V	5
6						μ	α	ю	f	o	w	ч	F	O	W	6
7	X					π	δ	ν	g	p	x	ъ	G	P	X	7
8	Z				X	ϕ	ϵ	б	h	q	y	ю	H	Q	Y	8
9	Y	KH			Z		π		i	r	z	\$	I	R	Z	9
A					[]		:	ð	к	\rightarrow	б	#	\wedge	\sim	3
B	*				.		.	#	\leftarrow	л	x	ц	H	V	y	ш
C	Σ				<	*	%	@	ф	м	т	д	Й	П	Ж	э
D					()	-	\leftarrow	е	н	∞		\neq	Я	\approx	ц
E			дA		+	:	>	=	↑	↓	ж	ф	л	С	б	4
F			3B		!	—	?	"	u	н	в	г	☆	с	б1	

11.3.4. Подпрограмма NUMBER

Подпрограмма NUMBER превращает действительную переменную (число с плавающей запятой) в фиксированное десятичное число для последующего вычерчивания подпрограммой SYMBOL. Формат:

CALL NUMBER(х,у,высота,плав,угол,точность)

где х, у, высота, угол — аналогичны параметрам вызова подпрограммы SYMBOL (см. выше);

плав — действительное число или переменная, преобразуемая и вычерчиваемая подпрограммой;

точность — целое число или переменная, задающая точность преобразования. Если значение параметра «точность» больше нуля, то оно определяет число вычерчиваемых знаков после десятичной точки. Если значение параметра равно 0, то вычерчивается целая часть числа и десятичная точка. Если «точность» = -1, то вычерчивается целая часть числа без десятичной точки. Если «точность» < -1, то вычерчиваются первые к цифр целой части числа, где $k = (n - \text{«точность»}) + 1$; n — число цифр в целой части числа. Величина параметра «точность» не должна превышать 9.

11.3.5. Подпрограмма SCALE — масштабирование массивов данных

Подпрограмма SCALE масштабирует массив данных исходя из минимального и максимального значения данных в массиве и длины оси. Формат:

CALL SCALE(массив,длина,точки,инкр)

где массив — указывает первое слово масштабируемого массива данных;

длина — действительное число или переменная, указывающая длину оси, к которой выбирается масштаб данных (должна быть больше 1,0 см);

точки — указывает число масштабируемых элементов массива. Оператор DIMENSION должен определять массив с учетом дополнительных двух слов в конце массива, куда подпрограмма SCALE запишет начальное значение и масштабный множитель;

инкр — целое число или переменная, указывающая приращение, используемое при масштабировании массива. Например, если инкр = 3, то будет масштабироваться каждый третий элемент массива. Отрицательное значение указывается в том случае, когда массив просматривается с конца.

11.3.6. Подпрограмма AXIS — вычерчивание осей

Подпрограмма AXIS чертит оси любой длины под любым углом, делит их на двухсантиметровые интервалы, выполняет надписи под делениями. Формат:

CALL AXIS (х,у,текст,число,длина,угол,перв,дельта)

где x , y — действительные числа или переменные, задающие (в сантиметрах) координаты начала осевой линии;

текст — указывает область заголовка, который центрируется и вычерчивается параллельно осевой линии. Символы имеют фиксированную высоту 0,28 см;

число — положительное или отрицательное целое число или переменная, задающая число символов в заголовке. Знак указывает, на какой стороне оси, положительной (против часовой стрелки) или отрицательной (по часовой стрелке) выполняются надписи;

длина — действительное число или переменная, задающая длину осевой линии (в см);

угол — действительное число или переменная, задающая угол, под которым чертится ось;

перв — действительное число или переменная, задающая начальное значение первой метки шкалы оси;

дельта — действительное число или переменная, определяющая число единиц данных на сантиметр шкалы.

11.3.7. Подпрограмма LINE — вычерчивание графиков

Подпрограмма LINE чертит график, используя данные в двух массивах X и Y . Формат:

CALL LINE (X, Y , точки, инкр, тип, экв)

где X и Y — указывают массивы, содержащие значения абсцисс и ординат и масштабные параметры;

точки — целое число или переменная, указывающая число точек в каждом из массивов X и Y ;

инкр — целое число или переменная, используемая аналогично описанию для подпрограммы SCALE (см. 11.3.5);

тип — целое число или переменная, описывающая тип линии и задающая частоту вычерчивания линии. Например, если «тип»=4, то специальный символ, задаваемый параметром «экв», будет вычерчиваться в каждой четвертой точке. Если «тип»=0, то символы не вычерчиваются. Положительное значение параметра «тип» задает соединение точек данных отрезками прямых, при отрицательном значении — соединяющие линии не вычерчиваются;

экв — целое число или переменная, задающая графический символ (от 0 до 13) (см. с. 246).

11.4. Функциональные подпрограммы для графопостроителей

Функциональные подпрограммы вызываются в программах на языке Фортран с помощью оператора CALL. После трансляции программу необходимо огредактировать, используя в качестве системной библиотеки для Редактора связей библиотеку SYS1.FORTLIB. Включение подпрограмм для графопостроителей в библиотеку SYS1.FORTLIB производится при генерации ОС ЕС.

11.4.1. Подпрограммы общего назначения

Подпрограмма CIRCL — вычерчивание дуг, окружностей или спиралей. Формат:

CALL CIRCL (x,y,уголн,уголк,радиусн,радиуск,тип)

где x, y — координаты начальной точки дуги (в см);

уголн, уголк — угол от положительной полуоси x до начальной и конечной точек дуги (в градусах);

радиусн, радиуск — начальный и конечный радиусы дуги;

тип 0.0 — сплошная линия, 0.5 — штриховая линия.

Подпрограмма DASHL — вычерчивание штриховой линии через заданную последовательность точек. Формат:

CALL DASHL (x,y,точки,инкр)

где x, y — массивы координат точек;

точки — число точек;

инкр — инкремент выбора точек из массива.

Подпрограмма DASHP — вычерчивание штриховой линии из текущего положения в заданную точку. Формат:

CALL DASHP (x,y,штрих)

где x, y — координаты конечной точки;

штрих — длина штриха (в см).

Подпрограмма ELIPS — вычерчивание эллипсов. Формат:

CALL ELIPS (x,y,бпось,мпось,угол,уголн,уголк,состояние)

где x, y — координаты начальной точки эллипса или дуги;

бпось, мпось — размеры большой и малой полуосей эллипса;

угол — угол между положительной полуосью x и положительным направлением большой оси эллипса;

уголн, уголк — угол от положительного направления большой оси эллипса до начальной и конечной точек дуги;

состояние — состояние пишущего элемента: 2 — опущен, 3 — поднят.

Подпрограмма FIT — вычерчивание ветви гиперболы. Формат:

CALL FIT(x1,y1,x2,y2,x3,y3)

где x1, y1, x2, y2, x3, y3 — координаты трех точек, через которые проводится гипербола.

Подпрограмма GRID — вычерчивание линейных сеток. Формат.

CALL GRID(x,y,Δx,Δy,Knx,Kny)

где x, y — координаты левого нижнего угла сетки;

Δx, Δy — расстояния между линиями сетки;

Knx, Kny — количество линий минус 1 по осям x и y.

Подпрограмма POLY — вычерчивание многоугольников или звезд. Формат:

CALL POLY (x,y,длина,вершина,угол)

где x, y — координаты начальной вершины;

длина — длина стороны фигуры;

вершина — количество вершин многоугольника (положительное) или звезды (отрицательное);

угол — угол между положительной полуосью x и первой стороной многоугольника или звезды.

Подпрограмма RECT — вычерчивание прямоугольников. Формат:

CALL RECT ($x, y, \text{высота}, \text{ширина}, \text{угол}, \text{состояние}$)

где x, y — координаты нижнего левого угла прямоугольника;

высота, ширина — высота и ширина прямоугольника;

угол — угол между положительной полуосью x и основанием прямоугольника;

состояние — состояние пишущего элемента: 2 — опущен, 3 — поднят.

11.4.2. Научно-технические подпрограммы

Подпрограмма CURVX — вычерчивание полиномиальной функции от x . Формат:

CALL CURVX ($x_п, x_k, k_1, n_1, k_2, n_2, k_3, n_3, k_4, n_4$)

где $x_п, x_k$ — начальное и конечное значения аргумента x ;

k_1, k_2, k_3, k_4 — коэффициенты полинома;

n_1, n_2, n_3, n_4 — степень каждого члена полинома.

Подпрограмма CURVY — вычерчивание полиномиальной функции от y . Формат:

CALL CURVY ($y_п, y_k, k_1, n_1, k_2, n_2, k_3, n_3, k_4, n_4$)

Операнды имеют то же значение, что и для подпрограммы CURVX, но для аргумента y .

Подпрограмма FLINE — вычерчивание графиков. Формат:

CALL FLINE ($x, y, \text{точки}, \text{инкр}, \text{тип}, \text{экв}$)

где x, y — имена массивов абсцисс (X) и ординат (Y) и соответствующих масштабных параметров;

точки — число точек в каждом массиве. Если отрицательное число — через точки проводится плавная кривая, положительное — точки соединяются прямыми отрезками;

инкр — инкремент выбора точек из массива;

тип — если 0, то точки графика соединяются линией, если n , то кроме линии в каждой n -й точке вычерчивается специальный символ, определяющий параметры «экв»; если — n , то вычерчиваются только символы в каждой n -й точке;

экв — число от 0 до 13, задающее специальный центрированный символ (см. с. 246).

Подпрограмма CRVPT — вычерчивание кривых, аппроксимирующих последовательность точек. Формат:

CALL CRVPT ($x, y, \text{экв}, \text{точки}, \text{инкр}, \text{высота}, \text{ширина}, \text{текст}, \text{число}, \text{текст1}, \text{число1}, \text{текст2}, \text{число2}, \text{степень}$)

где x, y — имена массивов абсцисс и ординат и соответствующих масштабных параметров;

экв — имя массива, содержащего целые числа (0—13), задающие центрированные символы. Первое число массива может быть отрицательным, тогда первый символ будет вычерчиваться в каждой точке;

точки — число точек; если отрицательное, то кривая вычерчивается на том же графике, что и во время предыдущего обращения к подпрограмме CRVPT;

инкр — инкремент выбора из массива; если отрицательный, то в конце вычерчивания пишущий элемент возвратится в начало координат;

высота — высота вычерчиваемого графика;

ширина — ширина вычерчиваемого графика;

текст — имя массива, содержащего заголовки;

число — число символов в заголовке;

текст1 — имя массива, содержащего заголовок оси X;

число1 — число символов в заголовке оси X;

текст2 — имя массива, содержащего заголовок оси Y;

число2 — число символов в заголовке оси Y;

степень — имя массива целых чисел, задающих степени полиномов. Если первое значение равно 0, то точки на графике будут соединены отрезками прямой, а следующее значение будет степенью первого полинома.

Подпрограмма FNUM — вычерчивание действительных чисел. Формат:

CALL FNUM(x,y,высота,число,точность)

где x, y — координаты левого угла первого символа;

высота — высота вычерчиваемого числа;

число — вычерчиваемое число;

точность — число знаков справа от десятичной точки.

Подпрограмма LGAXS — вычерчивание осей с логарифмической шкалой. Формат:

CALL LGAXS(x,y,текст,число,длина,угол,перв,дельта)

где x, y — координаты начальной точки оси;

текст — имя массива, содержащего заголовки оси;

число — число символов в заголовке оси, подписи чертятся на положительной или отрицательной стороне оси в зависимости от знака числа;

длина — длина осевой линии;

угол — угол между положительным направлением оси x графо-строителя и положительным направлением вычерчиваемой оси;

перв — начальное значение, вычерчиваемое у первого деления шкалы оси;

дельта — число логарифмических единиц данных на сантиметр оси.

Подпрограмма LGLIN — вычерчивание графиков в логарифмической или полулогарифмической шкале. Формат:

CALL LGLIN(x,y,точки,инкр,тип,экв,график)

где x, y — имена массивов абсцисс и ординат, а также их масштабных параметров;

точки — число точек;

инкр, тип, экв — те же, что и в подпрограмме FLIN (см. выше);

график — 0 — логарифмическая шкала (логарифмический масштаб по осям x и y), —1 — полулогарифмическая шкала (логарифмический масштаб по оси x и линейный по оси y), 1 — полулогарифмическая шкала (логарифмический масштаб по оси y и линейный по оси x).

Подпрограмма POLAR — вычерчивание графиков в полярной системе координат. Формат:

CALL POLAR(гх,ау,точки,инкр,тип,экв,радиус,масштаб)

где гх — имя массива, содержащего радиальные значения точек;
ау — имя массива, содержащего угловые значения точек в радианах;

точки — число точек;

инкр — инкремент выбора из массивов;

тип — если 0, точки графика соединяются отрезками прямых, если n , то кроме линии, соединяющей точки графика, в каждой n -й точке будет вычерчиваться специальный символ (определенный параметром «экв»), если $-n$, то вычерчиваются только символы;

экв — число от 0 до 13, задающее специальный центрированный символ (см. 246);

радиус — величина максимального радиуса; если >0 , то масштабирование осуществляется автоматически; если ≤ 0 , то в качестве масштабного множителя используется значение параметра «масштаб»;

масштаб — масштабный множитель.

Подпрограмма SCALG — масштабирование данных для вычерчивания в логарифмической шкале. Формат:

CALL SCALG(массив,длина,точки,инкр)

где массив — имя массива данных;

длина — длина оси;

точки — число точек данных в массиве;

инкр — инкремент выбора данных из массива.

Подпрограмма SMOOT — вычерчивание гладкой кривой через заданные точки. Формат:

CALL SMOOT(x, y , состояние)

где x, y — координаты точек;

состояние — положительное число — прямолинейный режим работы (1 — пишущий элемент остается в том же состоянии, 2 — пишущий элемент опущен, 3 — пишущий элемент поднят); отрицательное число — криволинейный режим работы (0 — начать вычерчивание разомкнутой гладкой кривой с точки с координатами (x, y); —1 — начать вычерчивание замкнутой гладкой кривой с точки с координатами (x, y); —2 — параметры x и y задают сле-

дующую точку, через которую должна пройти кривая, пишущий элемент опущен; —3 — параметры x и y задают следующую точку, через которую должна пройти кривая, пишущий элемент поднят; —24 — параметры x и y задают последнюю точку кривой; если кривая замкнутая, то последняя точка соединяется с первой).

11.4.3. Экономические подпрограммы

Подпрограмма AXISB — вычерчивание осей с экономической разметкой. Формат:

CALL AXISB(x, y , текст, число, длина, угол, перв, дельта, язык)

где x, y — координаты начальной точки оси;

текст — имя массива, содержащего заголовков оси; высота символов — 0,36 см;

число — число символов в заголовке; заголовок вычерчивается на стороне оси, соответствующей знаку числа;

длина — длина вычерчиваемой оси; если положительная, то масштабный множитель вычерчивается на английском языке; если отрицательная, то масштабный множитель вычерчивается на языке, задаваемом параметром «язык»;

угол — угол между положительным направлением оси x и вычерчиваемой осью;

перв — начальное значение, вычерчиваемое для первой метки оси;

дельта — число единиц данных на сантиметр шкалы;

язык — 0 — английский, 1 — русский.

Подпрограмма AXISC — вычерчивание осей с календарной разметкой. Формат:

CALL AXISC(x, y , текст, число, длина, угол, перв, дельта, язык)

где x, y , текст, число, длина, угол, язык — те же, что для подпрограммы AXISB (см. выше);

перв — число, задающее название месяца:

1.0 JAN	ЯНВ	7.0 JUL	ИЮЛ
2.0 FEB	ФЕВ	8.0 AUG	АВГ
3.0 MAR	МАР	9.0 SEP	СЕН
4.0 APR	АПР	10.0 OCT	ОКТ
5.0 MAY	МАЙ	11.0 NOV	НОЯ
6.0 JUN	ИЮН	12.0 DEC	ДЕК

дельта — число месяцев года между соседними метками оси;

язык — см выше.

Подпрограмма BAR — вычерчивание прямоугольных диаграмм. Формат:

CALL BAR(x, y , угол, высота, ширина, штрих, код, клпх)

где x, y — координаты нижнего левого угла вычерчиваемого прямоугольника;

угол — угол между основанием прямоугольника и положительной полуосью x ;

высота и ширина — высота и ширина прямоугольника;
 штрих — высота штрихуемой части прямоугольника;
 код 1 — штриховка отсутствует; 2 — штриховка слева направо;
 3 — штриховка справа налево; 4 — штриховка слева направо и
 справа налево;
 кпх — число линий штриховки в одном сантиметре основания
 штриховки.

Подпрограмма LBAXS — вычерчивание осей с экономической
 разметкой и логарифмической шкалой. Формат:

CALL LBAXS(х,у,текст,число,длина,угол,перв,дельта[,язык])

где х, у, текст, число, длина, угол, язык — те же, что и для под-
 программы AXISB (см. выше);

перв — начальное значение первой метки шкалы оси;

дельта — число логарифмических единиц данных на сантиметр
 оси;

язык — см. выше.

Подпрограмма SHADE — штриховка. Формат:

CALL SHADE(х1,у1,х2,у2,длина,угол,точки1,точки2,инкр1,инкр2)

где х1, у1 — имена массивов точек первой вычерчиваемой линии,
 а также масштабных параметров;

х2, у2 — имена массивов точек второй вычерчиваемой линии, а
 также масштабных параметров;

длина — расстояние между соседними линиями штриховки;

угол — угол линий штриховки по отношению к положительной
 полуоси х;

точки1, точки2 — число точек в массивах х1, у1 и х2, у2;

инкр1, инкр2 — инкременты выбора из массивов х1, у1 и х2, у2.

11.4.4. Чертежные подпрограммы

Подпрограмма AROHD — вычерчивание стрелок. Формат:

CALL AROHD(х,у,хк,ук,длина,ширина,код)

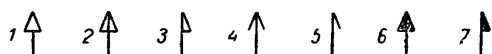
где х, у — координаты начала отрезка, на конце которого вычер-
 чивается стрелка; эти координаты игнорируются, если значение
 параметра «код» меньше 0. В этом случае началом отрезка явля-
 ется текущее положение пишущего элемента;

хк, ук — координаты конца вычерчиваемой стрелки;

длина — длина вычерчиваемой стрелки;

ширина — ширина стрелки; если 0, то ширина стрелки прини-
 мается равной $\frac{2}{3}$ параметра «длина»;

код — двузначное число; первая цифра: 0 — отрезок не вычер-
 чивается, вычерчивается только стрелка, 1 — вычерчивается отрез-
 ок и стрелка, 2 — вычерчивается отрезок со стрелками на обоих
 концах; вторая цифра — указывает вид стрелки:



Подпрограмма ARROW — вычерчивание линии со стрелкой на конце. Формат:

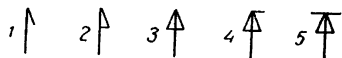
CALL ARROW (x,y,точки,инкр,тип)

где x, y — имена массивов, содержащих координаты заданных точек и масштабные параметры;

точки — число точек;

инкр — инкремент выбора из массивов;

тип — число, задающее вид стрелки:



Подпрограмма CNTRL — вычерчивание осевых линий. Формат:

CALL CNTRL (x,y,точки,инкр)

где x, y, точки, инкр — те же, что и для подпрограммы ARROW (см. выше).

Подпрограмма DIMEN — вычерчивание размерных линий. Формат:

CALL DIMEN (x,y,размер,угол,масштаб)

где x, y — координаты начальной точки вычерчиваемой размерной линии;

размер — величина проставляемого размера;

угол — угол между вычерчиваемой линией и положительной полуосью X;

масштаб — масштабный множитель.

Подпрограмма LABEL — вычерчивание надписей между заданными точками. Формат:

CALL LABEL (x1,y1,x2,y2,текст,точки,высота,код,число,плав,точность)

где x1, y1 — координаты начальной точки;

x2, y2 — координаты конечной точки;

текст — имя массива, содержащего символы текста;

точки — число вычерчиваемых символов;

высота — высота символов;

код — число, задающее расположение символов: 1 — надпись располагается на некотором расстоянии относительно линии между точками, 2 — надпись располагается на расстоянии относительно линии между точками, заданном параметром «число», 11 — то же, что и 1, кроме того, в конце надписи вычерчивается число, заданное параметром «плав», 12 — то же, что и 2; вычерчивается число, заданное параметром «плав»; при отрицательном значении кода надпись поворачивается на 180° относительно линии между точками;

число — расстояние от линии между точками до надписи;

плав — число в составе надписи, если код 11 или 12;

точность — число символов после десятичной точки в параметре «плав».

11.5. Средства описания изображений для графических дисплеев

11.5.1. Формат графического приказа

Изображение на экране формируется под управлением программы графических приказов, состоящей из последовательности чередующихся графических приказов и данных. Формат графического приказа (XX — код операции):

Байт1 Байт2

2A	XX
----	----

2-байтовый приказ

Байт1 Байт2 Байт3 Байт4

2A	XX	адрес
----	----	-------

4-байтовый приказ

Байт1 Байт2 Байт3 Байт4 Байт5 Байт6

2A	XX	адрес	адресная кон- станта или кон- станта данных
----	----	-------	---------------------------------------------------

6-бай-
товый
приказ

Графические приказы создаются с помощью макрокоманд, которые имеют следующий формат:

[метка] оператор [операнд,...]

Далее при описании макрокоманд построения графических приказов приводятся только форматы операндов, за которыми следует поясняющий текст.

11.5.2. Макрокоманды инициализации

Макрокоманда GINIT. Обязательна и является первой макрокомандой в программе графических приказов. Макрокоманда должна иметь метку, если приказы и данные, управляемые данной макрокомандой GINIT, символически указываются в приказах и данных, управляемых другой макрокомандой GINIT или GBFLM. Длина метки не должна превышать пяти знаков, так как после компиляции к ней слева приписываются буквы INB. Формат операндов:

[BPX=целое-число]

Начальное значение в растровых единицах счетчика координаты X положения луча (от 0 до 4095). По умолчанию равно 0.

[BPU=целое-число]

Начальное значение в растровых единицах счетчика координаты Y положения луча (от 0 до 4095). По умолчанию равно 0.

[BLC=целое-число]

Начальное значение в байтах счетчика адреса буфера (от 0 до 32 767). По умолчанию равно 0.

[BLIM=целое-число]

Максимальное значение в байтах счетчика адреса буфера (от 0 до 32 767). По умолчанию равно 32 767.

[BLP=целое-число]

Значение в байтах (от 0 до 32 767) буферной точки загрузки цепочки приказов, следующей за макрокомандой GINIT. По умолчанию равно 0.

Макрокоманда GBPST. Инициализирует счетчики координат положения луча.

[BPX=целое-число]

Указывает новое значение в растровых единицах (от 0 до 4095) счетчика координаты X положения луча. Если операнд опущен, а BRY указан, то сохраняется старое значение BPX.

[BRY=целое-число]

Указывает новое значение в растровых единицах (от 0 до 4095) счетчика координаты Y положения луча. Если операнд опущен, а BPX указан, то сохраняется старое значение BRY.

Если опущены оба операнда, то принимается $BPX=BRY=0$.

Макрокоманда GBFLM. Инициализирует счетчики адреса буфера и устанавливает границу буфера.

Макрокоманда должна иметь метку, если изменяется точка загрузки буфера, а приказы и данные, управляемые данной макрокомандой GBFLM, символически указываются в приказах или данных, управляемых другой макрокомандой GINIT или GBFLM. Длина метки не должна превышать пяти знаков, так как после компиляции к ней слева приписываются буквы INB. Формат операндов:

[BLC=целое-число]

Указывает новое значение в байтах (от 0 до 32 767) счетчика адреса буфера. Если операнд опущен, а какой-нибудь другой указан, то сохраняется старое значение BLC.

[BLIM=целое-число]

Указывает новое значение в байтах (от 0 до 32 767) границы буфера. Если операнд опущен, а какой-нибудь другой указан, то сохраняется старое значение BLIM.

[BLP=целое-число]

Указывает новое значение в байтах (от 0 до 32 767) буферной точки загрузки для последующей цепочки приказов. Если операнд опущен, а какой-нибудь другой указан, то сохраняется старое значение BLP.

Если опущены все операнды, то принимается $BLC = BLP = 0$, $BLIM = 32767$.

Макрокоманда GIBLC. Увеличивает при компиляции счетчик адреса буфера на указанное число байт. Формат операндов:

целое-число

Десятичное целое число без знака, указывающее число байт, добавляемое к счетчику адреса буфера.

11.5.3. Вспомогательные макрокоманды

Макрокоманда GBINF. Распечатывает в листинге адрес буферной точки загрузки, счетчик адреса буфера и границу буфера. Формат операндов:

[H|D|знак]

где H — печатать счетчик адреса буфера в шестнадцатеричной системе счисления; D — печатать счетчик адреса буфера в десятичной системе счисления; любой другой знак (знаки) — печатать в обеих системах счисления.

[точка-загрузки-буфера]

Любой знак вызывает печать адреса буферной точки загрузки в десятичной системе счисления.

[граница-буфера]

Любой знак вызывает печать значения границы буфера в десятичной системе счисления.

Если опущены все три операнда, то все три элемента печатаются в десятичной системе счисления. Если опущены один или два операнда, то соответствующие им элементы не печатаются.

Макрокоманда GBPOS. Распечатывает в листинге содержимое счетчиков координат X и Y положения луча в десятичной системе счисления. Поле операндов пусто.

Макрокоманда GSBPOS. Запоминает координаты положения луча. Формат операндов:

[адрес-X]

Символический адрес полуслова в основной памяти.

[адрес-Y]

Символический адрес полуслова в основной памяти.

Любой операнд может быть опущен.

Макрокоманда GSBLC. Запоминает текущее содержимое адреса буфера. Формат операнда:

[адрес]

Символический адрес полуслова в основной памяти.

11.5.4. Макрокоманды образования приказов

Мнемоника каждой макрокоманды соответствует мнемонике создаваемого графического приказа. Графические приказы начинаются с четного буферного адреса.

Макрокоманда GSRT. Иницирует перезапись изображения не чаще 50 раз в секунду. Должна быть первым в последовательности приказов. Приказ GSRT также переводит дисплей в состояния GESD и GPD1 (см. ниже). Поле операндов пусто.

Макрокоманда GERM. Переводит дисплей в режим построения абсолютной точки. Последующие байты данных, которые строятся макрокомандой GDV, управляют положением луча. Поле операндов пусто.

Макрокоманда GEVM. Переводит дисплей в режим построения абсолютного вектора. Последующие байты данных, которые строятся макрокомандой GDV, управляют положением луча. Поле операндов пусто.

Макрокоманда GER12. Переводит дисплей в инкрементальный точечный режим. Последующие байты данных, которые строятся макрокомандой GDV, задают инкремент для нового положения луча. Поле операндов пусто.

Макрокоманда GEV12. Переводит дисплей в инкрементальный векторный режим. Последующие байты данных, которые строятся макрокомандой GDV, задают инкремент для нового положения луча. Поле операндов пусто.

Макрокоманда GECF. Переводит дисплей в режим вычерчивания знаков. Последующие байты данных рассматриваются как коды символов. Знаки не будут защищены от замещения их данными, вводимыми с клавиатуры. Формат операнда:

[размер]

В или BASIC — указывает основной размер знака;

L или LARGE — указывает большой размер знака.

Если операнд опущен или указан неправильно, принимается BASIC.

Макрокоманда GESP. Приказ аналогичен приказу GECF, за исключением того, что знаки будут защищены от замещения их данными, вводимыми с клавиатуры. Поле операндов пусто.

Макрокоманда GESD. Переводит дисплей в состояние, которое разрешает обнаружение светового пера, если переключатель светового пера включен и остается в этом положении в течение выполнения приказа GSRT и последующего пересечения лучом апертуры светового пера. Если дисплей находится в режиме разрешения прерывания при обнаружении светового пера (режим GPD1), то перезапись прекращается и формируется сигнал внимания. Состояние обнаружения светового пера сохраняется до выполнения приказа GSRT, GESD или GDRD и может проверяться приказом перехода GTND (см. ниже). Если же дисплей в момент обнаружения светового пера находился в режиме GDRD, то состояние обнаружения запоминается, но сигнал внимания не формируется. Это состояние может проверяться приказом перехода GTDD (см. ниже). Поле операндов пусто.

Макрокоманда GENSD. Переводит дисплей в состояние, которое разрешает обнаружение светового пера независимо от переключателя светового пера или приказа GSRT. Если дисплей нахо-

дится в режиме разрешения прерывания при обнаружении светового пера (режим GPDI), то перезапись прекращается и формируется сигнал внимания. Состояние обнаружения светового пера сохраняется до выполнения приказа GSRT, GESD или GDRD и может проверяться приказом перехода GTND (см. ниже). Если дисплей в момент обнаружения светового пера находился в режиме GDRD, то состояние обнаружения запоминается, но сигнал внимания не формируется. Это состояние может проверяться приказом перехода DTDG (см. ниже). Поле операндов пусто.

Макрокоманда GDPD. Запрещает обнаружение светового пера. Из этого режима дисплей переводится приказами GESD, GSRT или GENSD. Поле операндов пусто.

Макрокоманда GDRD. Переводит дисплей в режим, при котором обнаружение светового пера не прекращает перезаписи, не выдает сигнала внимания, а только запоминается (всегда одно последнее обнаружение). Обнаружение запоминается до тех пор, пока не будет выдан приказ GTDD, GTND, GPDI или GSRT. В этом случае GTDD выполняет переход к адресу, указанному в приказе GTDD, GTND не выполняет никакой операции, GPDI прекращает перезапись и формирует сигнал внимания, GSRT синхронизирует начало перезаписи. Дисплей выводится из режима GDRD приказом GPDI или GSRT. Поле операндов пусто.

Макрокоманда GPDI. Переводит дисплей в режим, при котором любое отложенное или будущее обнаружение останавливает перезапись и формирует сигнал внимания. Поле операндов пусто.

Макрокоманда GTRU. Передает управление по указанному адресу буфера. Формат операндов:

адрес

Символический или абсолютный буферный адрес, куда передается управление.

[метка]

Символический адрес макрокоманды GINIT или GBFLM, управляющий сегментом программы графических приказов.

Адрес перехода содержится в третьем и четвертом байтах приказа.

Макрокоманда GTDD. Передает управление по указанному адресу буфера, если дисплей запомнил обнаружение светового пера, иначе приказ GTDD не выполняет никакой операции. Формат операндов:

адрес

Символический или абсолютный буферный адрес, куда передается управление.

[метка]

Символический адрес макрокоманды GINIT или GBFLM, управляющей сегментом программы графических приказов.

Макрокоманда GTND. Передает управление по указанному адресу буфера (если обнаружения не было) несмотря на то, что все условия для обнаружения выполнены. Формат операндов:

адрес

Символический и абсолютный буферный адрес, куда передается управление. Если световое перо обнаружено, то перехода не происходит, но состояние обнаружения сбрасывается.

[метка]

Символический адрес макрокоманды GINIT или GBFLM, управляющей сегментом программы графических приказов.

Макрокоманда GSXY. Запоминает содержимое регистров X и Y в указанной ячейке буфера. Формат операндов:

адрес

Символический или абсолютный буферный адрес первого из четырех смежных байт на четной границе.

[метка]

Символический адрес макрокоманды GINIT или GBFLM, управляющей сегментом программы графических приказов.

Макрокоманда GMVA. Запоминает в указанной ячейке буфера адресную константу. Формат операндов:

адрес1

Символический или абсолютный буферный адрес, куда засылается адресная константа.

[метка1]

Символический адрес макрокоманды GINIT или GBFLM, которая управляет сегментом программы графической программы, содержащим адрес1.

$BADDR = (\text{адрес2}[\text{метка2}])$

где адрес2 — символический или абсолютный буферный адрес, который как константа помещается в пятый и шестой байты приказа GMVA;

метка2 — символический адрес макрокоманды GINIT или GBFLM, управляющей сегментом программы графических приказов, содержащим адрес2.

Макрокоманда GMVD. Запоминает 2 байта данных в указанной ячейке буфера. Формат операндов:

адрес

Символический или абсолютный буферный адрес полуслова, куда засылаются данные.

[метка]

Символический адрес макрокоманды GINIT или GBFLM, управляющей сегментом программы графических приказов.

$BDATA = \text{шестнадцатеричное значение}$

Число в диапазоне 0000—FFFF.

Макрокоманда GNOP2. 2-байтовый приказ GNOP2 переводит дисплей в режим отсутствия операции до обнаружения следующего приказа на четной границе. Поле операндов пусто.

Макрокоманда GNOP4. 4-байтовый приказ GNOP4 переводит дисплей в режим отсутствия операции до обнаружения следующего приказа на четной границе. Формат операндов:

адрес

Символический или абсолютный буферный адрес, помещаемый в третий и четвертый байт приказа.

[метка]

Символический адрес макрокоманды GINIT или GBFLM, управляющей сегментом программы графических приказов.

Макрокоманда GECV (только для EC-7905). Вычерчивает символ с переменным интервалом, не зависящим от размера символов, и установленным ранее приказом GLVS. Формат операндов:

[размер]

В или BASIC — основной размер символа (принимается по умолчанию);

L или LARGE — увеличенный размер символа.

Макрокоманда CLVS (только для EC-7905). Определяет интервал между символами, созданными в режиме GECV. Формат операндов:

х-целое

Десятичное целое число от —64 до +64, определяющее горизонтальное расстояние между центрами символов в растровых единицах.

у-целое

Десятичное целое число от —64 до +64, определяющее вертикальное расстояние между центрами символов в растровых единицах.

Макрокоманда GMLD (только для EC-7905). Устанавливает яркость линий. Формат операндов:

яркость

В или BASIC — нормальный режим яркости (принимается по умолчанию);

L или LIGHT — повышенный режим яркости.

Макрокоманда GACC (только для EC-7905). Складывает двухбайтовую константу, определенную операндом BDATA, со значением, которое выбрано по адресу, определенному операндом «адрес». Результат записывается по адресу первого операнда. Формат операндов:

адрес

Буферный адрес четного байта данных:

[метка]

Символический адрес макрокоманды GINIT или GBFLM, управляющей сегментом программы графических приказов:

BDATA == шестнадцатеричное-число

Число в диапазоне 0000—FFFF.

Макрокоманда GTSL (только для EC-7905). Осуществляет переход с возвратом к подпрограмме. Формат операндов:
адрес

Буферный адрес четного байта подпрограммы. Первый приказ этой подпрограммы должен быть четырехбайтовым приказом перехода. Последний приказ подпрограммы должен передавать управление первому приказу.

[,метка]

Символический адрес макрокоманды GINIT или GBFLM, управляющей сегментом программы графических приказов.

Макрокоманда GLIC (только для EC-7905). Пересылает два байта, содержащих буферный адрес, для загрузки в 16-байтовый регистр итерации. Формат операндов:

адрес

Адрес четного байта буферной памяти для загрузки в регистр итераций.

[,метка]

Символический адрес макрокоманды GINIT или GBFLM, управляющей сегментом программы графических приказов.

Макрокоманда GSIC (только для EC-7905). Запоминает содержимое регистра итераций (два байта) в ячейке буфера. Формат операндов:

адрес

Адрес четного байта буфера, где запоминается содержимое регистра итераций.

[,метка]

Символический адрес макрокоманды GINIT или GBFLM, управляющей сегментом программы графических приказов.

Макрокоманда GTZE (только для EC-7905). Передает управление, если содержимое регистра итераций равно 0. Если значение регистра не равно нулю, то оно уменьшается на единицу. Формат операндов:

адрес

Адрес четного байта буфера, куда передается управление.

[,метка]

Символический адрес макрокоманды GINIT или GBFLM, управляющей сегментом программы графических приказов.

Макрокоманда GTNZ (только для EC-7905). Передает управление, если содержимое регистра итераций не равно 0. Кроме того, в этом случае значение регистра уменьшается на единицу. Формат операндов:

адрес

Адрес четного байта буфера, куда передается управление.

[метка]

Символический адрес макрокоманды GINIT или GBFLM, управляющей сегментом программы графических приказов.

Макрокоманда GCXY (только для ЕС-7905). Сравнивает содержимое регистров координат X и Y с четырьмя байтами информации буферной памяти. Формат операндов:

адрес

Буферный адрес четного байта сравниваемых данных.

[метка]

Символический адрес макрокоманды GINIT или GBFLM, управляющей сегментом программы графических приказов.

Макрокоманда GTEQ (только для ЕС-7905). Передает управление в зависимости от результата выполнения предыдущего графического приказа GCXY. Формат операндов:

адрес

Буферный адрес четного байта, куда передается управление в случае успешного сравнения в предыдущем приказе GCXY.

[метка]

Символический адрес макрокоманды GINIT или GBFLM, управляющей сегментом программы графических приказов.

Макрокоманда GTNE (только для ЕС-7905). Передает управление в зависимости от результата предыдущего графического приказа GCXY. Формат операндов:

адрес

Буферный адрес четного байта, куда передается управление в случае несравнения в предыдущем приказе GCXY.

[метка]

Символический адрес макрокоманды GINIT или GBFLM, управляющей сегментом программы графических приказов.

Макрокоманда GLXY (только для ЕС-7905). Выводит на экран ЭЛТ маркер, который можно перемещать по экрану с помощью координатного указателя или планшета. После выполнения приказа регистры координат X и Y положения луча примут значения, равные содержимому регистра метки. Поле операндов пусто.

Макрокоманда GEOS. Прекращает дальнейшее выполнение буферных приказов, вызывает установку битов ВНИМАНИЕ и СБОЙ В УСТРОЙСТВЕ в байте основного состояния и бита КОНЕЦ ПОСЛЕДОВАТЕЛЬНОСТИ ПРИКАЗОВ в первом байте уточненного состояния.

11.5.5. Макрокоманды образования данных

Макрокоманда GDV. Создает байты графических данных, управляющих положением луча. Формат данных зависит от приказа, предшествующего данным. Формат операндов:

X-значение, Y-значение

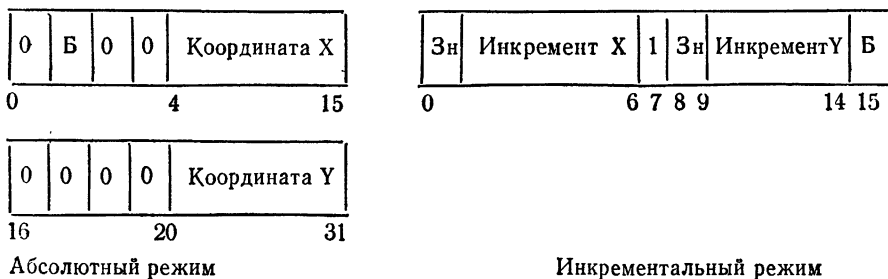
Указывает значения или инкременты координат X и Y тремя способами:

- 1) десятичные числа со знаком или без знака;
- 2) $*+n$ или $*-n$ (только для абсолютного режима), здесь * означает текущее значение счетчика координаты X или Y, n — десятичное целое число;
- 3) символы, которые определяются оператором EQU

луч

В или BLANK — выключение луча, U или UNBL — включение луча.

Для приказа абсолютного режима создаются 4 байта данных. Значение X и значение Y должны быть в диапазоне 0—4095. Для приказа инкрементального режима создаются 2 байта данных. Значение X и значение Y должны иметь в диапазоне от —255 до +255 (см. рис. 11.1).



Б=0 — луч включен
Б=1 — луч выключен

Рис. 11.1 Формат данных макрокоманды GDV

Макрокоманда GCNOP. Образует нулевой знак или знаки. Формат операндов:

[целое-число|C]

где целое-число — определяет число нулевых знаков (X'00'). Если операнд опущен, принимается значение 1;

C — приводит к образованию нулевого знака только в том случае, если значение счетчика буферного адреса является нечетным.

Макрокоманда GCNL. Создает требуемое число знаков новой строки. Формат операнда:

[целое-число]

Определяет число знаков новой строки. Если этот операнд опущен, то принимается значение 1.

Символические счетчики координат X и Y изменяются макрокомандой GCNL, отражая новое положение луча.

Макрокоманда GTXT. Генерирует команду Ассемблера DC, определяющую знаковую константу. Формат операндов:

знаки

Любая комбинация знаков стандартного знакового набора, заключенная в апострофы. Для длинных знаковых констант можно использовать карты продолжения.

целое число

Число пар апострофов (") или знаков амперсанда (&&) в цепочке знаков.

Макрокоманда GDCDS. Определяет константы или резервирует пространство буфера. Формат операндов:

$$\left\{ \begin{array}{l} C,n[.метка] \\ S,n \end{array} \right\}$$

где C — определяет константу;

S — резервирует пространство;

n — десятичное целое число (после C и S), указывающее константу длиной в полуслово (C) или число байт резервируемого пространства (S), или символический адрес (после C), из которого должна быть образована 2-байтовая константа;

метка — символический адрес макрокоманды GINIT или GBFLM, управляющей сегментом программы графических приказов.

11.5.6. Макрокоманды, блоки и программы управления областью вывода графических данных

Макрокоманда OACB. Строит блок управления областью вывода OACB. Формат блока OACB:

0	SLOA	адрес области вывода графических данных GDOA
4	LOA	длина области вывода GDOA (минимум 4 байта)
8	AORP	адрес программы переполнения
12	CRSA	начальный адрес текущей программы графических приказов
16	OLP	адрес следующего приказа
20	BLP	точка загрузки буфера

Макрокоманда OACB имеет следующий формат операндов:
начальный адрес

Адрес ячейки первого байта GDOA.

длина

Длина GDOA в байтах.

переполнение

Адрес программы переполнения.

Поля CRSA и OLP первоначально устанавливаются равными SLOA.

Поле BLP первоначально устанавливается равным 0.

Стандартная программа накопления графических приказов GSTOR. Накапливает в области вывода данных GDOA графические приказы и данные. Программа вызывается макрокомандой CALL или LINK со следующими операндами:

GSTOR

Точка входа стандартной программы накопления графических приказов,

адрес-осбр

Символический адрес указателя блока управления выводом. Указатель состоит из двух слов. Первое слово содержит адрес OACB, второе — адрес рабочей области, определяемой программистом. Если программа включает проблемно-ориентированную программу, то размер рабочей области — 100 полных слов. Если программа использует только стандартную программу GSTOR, то достаточно 16 полных слов.

таблпар

Символический адрес таблицы параметров, состоящей из двух слов. Первое слово содержит адрес первого байта графических данных, запоминаемых в GDOA, второе — число запоминаемых байт графических данных.

Программа оставляет последние 4 байта GDOA в резерве, и если делается попытка записать данные в эти 4 байта, то вызывается программа переполнения.

Коды возврата: 20 — число запоминаемых байт равно или меньше 0; 24 — GSTOR передала управление программе переполнения, которая не скорректировала OLP в ОСБР.

Макрокоманда GODEL. Строит таблицу параметров для программы GSTOR. Формат операндов:

начальный-адрес

Адрес первого байта графического приказа или графических данных.

адрес-параметров

Адрес области из двух слов, в которой должна быть построена таблица параметров.

Макрокоманда GUSTOR. Безусловно запоминает графические данные или приказы в GDOA и корректирует OLP в OACB. Формат операндов:

адрес-оас

Адрес блока управления ОАСВ.

начальный-адрес

Адрес запоминаемых графического приказа или данных.

{2|4}

Число запоминаемых данных.

11.5.7. Проблемно-ориентированные программы

Рассматриваются вопросы, связанные с работой проблемно-ориентированных программ.

Масштабирование. Определяется таблицей XYLM, строящейся следующим образом (на границе слова):

Смещение в байтах	Длина	Значение	
0	2	X1	} координаты нижнего левого угла кадра на экране ЭЛТ
2	2	Y1	
4	2	X2	} координаты верхнего правого угла кад- ра на экране ЭЛТ
6	2	Y2	
8	4	U1	} координаты нижнего левого угла масштабируемого прямоугольника
12	4	U2	
16	4	V1	} координаты верхнего правого угла масштабируемого прямоугольника
20	4	V2	

Коды возврата (десятичные) для проблемно-ориентированных программ:

4 — X1, Y1, X2 или Y2 вне диапазона 0—4095 или $X1 \geq X2$, $Y1 \geq Y2$;

8 — значения U, V дают X или Y вне диапазона 0—4095;

12 — характеристика — 0, мантисса — ненулевая для числа с плавающей запятой;

16 — $U1 \geq U2$; $V1 \geq V2$; $DX1 \geq DX2$; $DY1 \geq DY2$;

20 — неправильно указаны входные параметры;

24 — программа переполнения не скорректировала OLP;

28 — Э/к вариант = D (см. ниже);

32 — Э/к вариант = E или C (см. ниже);

36 — ошибка в указании установки луча для программы GSDPLT;

40 — ошибка во входных параметрах. Обработка продолжается;

44 — Э/к вариант = A или C (см. ниже). Обработка продолжается;

48 — Э/к вариант = B (см. ниже). Обработка продолжается.

Варианты выхода за пределы экрана и кадра (Э/к варианты):

A — точки, векторы или знаки образуются в пределах кадра;

B — точки, векторы или знаки образуются в пределах экрана;

C — точки, векторы или знаки образуются в пределах кадра до момента, когда имеет место выход за пределы экрана;

- D — точки, векторы или знаки образуются в пределах кадра до момента, когда имеет место выход за пределы кадра;
 E — точки, векторы или знаки образуются в пределах экрана до момента, когда имеет место выход за пределы экрана.

Вызов проблемно-ориентированных программ. Программы (кроме PENTRK) вызываются макрокомандой CALL или LINK со следующими операндами:

точка-входа

Название программы — точка входа в программу.

адрес-осбр

Символический адрес ОСБР.

таблпар

Символический адрес таблицы параметров. Таблица параметров описана ниже для каждой программы в отдельности.

Программа GCPRNT. Воспроизводит алфавитно-цифровые знаки. Таблица параметров для программы GCPRNT выглядит следующим образом:

Смещение	Длина	Значение
0	4	Адрес таблицы масштабирования XYLIM для определения местоположения первого знака
4	4	Uc } координаты начальной точки на экране
8	4	Vc } дисплея
12	1	Формат: B — с фиксированной запятой F — с плавающей запятой
13	1	Масштабирование: S — выполняется N — не выполняется
14	1	Размер знаков: B — основной L — большой
15	1	Режим знаков: P — защищенный F — незащищенный
16	4	Адрес области, содержащей последовательности знаков
20	2	Число воспроизводимых знаков, включая все знаки новой строки
22	1	G — воспроизведение должно начинаться в точке текущего положения луча, U — воспроизведение должно начинаться в точке с координатами Uc, Vc

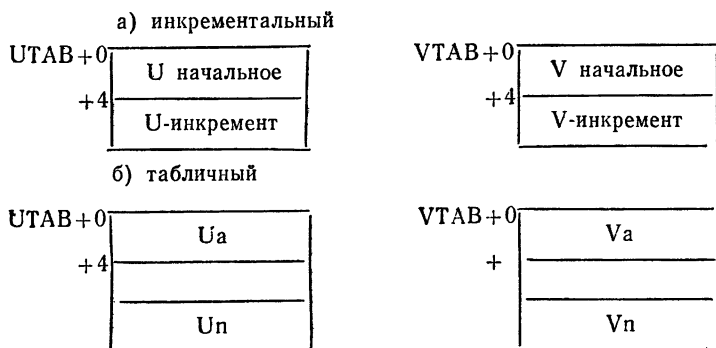
Программы GSVPLT/GSPLOT. Изображает графики: GSVPLT — рисует непрерывный график, GSPLOT — пунктирный. Таблица параметров для программ GSVPLT/GSPLOT выглядит следующим образом:

Смещение	Длина	Значение
0	4	Адрес таблицы масштабирования XYLIM
4	4	UTAB }
8	4	VTAB } Адреса таблиц UTAB и VTAB (см. ниже)
12	1	Формат
13	1	Масштабирование
14	1	Размер знаков
15	1	Режим знаков

см. выше

16	1	Абсолютная форма: А — абсолютные входные значения Р — относительные входные значения
17	1	Инкрементальная форма: W — ни U, ни V не являются инкрементальными X — массив U является инкрементальным Y — массив V является инкрементальным Z — оба массива являются инкрементальными
18	1	Тип графика А — точечный В — знаковый С — векторный D — векторно-точечный Е — векторно-знаковый
19	1	Вариант выхода за границы (А, В, С, D или Е)
20	2	Число пар (U, V)
22	1	Код воспроизводимого знака
23	1	Перекрытие знаков: <u>0</u> — разрешено, N — не разрешено

Таблицы **UTAB** и **VTAB** задают входные значения и имеют следующий формат:



Программа GCGRID. Изображает декартовую сетку. Таблица параметров:

Смещение	Длина	Значение
0	4	Адрес таблицы масштабирования XYLIM
4	4	U _i } интервалы сеток
8	4	
12	1	Формат } см. выше
13	1	
14	1	
		Тип координатной сетки:
		А — линейная по X и Y
		В — логарифмическая по X и линейная по Y
		С — логарифмическая по Y и линейная по X
		D — логарифмическая по X и Y
15	1	Резервируется

Программа GLABEL. Размечает оси координат. Таблица параметров:

Смещение	Длина	Значение
0	4	Адрес таблицы масштабирования XYLM
4	4	U _i или V _i — X- и Y-координатный интервалы
8	4	U _c или V _c — координата оси
12	1	Формат
13	1	Масштабирование
14	1	Размер знаков
15	1	Ось координат: X или Y
16	4	Адрес первого байта непрерывной области основной памяти, содержащей последовательность знаков для разметки оси
20	2	Число знаков в обозначении
22	2	Интервал в растровых единицах между обозначением и осью
24	2	Длина метки. Если метка не требуется, то должен быть 0
26	2	Резервируется

Программы GVARC/GARC. Образуют дуги окружности. Программа GVARC аппроксимирует дуги векторами, программа GARC — точками. Таблица параметров:

Смещение	Длина	Значение
0	4	Адрес таблицы масштабирования XYLM
4	4	U _c
8	4	V _c
12	1	Формат
13	1	Масштабирование
14	1	Тип приращения: D — плотность; A — угол
15	1	Вариант выхода за границы (по умолчанию E)
16	4	Радиус кривизны
20	2	Начальный угол в градусах
22	2	Угол в градусах или приблизительное расстояние (плотность) между последовательными точками дуги
24	2	Размер дуги в градусах
26	2	Резервируется

Программы GPVGRD/GPGRID. Образуют сетки полярных координат. Программа GPVGRD аппроксимирует окружности векторами, программа GPGRID — точками. Таблица параметров:

Смещение	Длина	Значение
0	4	Адрес таблицы масштабирования
4	4	V _c
8	4	U _c
12	1	Формат
13	1	Масштабирование
14	1	Тип приращения (см. выше)
15	1	Вариант выхода за границы (по умолчанию E)
16	4	Приращение по радиусу
20	2	NR — число радиусов. Для GPGRID — 0, 1, 2, 4 или 8, для GPVGRD — 360/NR должно быть целым числом

22	2	Плотность/угол (см. выше)
24	4	Максимальная длина радиуса
28	1	Тип полярной сетки: A — линейная, B — полулогарифмическая

Программа GSDPLT. Наносит данные. Таблица параметров:

Смещение	Длина	Значение
0	4	Адрес специальной таблицы XYSCALE (см. ниже)
4	4	UADR} адрес полей или таблиц, содержащих координаты X и Y
8	4	VADR}
12	1	Формат: A — U и V в формате с плавающей запятой B — U и V — целые числа C — U — целое число, V — значение с плавающей запятой D — U — значение с плавающей запятой, V — целое число
13	(1)	Входной формат U A — все значения U абсолютные B — первое значение U абсолютное, каждое последующее — инкремент к предыдущему C — первое значение абсолютное, каждое последующее образуется повторным прибавлением поля U-инкремент (см. ниже) к предыдущему значению
14	1	Входной формат V (то же, что и для входного формата U)
15	1	Тип изображения и режим: A — абсолютный векторный B — абсолютный точечный C — инкрементальный векторный D — инкрементальный точечный E — оптимальный векторный график F — оптимальный точечный график
16	1	Установка луча: A — начальное положение луча определяется первой парой (U, V) B — первая пара (U, V) определяет текущее положение луча C — первая пара (U, V) определяет текущее положение луча, вторая — куда должен быть помещен луч
17	1	Вариант выхода за границы (по умолчанию E)
18	1	Масштабирование: A — не выполняется, B — выполняется
19	1	Установка режима: A — должен быть образован приказ установки режима, B — режим уже установлен
20	1	Вариант сегмента: A — каждая пара (U, V) — точка или конечная координата вектора, B — каждая двойка пар (U, V) — начальная и конечная точка линейного сегмента
21	1	Резервируется
22	2	Число пар (U,V)
24	4	} целые числа, используемые для указания каждого последующего элемента в массивах U и V
28	4	
32	4	} целое число, прибавляемое к предшествующему значению U или V
36	4	

Таблица XYSCALE:

Смещение	Значение (длина всех элементов равна 4)	
0	KX1	Координаты нижнего левого и верхнего правого углов прямоугольного кадра области воспроизведения
4	KY1	
8	KX2	
12	KY2	
16	ЭХ1	Координаты нижнего левого и верхнего правого углов экрана. Величины должны быть в диапазоне 0—4095
20	ЭY1	
24	ЭХ2	
28	ЭY2	
32	ИХ1	Координаты нижнего левого и верхнего правого углов прямоугольника, который должен быть изображен в кадре, определенном выше
36	ИY1	
40	ИХ2	
44	ИY2	
48	Ключ-изменения X устанавливается в 0 при всяком изменении любой координаты X в данной таблице. При масштабировании первоначально устанавливается в 0	
52	Ключ-изменения Y используется аналогично ключу-изменения X	
56	Резервируется для GSDPLT	
58	Резервируется для GSDPLT	

Программа PENTRK. Отслеживает световое перо. Предназначена для вычерчивания с помощью светового пера и вызывается макрокомандой CALL или LINK с операндами:

PENTRK

Точка входа в программу.

адрес-осбр

Символический адрес ОСБР.

В результате в GDOA строится буферная программа слежения за световым пером. Обращение к этой программе можно осуществить с помощью приказов:

GMVA TRBUF+2,BADDR=#+6 определяет адрес возврата
GTRU TRBUF+4 передает управление программе слежения

где TRBUF — адрес точки входа в буферную программу, помещаемый программой PENTRK в сотое слово рабочей области.

При выходе из буферной программы дисплей будет в режимах GENSD и GDRD.

Диалоговый удаленный ввод заданий (ДУВЗ)

12.1. Общие сведения

Диалоговый удаленный ввод заданий ОС ЕС — это средство для ввода в ЭВМ программ, данных и заданий, редактирования их в диалоговом режиме и передачи заданий в пакетную обработку с последующим получением результатов на абонентском пункте. В ОС ЕС издания 4.1 обеспечивается работа с абонентским пунктом АП-70 (в модификации 4 издания 4.1 — и с ЕС-7906), в издании 6.1 дополнительно обеспечивается работа с АП-1 (НРБ), с локальными дисплеями ЕС-7906 (СССР), ЕС-7061, ЕС-7063 (ВНР), BSS (ГДР), а также с консолями оператора (до 3 штук). Кроме того, в модификации 1 издания 6.1 обеспечивается локальная станция ЕС-7920. Далее все устройства, обеспечиваемые в ДУВЗ, будем называть абонентскими пунктами (АП).

Диалоговый удаленный ввод заданий работает как системная задача в отдельной зоне основной памяти, обслуживая одновременно в режиме разделения времени нескольких абонентов. Задания, переданные в пакетную обработку, выполняются в фоновых зонах памяти. Абонент может запросить во время ввода или редактирования программы на языках Фортран и ПЛ/1 синтаксический анализ операторов программы.

12.2. Генерация ДУВЗ

12.2.1. Требования к операционной системе

Прежде чем генерировать конкретный вариант ДУВЗ, необходимо убедиться, что исходная операционная система ОС ЕС содержит необходимые программные средства. Другими словами, исходная операционная система должна быть сгенерирована с учетом следующих условий:

1. Абонентские пункты для использования ДУВЗ должны быть описаны макрокомандами IOCTRL, IODEVICE.

2. Если используются АП-1 ЕС-7920 или АП-70, то в макрокоманде DATAMGT должен быть указан BTAM, а если ЕС-7906,

EC-7061, EC-7063 или BSS, то необходима макрокоманда GRAPHICS.

3. Если в ДУВЗ используются консоли оператора, то необходимы макрокоманды SECONSLE (до 3) с маршрутными кодами 13, 14 и 15.

4. В макрокоманде SCHEDULR необходимо указать OPTIONS=CRJE и, если используются консоли оператора, CONOPTS=MCS.

5. В макрокоманде SUPRVSOR необходимо задать TIMER=INTERVAL или TIMER=JOBSTEP, а в системе MFT — еще OPTIONS=ATTACH.

6. Если ДУВЗ использует синтаксические анализаторы, то необходима макрокоманда CHECKER.

7. Для АП-1, АП-70 или EC-7920 необходима макрокоманда TELCMLIB.

8. Необходима макрокоманда MACLIB.

12.2.2. Макрокоманды генерации ДУВЗ

Для генерации ДУВЗ необходимо написать следующие макрокоманды:

CRJELINE — для каждого абонентского пункта;

CRJETABL — одну для всех пользователей;

CRJEUSER — одну или несколько для всех пользователей;

CRJEDEF — одну для всей системы (только в издании 6.1).

Макрокоманда CRJELINE. Формат:

```
[метка] CRJELINE TYPE=тип,DDSYSIN=имяdd,DDLNE=имяdd
[ ,RLN= { число } ] [ ,LERB= [ число1 ] , [ число2 ] ,
[ число3 ] , [ число4 ] ] [ ,ADDR=xx ] [ ,FEATURE=
= ([DIAL],[INTERRUPT]) ] [ ,ONLNT= {NO/YES}
[ ,MODE=PAGE/ROLL ]
```

где TYPE=тип — тип абонентского пункта в соответствии со следующей таблицей:

Код устройства	TYPE=
EC-8570 (АП-70)	8570
EC-8501 (АП-1)	8501
EC-7066	7066 или 7066-1 (экран 12×80) 7066-2 (экран 6×80) 7066-3 (экран 12×40) 7066-4 (экран 6×40) 7061 или 7061-1 (экран 12×80) 7061-2 (экран 16×64)
EC-7063	7063 или 7063-1 (экран 12×80) 7063-2 (экран 16×64)
BSS	7907
Консоль оператора	WTO
EC-7920	7920

Примечание. В издании 4.1 допустим только TYPE=8570, в модификации 4 издания 4.1 допустимы только TYPE=8570 и TYPE=7066;

DDSYSIN=имя dd — имя оператора DD в процедуре запуска ДУВЗ, описывающего набор данных для входного потока заданий;

DDLINE=имя DD — имя оператора DD в процедуре запуска ДУВЗ, описывающего абонентский пункт;

RLN={число
1} — относительный номер канала связи для АП-1, ЕС-7920 или АП-70;

LERB=([число1
255] , [число2
10] , [число3
5] , [число4
5]) —

указывает пороговые счетчики для блока LERB: 1 — количество передач, 2 — количество ошибок в данных, 3 — количество ошибок ТРЕБУЕТСЯ ВМЕШАТЕЛЬСТВО, 4 — количество ошибок по таймауту;

ADDR=xx — шестнадцатеричный байт в коде передачи, указывающий адрес АП-1; для других АП не кодируется;

FEATURE=([DIAL] , [INTERRUPT]) — DIAL указывает коммутируемый канал связи для АП-1 или АП-70; INTERRUPT указывает, что АП-70 оборудован средством прерывания;

ONLNT={NO|YES} — задает средства неавтономного тестирования для АП-1 или АП-70.

MODE={PAGE|ROLL} — PAGE указывает, что для локальных дисплеев вывод происходит в режиме страниц, ROLL указывает режим свертки.

По умолчанию для всех дисплеев, кроме BSS (ГДР) и ЕС-7920, принимается режим PAGE.

Макрокоманда CRJETABL. Формат:

метка CRJETABL JOB=число,USERS=число,SYSCRJE=символ
[JOBEXIT=имя][ONEXIT=имя][OFFEXIT=имя]
[,BUFNO={число
1}][,MSGNO={число
100}]
[,BRDCST={число
100}][,OUTNO={число
10}]
[,MSGRC={число
8}]
[,ALIAS=(имя,дополн-имя,...)]
[,USRMCMMD=(команда,...)]
[,USRSCMD=(подкоманда,...)]
[,CMDEXIT=имя]
[,PLILNO={число
2}][,FORTLNO={число
2}]

где JOB=число — указывает максимальное число (от 1 до 999) заданий, одновременно находящихся в системе ДУВЗ;

USERS=число — указывает максимальное число (от 1 до 9999) абонентов ДУВЗ;

SYSCRJE=символ — указывает выходной класс для заданий ДУВЗ;

JOBEXIT=имя — указывает программу обработки операторов заданий;

ONEXIT=имя — указывает программу обработки LOGON;
OFFEXIT=имя — указывает программу обработки LOGOFF;
BUFNO= $\left\{ \begin{array}{c} \text{число} \\ \underline{1} \end{array} \right\}$ — указывает максимальное число (от 1 до

100) одновременно выполняемых команд оператора ЭВМ;

MSGNO= $\left\{ \begin{array}{c} \text{число} \\ \underline{100} \end{array} \right\}$ — указывает максимальное число (от 1 до

999) задержанных сообщений;

BRDCST= $\left\{ \begin{array}{c} \text{число} \\ \underline{100} \end{array} \right\}$ — указывает максимальное число (от 1 до 100) циркулярных сообщений;

OUTNO= $\left\{ \begin{array}{c} \text{число} \\ \underline{10} \end{array} \right\}$ — указывает число строк (от 1 до 65 535), посылаемых на АП-1 или АП-70 группой, если они не имеют средств прерывания;

MSGRC= $\left\{ \begin{array}{c} \text{число} \\ \underline{8} \end{array} \right\}$ — указывает маршрутный код (от 1 до 16) для сообщений пользователя, направляемых оператору ЭВМ;

ALLIAS=(имя, дополн-имя,...) — указывает список команд или подкоманд с дополнительными именами. Дополнительное имя имеет от 1 до 8 символов, первый из которых должен быть буквой;

USRMCMC=(команда, ...) — указывает список дополнительных команд;

USRSCMD=(подкоманда, ...) — указывает список дополнительных подкоманд EDIT;

CMDEXIT=имя — указывает программу обработки всех дополнительных команд и подкоманд; операнд обязателен, если задан операнд USRMCMC или USRSCMD;

PL1LNO= $\left\{ \begin{array}{c} \text{число} \\ \underline{2} \end{array} \right\}$ — указывает максимальное число (от 0 до 37) строк продолжения для оператора языка ПЛ/1;

FORTLNO= $\left\{ \begin{array}{c} \text{число} \\ \underline{2} \end{array} \right\}$ — указывает максимальное число (от 0 до 19) строк продолжения для операторов языка Фортран.

Макрокоманда CRJEUSER. Формат:

[метка] CRJEUSER [идплз,пароль,...]

где идплз, пароль — указывает идентификатор абонента (от 1 до 7 символов) и его пароль (от 1 до 8 символов); в одной макрокоманде можно задать до 100 пар идентификаторов и паролей в алфавитном порядке. Первый символ идентификатора или пароля должен быть буквой.

Макрокоманда CRJEDEF (только издание 6.1).

Используется при необходимости изменения некоторых параметров системы, принимающих обычно значения по умолчанию. Формат:

[метка] CRJEDEF

$$\left[\text{STLNO} = \left\{ \begin{array}{c} \text{число} \\ 10 \end{array} \right\} \right] \left[\text{,INCR} = \left\{ \begin{array}{c} \text{число} \\ 10 \end{array} \right\} \right]$$

$$\left[\text{,MSGID} = \left\{ \begin{array}{c} \text{YES} \\ \text{NO} \end{array} \right\} \right] \left[\text{,LOGBC} = \left\{ \begin{array}{c} \text{YES} \\ \text{NO} \end{array} \right\} \right]$$

$$\left[\text{,CONT} = \left\{ \begin{array}{c} \text{HERE} \\ \text{BEGIN} \\ \text{NEXT} \end{array} \right\} \right]$$

$$\left[\text{,EDIT} = \left\{ \begin{array}{c} \text{OLD} \\ \text{NEW} \end{array} \right\} \right] \left[\text{,SEQNO} = \left\{ \begin{array}{c} \text{YES} \\ \text{NO} \end{array} \right\} \right]$$

$$\left[\text{,SCAN} = \left\{ \begin{array}{c} \text{YES} \\ \text{NO} \end{array} \right\} \right] \left[\text{,PL1} = \left\{ \begin{array}{c} \text{C48} \\ \text{C60} \end{array} \right\} \right]$$

$$\left[\text{,PL1MARG} = \left(\left\{ \begin{array}{c} 1 \\ 2 \end{array} \right\}, \left\{ \begin{array}{c} r \\ 72 \end{array} \right\} \right) \right]$$

$$\left[\text{,ATTR} = \{ \text{DATA} | \text{TEXT} | \text{FRTOP} | \text{FRTST} | \text{FORTB} | \text{PL1} | \text{DSLST} | \text{CLIST} \} \right]$$

$$\left[\text{,EXEC} = \left\{ \begin{array}{c} \text{LIST} \\ \text{NOLIST} \end{array} \right\} \right] \left[\text{,SEND} = \left\{ \begin{array}{c} \text{NOW} \\ \text{LOGON} \end{array} \right\} \right]$$

$$\left[\text{,LIST} = \left\{ \begin{array}{c} \text{NUM} \\ \text{NONUM} \end{array} \right\} \right] \left[\text{,INPUT} = \left\{ \begin{array}{c} \text{PROMPT} \\ \text{NOPROMPT} \end{array} \right\} \right]$$

$$\left[\text{,MINLINE} = \left\{ \begin{array}{c} \text{длина} \\ 50 \end{array} \right\} \right] \left[\text{,MAXLRECL} = \left\{ \begin{array}{c} \text{длина} \\ 120 \end{array} \right\} \right]$$

$$\left[\text{,WAIT} = \left\{ \begin{array}{c} \text{число} \\ 250 \end{array} \right\} \right] \left[\text{,OUTPUT} = \text{SMMSG} \right]$$

где STLNO = {число|10} — указывает номер начальной строки для нового файла данных;

INCR = {число|10} — указывает приращение, с которым будет образовываться номер вводимой записи;

MSGID = {YES|NO} — YES указывает, что сообщения, направляемые на абонентский пункт, будут иметь префикс вида IHKnnn;

LOGBC = {YES|NO} — YES указывает, что абоненту в момент подключения автоматически посылаются циркулярные сообщения;

CONT = {HERE|BEGIN|NEXT} — указывает подразумеваемый операнд команды CONTINUE, если она вводится без операндов;

EDIT = {OLD|NEW} — указывает характеристику набора данных по умолчанию, если она не указана в команде EDIT;

SEQNO = {YES|NO} — YES указывает, что по умолчанию команда EDIT присваивает входному набору данных характеристику SEQ, т. е. номер строки будет занимать колонки 73—80 каждой записи; NO указывает характеристику NOSEQ;

SCAN = {YES|NO} — YES указывает, что по умолчанию команда EDIT будет запрашивать синтаксический анализ для входного набора данных;

PL1 = {C48|C60} — указывает алфавит языка ПЛ/1, если он не указан в команде EDIT;

PL1MARG = $\left(\left\{ \begin{array}{c} 1 \\ 2 \end{array} \right\}, \left\{ \begin{array}{c} r \\ 72 \end{array} \right\} \right)$ — указывает левую и правую границы поля оператора для языка ПЛ/1, если они не указаны в команде EDIT;

ATTR=характеристика — указывает характеристику входного набора данных, если она не указана в команде **EDIT**;

EXEC=**{LIST|NOLIST}** — указывает, нужно ли распечатывать каждую выполняемую команду, если это не указано в команде **EXEC**;

SEND=**{NOW|LOGON}** — **NOW** указывает, что посылаемые командой **SEND** сообщения должны сразу распечатываться на принимающем конце; если адресат неактивен, то они теряются; **LOGON** указывает, что сообщения должны сохраняться и отсылаться, когда абонент подключится к системе;

LIST=**{NUM|NONUM}** — **NUM** указывает, что по команде **LIST** записи распечатываются вместе с номерами;

INPUT=**{PROMPT|NOPROMPT}** — **PROMPT** указывает, что для нового набора данных перед вводом каждой строки будет выдаваться ее номер;

MINLINE=**{длина|50}** — указывает минимальную длину выходной строки, которую можно определить командой **VARY**;

MAXLRECL=**{длина|120}** — указывает максимальную длину (от 120 до 144) записи в наборе данных;

WAIT=**{число|250}** — указывает интервал между двумя последовательными выводами на дисплей в режиме свертки в сотых долях секунды (от 50 до 850);

OUTPUT=**MSG** — указывает, что выходные результаты заданий будут сопровождаться сообщениями планировщика, если это не указано в команде **OUTPUT**.

12.2.3. Программные выходы

Системный программист может добавить в конкретную систему ДУВЗ дополнительные программы, получающие управление в определенных ситуациях. Программы должны быть реентерабельными и сохранять регистры в области сохранения, которую обеспечивает ДУВЗ. Адрес области сохранения содержится в регистре 13. Регистр 1 содержит адрес списка параметров.

Программный выход по LOGON. Точка входа программы выхода по **LOGON** задается операндом **ONEXIT** макрокоманды **CRJETABL**. Список параметров состоит из адреса идентификатора абонента, дополненного до 7 символов, и адреса учетной информации, заданной в команде **LOGON**. Учетная информация имеет следующий формат:

Длина 1	Параметр 1	. . .	Длина n	Параметр n	X'FF'
---------	------------	-------	---------	------------	-------

где Длина *i* — байт, в который заносится число символов в Параметре *i*. Если параметр заключен в круглые скобки, то в бите 1 устанавливается единица. Если в скобки заключен последний (или единственный) параметр, то единица также заносится в бит 0;

Параметр *i* — параметр, заданный операндом АССТ, включая и заключающие его апострофы.

Если учетная информация отсутствует, то в первый байт области для учетной информации помещается X'FF'.

Коды возврата программы выхода (в регистре 15):

0 — абонент получает доступ к системе ДУВЗ;

4 — абонент не получает доступа к системе ДУВЗ и ему посылается соответствующее сообщение.

Программный выход по LOGOFF. Точка входа программы выхода по LOGOFF задается операндом OFFEXIT в макрокоманде CRJETABL. Список параметров содержит адрес идентификатора абонента, дополненного до 7 символов, адрес поля, в котором указан тип АП, и адрес поля, формат которого выглядит следующим образом:

Время подключе- ния чч, мм	05	Время отключения чч, мм	04	Полное время сеанса, мин	Нормальное (X'00') или аварийное (X'80') завершение
----------------------------------	----	-------------------------------	----	-----------------------------------	-----------------------------------------------------------

5 байт

1 байт

5 байт

1 байт

4 байта

1 байт

Тип задается одним байтом:

Издание 4.1.

X'44' — АП ЕС-8570 на некоммутируемом канале связи;

X'45' — АП ЕС-8570 на коммутируемом канале связи.

Издание 6.1.

X'00' — консоль оператора;

X'02' — АП ЕС-8501 на некоммутируемом канале связи;

X'03' — АП ЕС-8501 на коммутируемом канале связи;

X'04' — АП ЕС-8570 на некоммутируемом канале связи;

X'05' — АП ЕС-8570 на коммутируемом канале связи;

X'08' — АП ЕС-8570 со средствами прерывания;

X'84' — ЕС-7066;

X'88' — BSS (ГДР); ЕС-7061; ЕС-7063;

X'D0' — локальная станция ЕС-7920.

Программный выход для обработки операторов языка управления заданиями. Точка входа программы выхода задается операндом JOBEXIT в макрокоманде CRJETABL. Список параметров содержит адрес оператора JCL и адрес идентификатора абонента, дополненного до 7 символов.

Коды возврата программы выхода (в регистре 15):

0 — задание можно передать в пакетную обработку;

4 — задание нельзя передавать в пакетную обработку;

8 — задание нельзя передавать в пакетную обработку;

в регистре 1 возвращается адрес 60-байтового сообщения абоненту.

Программный выход для дополнительных команд. Точка входа программы выхода задается операндом CMDEXIT в макрокоманде CRJETABL. Команды и подкоманды должны быть определены операндами USRMCMD и USRSCMD в этой же макро-

команде. Команды или подкоманды не должны превышать в длину 105 символов и не должны иметь продолжения. Все команды и подкоманды обрабатываются одной реентерабельной программой выхода. Список параметров содержит адрес таблицы следующего формата:

0	Резервируется	
4	Резервируется	Адрес области связи
8	Идентификатор абонента	
12		Резервируется
16	Резервируется	Длина информации

где адрес области связи — это адрес области длиной 105 байтов, содержащей команду абонента;

длина информации — число символов в строке после редактирования (например, после удаления символа возврата на шаг).

Коды возврата программы выхода:

0 — обработка команды завершена;

4 — обработка команды завершена; абоненту посылается сообщение из области связи.

12.2.4. Генерация модулей ДУВЗ

Генерация модулей ДУВЗ производится следующим заданием:

//имя	JOB	Параметры
//STEP1	EXEC	ASMFC
//ASM.SYSPUNCH	DD	DSN=CRJEWORК(IHKMAC),
//		DISP=(,KEEP),
//		SPACE=(TRK,(12,1,1)), UNIT=50xx,
		VOL=SER=метка
//SYSIN	DD	* Макрокоманда CRJELINE Макрокоманда CRJETABL Макрокоманда CRJEDEF END
/*		
//STEP2	EXEC	ASMFC
//ASM.SYSPUNCH	DD	DSN=CRJEWORК(IHKUSR), DISP=OLD,
//		VOL=REF=*.STEP1.SYSPUNCH
//SYSIN	DD	* Макрокоманды CRJEUSER
/*		
//STEP3	EXEC	ASMFC
//ASM.SYSPUNCH	DD	DSN=CRJEWORК(имя-программы-выхо- да), DISP=OLD,
//		VOL=REF=*.STEP1.SYSPUNCH
//SYSIN	DD	* Программа выхода
/*		
//STEP4		Тот же формат
//STEP5		Тот же формат
//STEP6		Тот же формат

//STEP7	EXEC	PGM=LINKEDIT, PARM=(XREF, LIST, DC, NCAL, LET)
//SYSLMOD	DD	DSN=SYS1.TELCMLIB, DISP=OLD
//SYSLIB	DD	DSN=CRJEWOR, DISP=(OLD, DELETE),
//		VOL=REF=*.STEP1.SYSPUNCH
//SYSUTI	DD	Параметры для рабочего набора данных Редактора связей
//SYSPRINT	DD	SYSOUT=A
//SYSLIN	DD	*
	INCLUDE	SYSLIB (IHKMAC)
	NAME	IHKMAC(R)
	INCLUDE	SYSLIB (IHKUSR)
	NAME	IHKUSR(R)
	INCLUDE	SYSLIB (имя-программы-выхода)
	NAME	имя-программы-выхода (R)
	...	
/*		
//STEP8	EXEC	PGM=LINKEDIT, PARM=(XREF, LIST, DC)
//SYSLMOD	DD	DSN=SYS1.LINKLIB, DISP=OLD
//SYSLIB	DD	DSN=SYS1.TELCMLIB, DISP=OLD
//SYSUTI	DD	Параметры для рабочего набора данных Редактора связей
//SYSPRINT	DD	SYSOUT=A
//SYSLIN	DD	*
	INCLUDE	SYSLIB (IHKMAC)
	INCLUDE	SYSLIB (IHKAVT)
	INCLUDE	SYSLIB (IHKCCI)
	ENTRY	IHKBGN
	NAME	IHKBGN(R)
//STEP9	EXEC	LINKEDIT, PARM=(XREF, LIST, DC, NCAL, LET)
//		
//SYSLIB	DD	DSN=SYS1.TELCMLIB, DISP=OLD
//SYSLMOD	DD	DSN=SYS1.LINKLIB, DISP=OLD
//SYSUTI	DD	Параметры для рабочего набора данных Редактора связей
//SYSPRINT	DD	SYSOUT=A
//SYSLIN	DD	*
	INCLUDE	SYSLIB (IHKUSR)
	INCLUDE	SYSLIB (IHKSMG)
	INCLUDE	SYSLIB (IHKINI)
	ENTRY	IHKINI
	NAME	IHKINT(R)
/*		

Шаги 3—6 могут быть исключены, как и соответствующие операторы в шаге 7, если программы выхода не включаются в систему.

12.2.5. Размещение и инициализация наборов данных системы ДУВЗ

После генерации модулей ДУВЗ необходимо выполнить программу IHKINT для инициализации системной библиотеки ДУВЗ. Сюда же можно включить операторы DD для размещения библиотек пользователей и активной области ДУВЗ.

Задание на инициализацию системной библиотеки ДУВЗ выглядит следующим образом:

//ИМЯ	JOB	Параметры
//INIT	EXEC	PGM=IHKINT
//SYSLIB	DD	DSN=CRJE.SYSLIB, UNIT=50xx, VOL=
//		=SER=метка,
//		DISP=(,CATLG), DCB=(LRECL=80,
//ИМЯ1	DD	RECFM=FB, BLKSIZE=880),
//		SPACE=(TRK,(a,1))
//		DSN=CRJE.LIB. и дплз, VOL=SER=метка,
//		UNIT=50xx, DISP=(,CATLG),
//		SPACE=(TRK,(b,d),,CONTIG),
		DCB=(LRECL= { 88 } , BLKSIZE= { 880 } ,
		RECFM=FB)
//ИМЯ2		...
//ACTIVE	DD	DSN=CRJE.ACTIVE,
//		VOL=SER=метка, UNIT=50xx,
		DISP=(,CATLG),
		TRK
//		SPACE={CYL}, (количество),, CONTIG)

где а — количество дорожек для системной библиотеки, вычисляемое по формуле:

$$a = 5 + \frac{A + Ц + 3 + C}{K};$$

б — количество дорожек для библиотеки пользователей ДУВЗ, вычисляемое по формуле:

$$b = \frac{H \cdot O}{M};$$

д — количество блоков в оглавлении, равное $\frac{H}{6}$;

количество — количество дорожек для активной области, вычисляемое по формуле:

$$\text{количество} = S + \frac{A + Ц + 3 + C + B}{T} \cdot 1,1 \left(1 + \frac{N}{T} \right)$$

В вышеприведенных формулах:

А — число абонентов;

Ц — число циркулярных сообщений;

З — максимальное число заданий;

С — максимальное число задержанных сообщений;

Б — число активных абонентов;

Н — число наборов данных ДУВЗ;

О — среднее число операторов в наборах данных ДУВЗ;

Н — среднее число строк в активном наборе данных;

К — 33 для НМД типа ЕС-5050; 77 — для ЕС-5061; 121 — для ЕС-5066;

М — 30 для НМД типа ЕС-5050; 70 — для ЕС-5061; 110 — для ЕС-5066;

Т — 21 для НМД типа ЕС-5050; 31 для ЕС-5061; 41 — для ЕС-5066.

12.2.6. Процедура запуска ДУВЗ

Перед использованием системы ДУВЗ в библиотеку SYS1.PROCLIB необходимо занести процедуру запуска ДУВЗ, имеющую следующий вид:

//CRJE	EXEC	PGM=INKBGN[,PARM=(bb, $\left[\begin{matrix} 27K \\ 20K \\ 16K \end{matrix} \right]$,GH)]
//ACTIVE	DD	DSN=имя1, DISP=OLD
//LINE1	DD	UNIT=тип
//LINE2	DD	UNIT=тип
//LIB1	DD	VOL=SER=метка, UNIT=50xx, DISP=OLD
//LIB2	DD	VOL=SER=метка, UNIT=50xx, DISP=OLD
//JOBIN1	DD	VOL=SER=метка2, UNIT=50xx, SPACE=(800,(20, 10), RLSE, CONTIG)
//JOBIN2	DD	VOL=SER=метка2, UNIT=50xx, SPACE=(800,(20, 10), RLSE, CONTIG)
//CRxxxxxx	DD	VOL=SER=xxxxxx, UNIT=50xx, DISP=OLD

где bb — число дополнительных блоков основной памяти по 2 Кбайта (от 1 до 23) для транзитной области ДУВЗ (по умолчанию транзитной области отводится четыре блока по 2 Кбайта). В режиме SVS целесообразно задавать максимальное число дополнительных блоков для транзитной области; тем самым все нерезидентные модули ДУВЗ будут находиться в виртуальной памяти;

$\left\{ \begin{matrix} 27K \\ 20K \\ 16K \end{matrix} \right\}$ — определяет размер резидентной области для синтак-

сического анализатора с языка ПЛ/1. Если этот или следующий параметр опускается, то соответствующие синтаксические анализаторы не включаются в систему ДУВЗ;

GH — указывает, что в систему ДУВЗ включается синтаксический анализатор с языка Фортран;

имя1 — указывает имя набора данных, являющегося активной областью ДУВЗ;

тип — указывает тип (например, 7066) или адрес локального дисплея или удаленного АП. Для консолей оператора ЭВМ операторы DD не кодируются; имя оператора DD должно совпадать с указанным в операнде DDLINE макрокоманды CRJELINE;

метка — указывает метку типа, на котором находятся библиотеки пользователей ДУВЗ;

метка2 — указывает метку тома, на котором будут размещаться рабочие наборы данных для передачи заданий в пакетную обработку; имя оператора DD должно совпадать с указанным в операнде DDSYSIN макрокоманды CRJELINE;

xxxxxx — указывает метку тома, на котором находятся библиотеки ОС ЕС, которые будут редактироваться в системе ДУВЗ.

12.2.7. Процедура системного ввода

Для передачи заданий в пакетную обработку требуется процедура с именем RDRCRJE, которая выглядит следующим образом:

```
//IEFPROC          EXEC          PGM=IEFIRC, REGION=48K,
//                  PARM=
//                  ='уплввддеееааррркмттттттттггббс'
//IEFRDER          DD            UNIT=5010, LABEL=(,NL),VOLUME=
//                  =SER=SYSIN,
//                  DCB=(BLKSIZE=800, RECFM=FB,
//                  BUFL=800,
//                  BUFNO=1, LRECL=80)
//IEFPDSI          DD            DSN=SYS1. PROCLIB, DISP=SHR
//IEFDATA          DD            UNIT=SYSDA, SPACE=(80,(500, 500),
//                  RLSE, CONTIG),
//                  DCB=(BUFNO=2, LRECL=80, RECFM=F,
//                  BUFL=80,
//                  BLKSIZE=80)
```

где у — цифра, указывающая необходимость кодирования учетной информации и имени программиста в операторе JOB:

Цифра	Нужна учетная информация	Нужно ли имя программиста
0	Нет	Нет
1	Нет	Да
2	Да	Нет
3	Да	Да

пп — приоритет, присваиваемый заданию (от 00 до 14), если он явно не задан в операторе JOB;

ввв — максимальное подразумеваемое значение времени выполнения шага задания в минутах;

ddd — подразумеваемое значение первичного числа дорожек для наборов данных SYSOUT;

eee — подразумеваемое значение вторичного числа дорожек для наборов данных SYSOUT;

aaa — диспетчерский приоритет для программы системного ввода;

rrr — подразумеваемое значение для размера зоны, присваиваемой шагам заданий;

к — указывает диспозицию команд, встретившихся во входном потоке; 0 — команда передается на выполнение; 1 — команда пересылается оператору ЭВМ и передается на выполнение; 2 — команда пересылается оператору ЭВМ для подтверждения, что ее нужно выполнять; 3 — команда игнорируется;

м — указывает возможность обхода обработки метки (0 или 1). Нуль приводит к тому, что параметр BLP в операнде LABEL оператора DD игнорируется;

тттттт — указывает подразумеваемое устройство для наборов данных SYSOUT, которое может быть задано адресом, типом или именем группы;

гггг — должно быть 'E000';

бб — подразумеваемое значение для операнда MSGLEVEL оператора JOB;

с — подразумеваемое значение для операнда MSGCLASS оператора JOB.

Пример поля PARM:

PARM='30203002001024903030SYSDA E00001R'

В издании 6.1 процедура RDRCRJE автоматически заносится в библиотеку процедур при генерации операционной системы.

12.3. Команды оператора АП

Операнды команд оператора АП отделяются друг от друга пробелом или запятой. В издании 6.1 допустима сокращенная форма оператора команды.

12.3.1. CANCEL — отменить задание

Команда CANCEL удаляет из системы ДУВЗ любое задание, введенное ранее данным оператором АП. Формат:

{ CANCEL } имязад
C

где имязад — имя задания, удаляемого из системы ДУВЗ.

После удаления задания оператор АП получает сообщение.

12.3.2. CONTINUE — продолжить вывод

Команда CONTINUE используется для продолжения прерванного вывода результатов выполнения задания, начатого командой OUTPUT. Формат:

{ CONTINUE } [H[ERE]
CO B[EGIN]
N[EXT]]

где H[ERE] — указывает, что вывод продолжается с того места, где он был прерван;

B[EGIN] — указывает, что вывод возобновляется с начала данного набора данных, если он был прерван во время его вывода или с прерванного блока сообщений, если выводились сообщения;

N[EXT] — указывает, что вывод продолжается с начала следующего набора данных или со следующего блока сообщений.

12.3.3. DELETE — удалить набор данных

Команда DELETE удаляет набор данных из библиотеки пользователя ДУВЗ или из ОС ЕС. Формат:

{ DELETE } { имянд
D 'имянд [(имя-раздела)]' }

где имянд — имя набора данных в библиотеке пользователя ДУВЗ;

'имянд[(имя-раздела)]' — указывает полностью имя набора данных ОС ЕС, причем первым квалификатором в имени набора данных должен быть идентификатор оператора АП, выдавшего команду. Набор данных должен быть последовательным или библиотечным, каталогизированным и находиться на одном томе. Для библиотечного набора данных можно указать имя раздела.

12.3.4. EDIT — редактировать набор данных

С помощью команды EDIT пользователь переходит в режим редактирования. Формат:

$$\text{EDIT} \left\{ \begin{array}{l} \text{имянд} \quad [(\text{идплз} \quad [\text{ключ}])] \\ \text{'имянд} \quad (\text{имя-раздела})' \end{array} \right\} \left[\frac{\text{NEW}}{\text{OLD}} \right] \left[\frac{\text{NUM}}{\text{NONUM}} \right] \left[\frac{\text{LONG}}{\text{Lnn[n]}} \right] \left[\frac{\text{S[CAN]}}{\text{NOS[CAN]}} \right]$$

$$\left[\begin{array}{l} \text{PL/I (параметры)} \\ \text{FORT} \left\{ \begin{array}{l} \text{G} \\ \text{H} \end{array} \right\} \\ \text{DSLST} \\ \text{CLIST} \\ \text{DATA} \\ \text{TEXT} \end{array} \right]$$

где имянд — имя создаваемого набора данных ДУВЗ или имя редактируемого набора данных ДУВЗ или ОС ЕС;

идплз — идентификатор пользователя ДУВЗ, в библиотеке которого находится редактируемый незащищенный набор данных ДУВЗ;

ключ — ключ защиты, присвоенный редактируемому защищенному набору данных ДУВЗ, находящемуся в библиотеке пользователя ДУВЗ, имеющего идентификатор идплз;

имя-раздела — имя раздела библиотечного каталогизированного набора данных ОС ЕС;

NEW — указывает, что создается новый набор данных ДУВЗ;

OLD — указывает, что редактируется существующий набор данных ДУВЗ или ОС ЕС;

NUM — означает, что восемь последних символов каждой записи в наборе данных будут содержать номер строки. Новому набору данных присваивается характеристика SEQ;

NONUM — указывает, что вводимые строки нумероваться не будут; набору данных присваивается характеристика NOSEQ;

LONG — указывает, что обрабатываются данные, размер записи которых равен 120 байтам (допустим только в модификации 1 издания 6.1);

Lnn[n] — указывает, что обрабатываются данные с размером записи, равным nn или nnn байт. Максимальный размер записи — 144 байта (допустим только в модификации 1 издания 6.1);

S[CAN] — задает автоматический синтаксический контроль вводимых строк;

NOS[CAN] — указывает, что синтаксический контроль вводимых строк производиться не будет;

PL/1 (параметры) — указывает операторы языка ПЛ/1. Параметры задаются в формате:

$$[\text{число1} \text{ число2}] \left[\begin{array}{l} \text{C[HAR]48} \\ \text{C[HAR]60} \end{array} \right]$$

где **число1** **число2** — определяют соответственно левую и правую границы оператора; **C[HAR]48** и **C[HAR]60** — указывают соответственно 48-символьный или 60-символьный алфавит;

FORT $\left\{ \begin{array}{l} \text{G} \\ \text{H} \end{array} \right\}$ — операторы языка Фортран;

DSLIST — указывает, что набор данных ДУВЗ содержит список имен наборов данных ДУВЗ. Каждая строка набора данных ДУВЗ содержит имя одного набора данных ДУВЗ в формате:

$$\text{имянд}[(\text{идплз}[\text{ключ}])]$$

где **имянд** не должно указывать набор данных ДУВЗ с характеристикой **DSLIS**T;

CLIST — указывает, что каждая строка набора данных содержит команду или подкоманду (кроме **LOGON**, **LOGOFF** и **EXEC**);

DATA — указывает, что набор данных содержит только данные;

TEXT — указывает, что набор данных содержит только данные, но в отличие от параметра **DATA** вводимые строчные буквы не преобразуются в прописные.

Примечание. В ОС ЕС издания 6.1 последние три операнда и операнд **NEW/OLD** команды **EDIT** принимают по умолчанию значения, заданные при генерации системы ДУВЗ макрокомандой **CRJEDEF** (см. 12.2.2).

Подкоманды команды EDIT. Имеются три режима работы пользователя ДУВЗ: режим команды, режим ввода и режим редактирования. Вначале пользователь находится в режиме команд. Переход в режим редактирования из режима команд производится командой **EDIT** **имянд OLD**. В режиме редактирования пользователь может выдавать подкоманды **EDIT**. Возврат в режим команд осуществляется подкомандой **END**. Переход в режим ввода из режима команд производится командой **EDIT** **имянд NEW**. В режиме ввода пользователь вводит новые данные. Выход из режима ввода и переход в режим редактирования осуществляются вводом нулевой строки. Из режима редактирования также можно перейти в режим ввода с помощью нулевой строки или командой **INPUT**. Возврат из режима ввода в режим команд всегда происходит через режим редактирования.

В режиме редактирования пользователь работает с набором данных, находящимся в активной области ДУВЗ. Последующие подкоманды можно вводить только в режиме редактирования.

Неявная подкоманда. Используется для удаления замещения, или ввода строки в активном наборе данных. Формат:

номстр[текст]

где номстр — номер удаляемой или вставляемой строки;

текст — текст, который вводится в данной строке в активный набор данных. Текст может иметь длину до 80 (NOSEQ), 72 (SEQ) или 144 (LONG) символов. Если операнд «текст» опущен, то строка с данным номером удаляется из активного набора данных.

Подкоманда CANCEL (см. 12.3.1). Может выдаваться и в режиме редактирования для отмены ранее введенного задания. Формат:

CA[NCEL]имяз

где имяз — имя удаляемого задания.

Подкоманда CHANGE. Заменяет последовательность символов в строке или группе строк активного набора данных, т. е. выполняет так называемое контекстное редактирование. Формат:

C[HANGE] номстр[номстр]Δтекст1Δтекст2Δ[A[LL]]

где номстр[номстр] — номер строки или группы строк, в которые вносятся изменения;

Δтекст1 Δтекст2Δ — операнд «текст1» указывает заменяемые символы, а операнд «текст2» — замещающие символы. Каждая из последовательностей может содержать до 40 символов и иметь различную длину. Разделителем Δ может быть любой нецифровой символ, но не запятая. Он не должен использоваться в последовательностях «текст1» и «текст2»;

A[LL] — указывает, что замещается каждое вхождение символов «текст1»; если операнд ALL опущен, то замещается только первое вхождение «текст1».

Примечание. Если разделителем является дефис, то он не должен быть последним символом в строке (т. е. нужно ввести дополнительный пробел), иначе он будет рассматриваться как признак продолжения строки.

Подкоманда DELETE. Удаляет строку или группу строк из активного набора данных. Формат:

D[ELETE] [номстр[номстр]]

где номстр [номстр] — номера удаляемых строк; если указаны два номера, то удаляется группа строк от первого до второго номера включительно; если поле операндов пусто, то удаляется последняя строка в наборе данных.

Подкоманда END. Возвращает пользователя в режим команд; при этом содержимое участка активной области, отведенного пользователю, не сохраняется. Чтобы сохранить данные, находящиеся в активной области, пользователь должен выдать предварительно подкоманду SAVE (см. ниже). Формат:

E[ND]

Подкоманда FIND (издание 6.1). Находит в активной области строку, в которой впервые встречается заданная последовательность символов. Формат:

F[IND] [номстр]ΔпоследсимΔ

где номстр — номер строки в активной области, с которой начинается поиск; если этот операнд опущен, то поиск начинается с начала активной области;

послсим — задает последовательность отыскиваемых символов; максимально можно указать до 56 символов; вместо Δ необходимо указать любой непустой символ, не встречающийся в отыскиваемой последовательности.

Подкоманда INPUT. Переводит пользователя из режима редактирования в режим ввода. Формат:

$$I[INPUT] \quad [\text{номстр} \left[\begin{array}{c} [\text{приращение}] [I] \\ R \end{array} \right] \left[\begin{array}{c} P[ROMPT] \\ NOP[ROMPT] \end{array} \right]]$$

где номстр — номер первой строки, вводимой в режиме ввода; если опущен, то вводимые строки добавляются к концу активного набора данных;

приращение — задает приращение, с которым будут присваиваться номера новым вводимым строкам; это приращение используется временно, если опущено, то берется приращение, заданное в характеристике набора данных (при создании набора данных эта характеристика обычно равна 10; ее можно изменить подкомандой RENUMBER (см. ниже));

I — указывает, что новые строки не должны замещать существующие; если строка со следующим номером уже существует, то пользователю АП посылается соответствующее сообщение;

R — указывает, что вводимые строки должны замещать существующие начиная с номера, указанного в команде.

Примечание. Если I и R опущены, то система не будет обращать внимание на то, имеются или нет строки с очередным номером.

P[ROMPT] — указывает, что перед вводом очередной строки система выдаст на АП ее номер;

NOP[ROMPT] — указывает, что система не будет выдавать номера очередных строк оператору АП (вводимые строки тем не менее нумеруются).

Подкоманда LIST. Распечатывает набор данных и приводит к выдаче на АП строк из активного набора данных. Формат:

$$L[IST] \quad [\text{номер}[\text{номстр}]] \left[\begin{array}{c} \text{NUM} \\ \text{NONUM} \end{array} \right]$$

где номстр [номстр] — номера выводимых строк; если указаны два номера, то выводится группа строк от первого до второго номера включительно; если номстр опущен, то выводится весь набор данных;

NUM — указывает, что вместе со строками выводятся их номера;

NONUM — указывает, что номера строк не выводятся.

Подкоманда MERGE. Используется для объединения строк активного набора данных и набора данных в библиотеке пользова-

теля ДУВЗ и для копирования строк в активном наборе данных. Формат:

$$M[ERGE] \left\{ \begin{array}{c} \text{имянд} \\ * \end{array} \right\} [\text{номстр} \text{ номстр}] [\text{номстр}]$$

где имянд — имя набора данных пользователя ДУВЗ, который объединяется с активным набором данных. Этот операнд можно задать следующими способами:

имянд — имя набора данных в библиотеке пользователя АП, выдавшего подкоманду MERGE;

имянд(идплз) — имя незащищенного набора данных в библиотеке пользователя АП, имеющего идентификатор идплз;

имянд(идплз ключ) — имя защищенного набора данных в библиотеке пользователя АП, имеющего идентификатор идплз; «ключ» указывает ключ защиты для данного набора данных;

* — указывает, что строки будут копироваться внутри активного набора данных;

номстр номстр — группа строк, вставляемых в активный набор данных; если этот операнд опущен, то вставляется весь набор данных;

номстр — номер строки в активном наборе данных, после которого вставляются новые строки; если этот операнд опущен, то строки добавляются к концу активного набора данных.

Подкоманда RENUMBER. Перенумеровывает строки в активном наборе данных. Формат:

$$REN[UMBER] \left[\begin{array}{c} \text{номстр} \\ \underline{10} \end{array} \left[\begin{array}{c} \text{приращение} \\ \underline{10} \end{array} \right] \right]$$

где номстр — номер первой строки;

приращение — приращение для номера каждой последующей строки.

Если набор данных имеет характеристику SEQ, то новые номера также заносятся в последние восемь байт записи.

Подкоманда SAVE. Сохраняет в библиотеке пользователя активный набор данных. Формат:

$$S[AVE] [\text{имянд}] [K[EY] (\text{ключ})]$$

где имянд — имя, под которым запоминается активный набор данных. Если имянд опущено, то принимается имя набора данных, указанное в команде EDIT;

K[EY] (ключ) — указывает, что набор данных защищается ключом, указанным в операнде, «ключ» — это любой идентификатор из трех символов.

Подкоманда SCAN. Запрашивает синтаксический анализ операторов в активном наборе данных. Формат:

$$SC[AN] [\text{номстр} [\text{номстр}]] \left\{ \begin{array}{c} \text{ON} \\ \text{OFF} \end{array} \right\}$$

где номстр [номстр] — номер проверяемой строки; если указаны два номера, то проверяется группа строк от первого до второго

Примечание. В поле операндов можно указать до 10 имен наборов данных, которые будут переданы во входной поток в том порядке, в котором указаны в команде. Любой из этих наборов данных может иметь характеристику LSLIST. В этом случае во входной поток включаются все наборы данных, перечисленные в данном наборе.

Подкоманда TABSET. Устанавливает логический формат для ввода и физический формат для вывода на АП, оборудованных средством табуляции. Формат:

$$\text{TAB[SET]} \quad \left[\begin{array}{c} \text{число ...} \\ \text{OFF} \end{array} \right] \quad \left[\begin{array}{c} \text{IN[PUT]} \\ \text{OUT[PUT]} \end{array} \right]$$

где число ... — формат ввода или вывода (можно определить до 10 позиций). Каждое число указывает номер позиции в строке, с которой начинается поле. Первая позиция имеет номер 0. При нажатии клавиши горизонтальной табуляции на АП при вводе в строку заносятся пробелы до позиции, заданной операндом «число»;

OFF — означает отмену табуляции;

IN[PUT] — указывает, что команда TABSET относится только к вводимым строкам;

OUT[PUT] — указывает, что команда TABSET относится только к выводимым строкам.

12.3.5. EXEC — выполнить группу команд

Команда EXEC выполняет группу команд, содержащихся в наборе данных ДУВЗ с характеристикой CLIST. Формат:

$$\left\{ \begin{array}{c} \text{EXEC} \\ \text{EX} \end{array} \right\} \quad \text{имянд} \quad \left[\begin{array}{c} \text{L[IST]} \\ \text{NOL[IST]} \end{array} \right]$$

где имянд — имя набора данных с характеристикой CLIST, содержащего группу команд; имя может указываться в следующем виде:

имянд — имя набора данных в библиотеке пользователя ДУВЗ, выдавшего команду;

имянд(идплз) — имя незащищенного набора данных в библиотеке пользователя ДУВЗ, имеющего идентификатор «идплз»;

имянд(идплз ключ) — имя и ключ защищенного набора данных в библиотеке пользователя ДУВЗ, имеющего идентификатор «идплз»;

L[IST] — указывает, что каждая выполняемая команда будет выводиться на АП;

NOL[IST] — означает, что выполняемые команды не будут выдаваться на АП.

Набор данных, указываемый командой EXEC, может содержать любые команды и подкоманды, за исключением команд LOGON, LOGOFF и EXEC.

12.3.6. LISTBC — выдать копии циркулярных сообщений

Команда LISTBC выдает на АП копии циркулярных сообщений. Формат:

$$\left\{ \begin{array}{l} \text{LISTBC} \\ \text{LISTB} \\ \text{LBC} \\ \text{LB} \end{array} \right\}$$

12.3.7. LISTDS — выдать информацию о наборе данных ДУВЗ

Команда LISTDS выдает информацию о наборе данных в библиотеке пользователя ДУВЗ. Формат:

$$\left\{ \begin{array}{l} \text{LISTDS} \\ \text{LISTD} \\ \text{LDS} \end{array} \right\} \quad \text{имянд} \quad [\text{S}[\text{TATUS}]] \quad [\text{H}[\text{ISTORY}]]$$

где имянд — имя набора данных в библиотеке пользователя ДУВЗ, выдавшего команду; если остальные операнды опущены, то будет выдана информация о характеристике содержания набора данных и о том, имеются ли в позициях 73—80 каждой записи номера строк;

S[TATUS] — запрашивает размер набора данных и его ключ;

H[ISTORY] — запрашивает дату создания набора данных, дату внесения последних изменений и количество обращений к набору данных со времени его создания (т. е. указание этого набора данных в командах EDIT OLD, SUBMIT, EXEC и MERGE).

12.3.8. LISTLIB — выдать оглавление библиотеки

Команда LISTLIB выдает на АП имена всех наборов данных в библиотеке пользователя ДУВЗ и информацию по каждому набору данных. Формат:

$$\left\{ \begin{array}{l} \text{LISTLIB} \\ \text{L} \end{array} \right\} \quad [\text{S}[\text{TATUS}]] \quad [\text{H}[\text{ISTORY}]]$$

где S[TATUS] — запрашивает дополнительно к именам и характеристикам размер и ключ каждого набора данных;

H[ISTORY] — запрашивает дату создания, дату последнего изменения и количество обращений для каждого набора данных.

12.3.9. LOGOFF — закончить сеанс

С помощью команды LOGOFF пользователь отключается от системы ДУВЗ. Формат:

$$\left\{ \begin{array}{l} \text{LOGOFF} \\ \text{OFF} \end{array} \right\}$$

На АП будет выдана информация о времени завершения и продолжительности сеанса.

12.3.10. LOGON — начать сеанс

С помощью команды LOGON пользователь подключается к системе ДУВЗ. Формат:

$$\left\{ \begin{array}{l} \text{LOGON} \\ \text{LO} \end{array} \right\} \text{ идплз/пароль} [A[CCT] \text{ (учетная-информация)}] \left[\begin{array}{l} \text{BC} \\ \text{NOBC} \end{array} \right] \\ \left[\begin{array}{l} \text{M[SGID]} \\ \text{NOM[SGID]} \end{array} \right]$$

где идплз/пароль — идентификатор пользователя ДУВЗ (до 7 символов) и пароль (до 8 символов), присвоенный пользователю системным программистом;

A[CCT] (учетная-информация) — учетная информация (до 33 символов). Параметры, содержащие пробелы или запятые, заключаются в апострофы;

BC — запрашивает вывод циркулярных сообщений;

NOBC — указывает, что циркулярные сообщения выдаваться на АП не будут;

M[SGID] — означает, что вместе с сообщениями на АП будут выдаваться их идентификаторы;

NOM[SGID] — указывает, что сообщения будут выдаваться на АП без идентификаторов.

12.3.11. OUTPUT — выдать результаты задания

Команда OUTPUT запрашивает выдачу на АП результатов задания, переданного в пакетную обработку. Выдаются результаты, направленные в системный класс вывода для ДУВЗ. Формат:

$$\left\{ \begin{array}{l} \text{OUTPUT} \\ \text{O} \end{array} \right\} \text{ имяз} \left[\begin{array}{l} \text{SMSG} \\ \text{NOSMSG} \end{array} \right]$$

где имяз — имя задания, переданного в пакетную обработку;

SMSG — означает, что кроме выходных наборов данных выдаются сообщения планировщика заданий ОС ЕС;

NOSMSG — указывает, что сообщения планировщика заданий ОС ЕС не выдаются (издание 6.1).

12.3.12. SEND — послать сообщение

Команда SEND посылает сообщение оператору ЭВМ или другому пользователю ДУВЗ. Описание команды см. в 12.3.4.

12.3.13. STATUS — запросить информацию о задании

Команда STATUS запрашивает информацию о задании, переданном в пакетную обработку. Формат:

$$\left\{ \begin{array}{l} \text{STATUS} \\ \text{ST} \end{array} \right\} [\text{имяз}]$$

где имяз — указывает имя задания, переданного в пакетную обработку; если операнд опущен, то запрашивается информация о всех заданиях, переданных данным абонентом в пакетную обработку.

12.3.14. SUBMIT — передать в пакетную обработку

Команда SUBMIT передает одно или несколько заданий в пакетную обработку. Формат:

$$\left\{ \begin{array}{l} \text{SUBMIT} \\ \text{SUB} \end{array} \right\}_{\text{имянд}}$$

где имянд — см. 12.3.4.

12.3.15. TABSET — установить табуляцию

Команда TABSET устанавливает логический формат для ввода и физический формат для вывода на АП. Формат:

$$\left\{ \begin{array}{l} \text{TABSET} \\ \text{T} \end{array} \right\} \left[\begin{array}{l} \text{число} \dots \\ \text{OFF} \end{array} \right] \left[\begin{array}{l} \text{IN[PUT]} \\ \text{OUT[PUT]} \end{array} \right]$$

Описание операндов см. в 12.3.4.

12.3.16. VARY — изменить характеристики ввода-вывода (издание 6.1)

Команда VARY изменяет некоторые параметры, относящиеся к АП. Формат:

$$\left\{ \begin{array}{l} \text{VARY} \\ \text{V} \end{array} \right\} \left\{ \begin{array}{l} [\text{DC}=\text{x}] [\text{DL}=\text{y}] [\text{HT}=\text{z}] [\text{EOM}=\text{m}] \\ [\text{HT}=\text{z}] [\text{WAIT}=\text{nnn}] \\ [\text{LINE}=\text{hhh}] \\ [\text{MODE}=\{\text{PAGE}|\text{ROLL}\}] \end{array} \right\}$$

где DC=x — задает знак отмены символа. По умолчанию этим знаком является @ ;

DL=y — задает знак отмены строки. По умолчанию этим знаком является [;

HT=z — задает знак табуляции. По умолчанию им является — ;

EOM=m — задает знак конца строки для ввода с пульта оператора. По умолчанию этим знаком является % ;

WAIT=nnn — задает время ожидания между двумя последовательными операциями записи на дисплей в режиме свертки (ROLL). По умолчанию принимается 250, т. е. 2,5 с ;

LINE=hhh — задает максимальное количество символов в строке (для пишущих машинок) ;

MODE={PAGE|ROLL} — PAGE задает переход в режим страничного вывода, ROLL — переход в режим свертки. Этот операнд можно использовать только для дисплейных АП и только в случае, если при генерации ДУВЗ в макрокоманде CRJELINE для данного АП указан режим свертки (ROLL).

Базисный телекоммуникационный метод доступа

Базисный телекоммуникационный метод доступа (БТМД) предоставляет программисту набор макрокоманд и соответствующих программ, выполняющих следующие функции:

- определение системы телеобработки;
- управление буферами основной памяти;
- трансляцию сообщений из кода передачи в код ЭВМ и обратно;
- активизацию системы телеобработки;
- прием и передачу сообщений;
- обнаружение и исправление ошибок;
- оперативную проверку устройств.

В операционной системе ОС ЕС издания 4.1 обеспечиваются мультиплексоры передачи данных МПД-1А, МПД-2, МПД-3, режим ЭВМ-ЭВМ, абонентские пункты АП-2, АП-4, АП-61, АП-63, АП-70 и телеграфные аппараты, работающие в коде МТК-2.

В операционной системе ОС ЕС издания 6.1 дополнительно обеспечиваются мультиплексоры передачи данных МПД-1 (ЕС-8401 и ЕС-8410), МПД-4 и абонентские пункты 7920 (локальный и удаленный), АП-1, АП-3, АП-5, АП-62 и АП-64, телетайпы Т51.

13.1. Определение системы телеобработки

13.1.1. DCB — определить группу каналов связи

Макрокоманда DCB определяет группу каналов связи с одинаковыми характеристиками. Формат:

```
[метка] DCB DSORG=CX,MACRF={ (R) | (W) | (R,W) }, DDNAME=имя-dd
          [BUFNO=число] [BUFL=длина] [BUFCB=адрес] [EXLST=
          =адрес] [BFTEK=D] [LERB=адрес] [EROPT={ [E] [R]
          [W] [C] [N] [T] [DEVD={BS|WT} } [MODE={ [IBC] [,CNTRL] }
          [MONDLY={nn|15} [IAM={YES|NO} } [WRU={YES|NO} }
          [EOM=x'hh'] [EOT='hh1F'] [READYQ={0|адрес}]
```

где DSORG=CX — определяет группу каналов связи для БТМД;
MACRF={ (R) | (W) | (R, W) } — указывает, что доступ к каналам связи будет осуществляться макрокомандами READ, WRITE или обеими вместе;

DDNAME=имя-dd — имя соответствующего оператора DD;
 BUFNO=число — задает число буферов для построения буферного пула; максимальное значение — 255;
 BUFL=длина — задает длину буферов (кратную 4) для построения буферного пула; максимальное значение — 32 760;
 BUFCB=адрес — адрес блока управления буферным пулом, построенным программистом;
 EXLST=адрес — адрес списка выходов на пользовательские программы;
 BFTEK=D — задает динамическую буферизацию; необходимо также построить буферный пул;
 LERB=адрес — адрес блока регистрации ошибок канала связи;
 EROPT=код:
 E — задает стандартные процедуры исправления ошибок;
 R — задает исправление ошибок чтения в дополнение к стандартным процедурам для стартстопных устройств (кроме телеграфного аппарата); не указывается для синхронных устройств или если используется динамическая буферизация;
 W — задает исправление ошибок записи в дополнение к стандартным процедурам для стартстопных устройств (кроме телеграфного аппарата); не указывается для синхронных устройств или если используется динамическая буферизация;
 C — задает использование в блоке регистрации ошибок (LERB) пороговых и суммирующих счетчиков для ОШИБОК В ДАННЫХ, ТРЕБУЕТСЯ ВМЕШАТЕЛЬСТВО И НЕТЕКСТОВЫХ ОШИБОК ТАЙМАУТА;
 N — указывает, что процедуры исправления ошибок не используются; N не указывается для синхронных устройств;
 T — задает использование средств оперативной проверки каналов связи;
 DEVD={BS|WT} — BS указывает синхронный режим, WT — группу телеграфных аппаратов;
 MODE=([IBC] [,CNTRL]) — определяет режим работы для синхронного способа передачи: IBC — определяет режим с КОНТРОЛЕМ ПРОМЕЖУТОЧНОГО БЛОКА; CNTRL — указывает, что в ситуации соперничества на двухпунктовом некоммутируемом канале связи управление передается центральной ЭВМ. Если CNTRL опущено, то в ситуации соперничества управление передается удаленной станции;
 MONDLY={nn|15} — задает задержку для разгона мотора телеграфного аппарата в символах; максимальное значение — 20;
 IAM={YES|NO} — YES указывает, что телеграфный аппарат может запросить идентификатор ЭВМ;
 WRU={YES|NO} — YES указывает, что ЭВМ и телеграфный аппарат могут запросить друг у друга идентификаторы;

ЕОМ=X'hh' — задает символ конца сообщения для телеграфного аппарата в шестнадцатеричном представлении кода передачи. Для МПД-1, МПД-1А, МПД-2, МПД-3 и МПД-4 ЕОМ=X'37';

ЕОТ=X'hh1F' — задает символ конца передачи (КП) для телеграфного аппарата в шестнадцатеричном представлении кода передачи. Для МПД-1, МПД-1А, МПД-2, МПД-3 и МПД-4 ЕОТ=X'2F1F';

READYQ={0|адрес} (только для локальной станции ЕС-7920) — 0 указывает, что если при выполнении макрокоманды ЧТЕНИЕ НАЧАЛЬНОЕ неготовое устройство переходит в состояние готовности, то БТМД выдаст макрокоманду RESETPL; «адрес» — адрес программы пользователя, которой асинхронно передается управление, если неготовое устройство переходит в состояние готовности. При этом старший байт в регистре 1 содержит относительный номер канала связи, а младшие три байта — адрес соответствующего блока DEB.

Если операнд READYQ опущен, то сбой ТРЕБУЕТСЯ ВМЕШАТЕЛЬСТВО рассматривается как постоянная ошибка ввода-вывода.

13.1.2. DFTRMLST — построить абонентский список

Макрокоманда DFTRMLST строит абонентский список, используемый при операциях ЧТЕНИЕ и ЗАПИСЬ. Для локального ЕС-7920 не используется. Формат:

[метка]DFTRMLST тип, операнды-зависящие-от-устройств

где тип — определяет тип абонентского списка в зависимости от типа удаленной станции или конфигурации канала связи:

- OPENLST — открытый список программного опроса стартстопных многоточковых каналов связи;
- WRAPLST — замкнутый список программного опроса стартстопных многоточковых каналов связи;
- SSALST — открытый список автоопроса стартстопных многоточковых каналов связи;
- SSAWLST — замкнутый список автоопроса стартстопных многоточковых каналов связи;
- DIALST — список вызовов или список ответов для коммутируемых каналов связи, кроме одностороннего ЕС-7925 на коммутируемом канале связи;
- IDLST — список вызовов или список ответов с проверкой идентификатора (ИД) для телеграфных аппаратов на коммутируемых каналах связи;
- BSCLST — список вызовов или список ответов с проверкой ИД для синхронных коммутируемых каналов связи; определяется ИД, по-

	сылаемый к удаленной станции, и ИД, принимаемый от удаленной станции;
SWLST	— список вызовов или список ответов с расширенной проверкой ИД для синхронных коммутируемых каналов связи с автоматическим или ручным набором номера или автоматическим ответом;
AUTOLST	— открытый список автоопроса для синхронных многопунктовых каналов связи;
AUTOWLST	— замкнутый список автоопроса для синхронных многопунктовых каналов связи;
WTTALST	— абонентский список для телеграфных аппаратов на некоммутируемом канале связи;
WTLIST	— список вызовов или список ответов для синхронных коммутируемых каналов с ручным вызовом или ответом;

операнды-зависящие-от-устройств — задают параметры абонентского списка:

- xx — шестнадцатеричное представление символа выборки или опроса в коде передачи;
- ххуу — шестнадцатеричное представление двух символов выборки или опроса в коде передачи;
- кол-цифр-номера — десятичное число, задающее количество цифр в телефонном номере;
- цифры-номера — цифры телефонного номера;
- кол-прием — количество символов в ожидаемом от удаленной станции ИД;
- прин-ИД — шестнадцатеричное представление ожидаемого ИД в коде передачи;
- кол-посыл — количество символов в посылаемом ИД;
- посыл-ИД — шестнадцатеричное представление посылаемого ИД в коде передачи.

Количество элементов в открытых или замкнутых списках не должно превышать 31.

Ниже приводятся форматы поля операндов макрокоманды DFTRMLST для различных АП.

АП-1 некоммутируемый

Опрос { OPENLST }, (ххуу, ...)
 Выборка OPENLST, (ххуу, ...)

где хх — адрес АП (латинские буквы А — Z в коде передачи);
 уу — адрес компонента АП: 0 — ввод с любого компонента;
 1 — пишущая машинка; 2 — дополнительное устройство вывода;
 3 — перфоленточное устройство ввода; 5 — клавиатура; 6 — перфоленточное устройство вывода; 8 — дополнительное устройство ввода; 9 — вывод на каждый готовый компонент.

АП-1 коммутируемый

Ответ-выборка или ответ-опрос DIALST, 0, (ххуу, ...)
Вызов-выборка или вызов-опрос DIALST, кол-цифр-номера,
цифры-номера, (ххуу, ...)

где ххуу — имеет то же значение, что и для некоммутируемого АП-1.

АП-1 с автоопросом

{ SSALST
SSAWLST }, (ххуу, ...)

где ххуу — см. выше для некоммутируемого АП-1.

АП-2, АП-3 (стартстопные) коммутируемые

Ответ DIALST, 0

Вызов DIALST, кол-цифр-номера, цифры-номера

где кол-цифр-номера — количество цифр в номере плюс единица;
цифры-номера — цифры телефонного номера с добавлением
X'0C'.

Синхронный коммутируемый канал связи

Автоматический вызов с расширенной проверкой ИД (используется в макрокоманде WRITE TC)

Автоматический ответ с расширенной проверкой ИД (используется в макрокоманде READ TC)

Автоматический вызов без расширенной проверки ИД (используется в макрокомандах READ TI или WRITE TI)

Автоматический ответ без расширенной проверки ИД (используется в макрокоманде READ TI)

Автоматический вызов без проверки ИД

Автоматический ответ без проверки ИД

Ручной вызов с расширенной проверкой ИД (используется в макрокоманде WRITE TC)

Ручной вызов без расширенной проверки ИД (используется в макрокоманде WRITE TC)

Ручной ответ без расширенной проверки ИД для модемов без средств звуковой сигнализации

Ручной ответ без расширенной проверки ИД для модемов со средствами звуковой сигнализации

SWLST, AD, кол-цифр-номера, цифры-номера, длина-элемента, [признак], кол-посыл, посыл-ИД [{,санкц-посл [упр-вел] [данные-польз]]}...]

SWLST, AN, длина-элемента, [признак], кол-посыл, посыл-ИД [{,санкц-посл [упр-вел] [данные-польз]]}...]

BSClST, кол-цифр-номера, цифры-номера, кол-прием, прин-ИД, кол-посыл, посыл-ИД

BSClST, 0, кол-прием, прин-ИД, кол-посыл, посыл-ИД

DIALST, кол-цифр-номера, цифры-номера

DIALST, 0

SWLST, MD, длина-элемента, [признак], кол-посыл, посыл-ИД [{,санкц-посл [упр-вел] [данные-польз]]}...]

WTLIST, 0, кол-прием, прин-ИД, кол-посыл, посыл-ИД

WTLIST, 0, кол-прием, прин-ИД, кол-посыл, посыл-ИД, длина, область

BSClST, 0, кол-прием, прин-ИД, кол-посыл, посыл-ИД

где длина-элемента — задает длину элемента в списке, который должен вместить санкционированную последовательность (см. ниже), данные пользователя (см. ниже) (0 и 4 байта) и управляющую величину (см. ниже) (1 байт);

признак — 4 указывает, что в каждом элементе списка нужно резервировать 4 байта для данных пользователя; 0 указывает, что резервировать память не нужно; по умолчанию принимается 0;

кол-посыл — длина в байтах поля (от 2 до 17), вмещающего посылаемый ИД вместе с символами ДА-0 (ответ) или КТМ (вызов);

посыл-ИД — шестнадцатеричное представление в коде передачи посылаемого удаленной станции ИД вместе с символами ДА-0 или КТМ;

санкц-посл — шестнадцатеричное представление санкционированного ИД вместе с символами КТМ (ответ) или ДА-0 (вызов);

упр-вел (список ответов) — 0 означает, что после приема ИД БТМД пошлет идентификатор ЭВМ и прочтет блок сообщения; 1 означает, что БТМД разорвет соединение, после чего будет заново ждать вызова; 2 — означает, что БТМД сразу же завершит макрокоманду READ TC;

упр-вел (список вызовов) — 0 означает, что после приема ИД БТМД сразу же завершит операцию WRITE TC; 1 означает, что БТМД повторно пошлет ИД КТМ;

данные-польз — перемещаемое выражение, помещаемое в элемент списка.

Синхронный многопунктовый канал связи

Опрос { AUTOLST
AUTOWLST }, (посыл-ИД...)

Выборка OPENLST, (посыл-ИД, ...)

где посыл-ИД — от одного до семи символов опроса или выборки, за которыми следует символ КТМ в шестнадцатеричном представлении кода передачи (в коде КОИ-7 — X'05').

Последний элемент списка опроса должен содержать символы КП в шестнадцатеричном представлении кода передачи (в коде КОИ-7 — X'04').

АП-5

Абонентский список WRAPLST, (04)
для макрокоманд
READ TI и READ TIR

АП-61, АП-62, АП-63, АП-64 (стартстопные)

Опрос { OPENLST
WRAPLST }, (ххуу, ...)

Выборка OPENLST, (ххуу, ...)

где хх — шестнадцатеричный адрес АП;

уу — шестнадцатеричный адрес компонента АП; уу=7F означает, что имеет место общий опрос.

АП-70 коммутируемый

Ответ DIALST, 0

Вызов DIALST ,кол-цифр-номера, цифры-номера

АП-70 на многопунктовом канале связи

Программный опрос { OPENLST
WRAPLST } , (хх, ...)

Автоопрос { SSALST
SSAWLST } , (ххуу, ...)

Выборка OPENLST, хх

где уу — шестнадцатеричное представление пробела в коде передачи.

Телеграфный аппарат некоммутируемый

Для макрокоманды READ TE

Запрос ИД телеграфного аппарата

WTTALST ,0,кол-прием,прин-ИД,
кол-посыл,посыл-ИД

Посылка ИД ЭВМ к ТА

WTTALST ,0,0,0,кол-посыл,
посыл-ИД

Телеграфный аппарат коммутируемый

Для макрокоманды READ TIE

Для макрокоманды WRITE TIE

IDLST ,0,кол-посыл,посыл-ИД

IDLST, кол-цифр-номера, цифры-номера, кол-посыл,посыл-ИД

13.1.3. CHGNTRY — изменить элемент абонентского списка

Макрокоманда CHGNTRY отменяет или возобновляет опрос или выборку станции или ее компонента. Формат:

[метка] CHGNTRY { адрес списка } ,тип,позиция, [число],
адрес=dcб действие

где адрес-списка — адрес первого элемента в списке;

адрес-dcb (только для локальных EC-7920) — адрес блока DCB;

тип—тип списка; допускаются следующие значения: OPENLST, WRAPLST, SSALST, SSAWLST, AUTOWST, SWLST и ATTLLST (только для локальных EC-7920);

позиция — первоначальная позиция изменяемого элемента (1, 2 и т. д.);

число — число символов опроса или выборки в каждом элементе списка; может быть опущен для типов SSALST, SSAWLST и SWLST; всегда опускается для ATTLLST;

действие — SKIP означает, что данный элемент нужно сделать неактивным; ACTIVATE означает, что данный элемент нужно сделать активным.

Коды возврата:

00 — успешное выполнение;

04 — абонентский список находится в работе;

08 — указанный элемент отсутствует в списке.

13.2. Управление буферами

Для построения буферного пула можно использовать макрокоманды: BUILD, GETMAIN, GETPOOL (см. 7.4). Кроме того, БТМД может построить буферный пул при открытии группы каналов связи, если в макрокоманде DCB закодированы операнды BUFNO и BUFL (см. 13.1.1). Буферы можно получать из буферного пула и отправлять их обратно с помощью макрокоманд REQBUF и RELBUF.

13.2.1. REQBUF — запросить буфер

Макрокоманда REQBUF запрашивает определенное число буферов из буферного пула. Формат:

[метка] REQBUF адрес-dcb,регистр [число]

где адрес-dcb — адрес соответствующего блока DCB;

регистр — регистр (от 2 до 12), в который БТМД должен поместить адрес первого выделенного буфера;

число — число запрашиваемых буферов.

Коды возврата (десятичные):

00 — успешное выполнение;

04 — частичное удовлетворение запроса. Запрошено больше буферов, чем имеется в буферном пуле;

08 — свободных буферов нет;

12 — нет буферного пула;

16 — нет программы управления буферным пулом. (Эта программа автоматически загружается в память при открытии каналов связи, если в DCB указано BUFTEK=D или если в DCB имеется адрес блока управления буферами.)

13.2.2. RELBUF — освободить буфер

Макрокоманда RELBUF возвращает в буферный пул цепочку буферов. Формат:

[метка] RELBUF адрес-dcb,адрес-буфера

где адрес-dcb — адрес соответствующего блока DCB;

адрес-буфера — регистр (от 2 до 12), в котором содержится адрес первого освобождаемого буфера.

Коды возврата (десятичные):

00 — успешное выполнение;

04 — буфер уже находится в пуле;

12 — нет буферного пула;

16 — нет программы управления буферным пулом.

13.3. Трансляция сообщений

Таблицы трансляции строятся макрокомандой ASMTRTAB, трансляция сообщений производится макрокомандой TRNSLATE. Коды ДКОИ, КОИ-7 и МТК-2 приведены в табл. 13.1, 13.2 и 13.3.

Вторые четыре бита	Первые четыре бита															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	ПУС	АР1	ВЦФ		Пробел	&	-		ц	й	я	б	{	}	—	0
1	НЗ	СУ1	НЗН		≡		/		а	ј	—	ы	А	Ј		1
2	НТ	СУ2	РП	СИН					б	ћ	ѕ	з	В	К	ѕ	2
3	КТ	СУ3							с	ѣ	џ	ш	С	Л	Т	3
4	ВЫП	ВСТ	БК	ВКП					д	т	ц	э	Д	М	У	4
5	ГТ	НС	ПС	ОСУ					е	п	у	щ	Е	N	У	5
6	НП	ВШ	КБ	ВП				ю	ф	о	ш	ч	Ф	О	У	6
7	ЗБ	ОЖД	АР2	КП				а*	g	р	х	ѣ	Г	Р	Х	7
8		АН						б	h	q	у	ю	Н	Q	У	8
9		КН						в	і	т	з	А*	І	Р	Z	9
A	НРВ	УУК	УР		[]		:	џ	к	р*	б	Х*	Н*	Т	З
B	ВТ	СП1	СП2	СП3	.	Ѧ	,	#	е*	п	с*	ц	И	О*	У	Ш
C	ПФ	РФ		СТП	<	*	%	Ѡ	ф	м	т	Д	Й	П	Ж	Э
D	ВК	РГ	КТМ	НЕТ	()	—	ѡ	з	н	у*	Е*	К*	Я	В*	Щ
E	ВЫХ	РЗ	ДА		+	;	>	=	х*	о*	ж	ф	П	Р*	б	Ч
F	ВХ	РЗ	ЗВ	ЗМ	!	Λ	?	"	и	п	в*	Г	М*	С*	Ы	≡

Примечание. Буквы русского алфавита, помеченные *, должны иметь код, соответствующий латинской букве, если они не отличаются шрифтом.

Шестнадцатеричное представление кода КОИ-7

Таблица 13.2

Вторые четыре бита	Первые четыре бита															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	ПЧС	AP1	Пробел	0	Q	P	▀	p			Пробел	0	ю	п	ю	п
1	НЗ	СЧ1	!	1	A	Q	a	q			!	1	a	я	а	я
2	НТ	СЧ2	"	2	B	R	b	r			"	2	б	р	б	р
3	КТ	СЧ3	#	3	C	S	c	s			#	3	ц	с	ц	с
4	КП	СТП	⌘	4	D	T	d	t			⌘	4	д	т	д	т
5	КТМ	НЕТ	%	5	E	U	e	u			%	5	е	у	е	у
6	ДА	СИН	&	6	F	V	f	v			&	6	ф	ж	ф	ж
7	ЗВ	КБ	▀	7	G	W	g	w			▀	7	г	в	г	в
8	ВШ	АН	(8	H	X	h	x			(8	х	б	х	б
9	ГТ	КН)	9	I	Y	i	y)	9	и	ы	и	ы
A	ПС	ЗМ	*	:	J	Z	j	z			*	:	й	з	й	з
B	ВТ	AP2	+	;	K	[k	{			+	;	к	ш	к	ш
C	ПФ	PH4(PФ)	,	<	L	\	l				,	<	л	э	л	э
D	ВК	PH3(PГ)	-	=	M]	m	}			-	=	м	щ	м	щ
E	РЧС	PH2(PЗ)	.	>	N	^	n	-			.	>	н	ч	н	ч
F	ЛАТ	PH1(PЗ)	/	?	O	—	o	3б			/	?	о	ь	о	3б

Таблица 13.3

Шестнадцатеричное представление кода МТК-2

Вторые четыре бита	Первые четыре бита					
	0	1	2	3	4	5
0	РУС	Е	РУС	З	РУС	Е
1	Т	З	5	+	Т	З
2	ВК	Д	ВК	КТМ	ВК	Д
3	О	В	9	?	О	Б
4	Пробел	5	Пробел	7	Пробел	С
5	Н	У	Щ	6	Х	Ы
6	Н	Р	,	Э	Н	Ф
7	М	Х	.	/	М	Ь
8	ПС	А	ПС	—	ПС	А
9	Л	W)	2	Л	В
А	Р	Ј	4	Ю	Р	Й
В	Г	ЦИФ	Щ	ЦИФ	Г	ЦИФ
С	І	У	8	7	И	У
Д	Р	Q	О	1	П	Я
Е	С	К	:	(Ц	К
Р	У	ПАТ	=	ПАТ	ЖС	ЛАТ

13.3.1. ASMTRTAB — построить таблицу трансляции

Макрокоманда ASMTRTAB строит 256-байтовую таблицу трансляции. Формат:

ASMTRTAB имя-таб,...

где имя-таб — указывает нужную таблицу трансляции:

для входных сообщений

ТА RCTA

другие АП RCK7

для выходных сообщений

ТА SCTA

АП-70 (ВНР) S70K

другие АП SCKR (русская половина КОИ-7)

SCKL (латинская половина КОИ-7)

13.3.2. TRNSLATE — транслировать сообщение

Макрокоманда TRNSLATE транслирует сообщение в нужный код с помощью таблицы трансляции, заданной макрокомандой ASMTRTAB. Формат:

TRNSLATE [адрес-dcb], имя-таб, область, длина

где адрес-dcb — адрес DCB; кодируется, если длина указана как 'S';

имя-таб — имя таблицы трансляции (см. 13.3.1);

область — адрес области в основной памяти, где содержится транслируемое сообщение;

длина — задает число транслируемых байт. Если транслируется цепочка буферов, то этот операнд кодируется как 'S'.

13.4. Активизация и деактивизация системы телеобработки

13.4.1. OPEN и CLOSE — открыть и закрыть группу каналов связи

Макрокоманда OPEN инициализирует блок DCB и загружает необходимые программы метода доступа. Макрокоманда CLOSE выполняет обратные действия. Формат:

[метка] { OPEN
CLOSE } (адрес-dcb , ...), [MF = { L
(Е, имя-списка) }]

где адрес-dcb — адрес блока DCB;

имя-списка — адрес макрокоманды в списковой форме.

13.4.2. LOPEN — повторно открыть канал связи

Макрокоманда LOPEN выдается в случаях, когда программа обнаружила, что канал связи не был включен (код возврата X'14' после макрокоманд READ или WRITE). Макрокоманда LOPEN повторно включает канал связи. Формат:

[метка] LOPEN адрес-DECB

где адрес-DECB — указывает адрес соответствующего блока DECB.

Коды возврата (десятичные):

- 00 — успешное выполнение;
- 04 — канал связи не открыт;
- 08 — канал связи занят;
- 12 — неверный относительный номер канала связи;
- 16 — не открыт блок DCB;
- 20 — запрос отвергнут, так как локальный дисплей EC-7920 используется программой неавтономного тестирования OLTER.

13.5. Прием и передача сообщений

13.5.1. READ и WRITE — читать и писать блок данных

Прием и передача сообщений производятся макрокомандами READ и WRITE. Формат:

$$[\text{метка}] \left\{ \begin{array}{l} \text{READ} \\ \text{WRITE} \end{array} \right\} \text{ адрес-decb, тип, адрес-dcb, } \left\{ \begin{array}{l} [\text{область-ввода-} \\ \text{вывода}] \\ \{ [\text{область-ввода}], \\ [\text{область-вывода}] \} \end{array} \right\}$$
$$\left\{ \begin{array}{l} \{ [\text{длина-области-ввода-вывода}] \\ \{ [\text{длина-области-вывода}] [\text{длина-области}] \} \} \\ [\text{элемент}] \\ [\text{относительный-номер-канала}] [., \text{MF} = \{ \text{L} | \text{E} \}] \end{array} \right\}$$

где адрес-decb — адрес блока DECB;

тип — тип операции в зависимости от типа АП и канала связи (табл. 13.4, 13.5 и 13.6). Указание T в списковой форме макрокоманды свидетельствует о том, что тип будет указан в соответствующей исполнительной макрокоманде;

адрес-dcb — адрес блока DCB;

область-ввода-вывода — адрес области ввода (READ) и вывода (WRITE). При динамической буферизации этот операнд кодируется как 'S'; операнд может быть опущен для READ с типом TIQ и TQ и WRITE с типом TR, TQ, TN, TA, TB, TW;

область-ввода, область-вывода (только для синхронной передачи в макрокоманде WRITE диалогового типа TIV, TIVX, TTV, TTVX) — указывает адрес области ввода и вывода; при динамической буферизации область ввода может кодироваться как 'S';

длина-области-ввода-вывода — длина области, определенной операндом «область-ввода-вывода»; в макрокоманде WRITE можно кодировать 'S'; операнд не нужно указывать для READ с типом TIQ и TQ и WRITE с типом TR, TQ, TN, TA, TB, TW;

длина-области-ввода, длина-области-вывода (только для синхронной передачи в макрокоманде WRITE диалогового типа) — длина областей, определенных операндами «область ввода» и «область вывода»; этот операнд можно кодировать как 'S';

элемент — адрес абонентского списка или элемента:

для OPENLST, AUTOLST или SSAWLST указывает адрес элемента внутри списка;

Операция	Код	АП-1		АП-2		АП-3		АП-5	АП-61, АП-63, АП-64	АП-70					ТА	
		В	К	В	К	В	П			К	П	К	У	В, О		
														В		О
Запись продолженная	ТТ	×	×						×		×	×	×	×	×	
Запись, продолженная со сбросом	ТТТ	×	×						×		×	×	×	×	×	
Запись, продолженная в режиме диалога	ТТТ	×	×													
Запись в режиме диалога	ТВ					×										
Запись в режиме диалога со сбросом	ТВТ															
Запись с частичным стиранием	ТВТ															
Запись с частичным стиранием и со сбросом	ТВТ															
Запись по адресу строки	ТЛ								×							
Запись по адресу строки со сбросом	ТЛТ								×							
Запись со стиранием	ТС								×							
Запись со стиранием со сбросом	ТСТ								×							
Запись положительного подтверждения	ТА	×	×						×				×	×	×	
Запись отрицательного подтверждения	ТН	×	×						×				×	×	×	
Запись разрывающая	ТВ															
Запись запроса	ТQ															
Запись с позиции и со сбросом	ТХТ															
Запись с позиции	ТХТ															
Запись с частичным стиранием	ТВ															

Примечания. В — выделенный канал, К — коммутируемый, П — со средствами проверки, У — со средствами управления передачей, О — управление станцией.

13.5.4. TWAIT — ожидать окончания одной из операций ввода-вывода

Макрокоманда TWAIT переводит программу в состояние ожидания, пока не завершится хотя бы одна из операций ввода-вывода. Формат:

[метка] TWAIT (регистр-ЕСВ),ECBLIST=адрес-списка

где регистр-ЕСВ — регистр, в который будет помещен адрес блока ЕСВ для завершившегося события;

адрес-списка — адрес списка блоков ЕСВ.

Т а б л и ц а 13.5

Операции ЧТЕНИЕ и ЗАПИСЬ (синхронные каналы связи)

Операция	Код	Некоммутируемый двух-пунктовый канал связи	Многopунктовый канал связи	Коммутируемый канал связи
Чтение начальное	TI	×	×	×
Чтение с соединением	TC			×
Чтение продолженное	TT	×	×	×
Чтение повторное	TP	×	×	×
Чтение запроса	TQ	×		
Чтение запроса начальное	TIQ	×	×	×
Чтение с запросом переворота	TRV	×	×	×
Запись начальная	TI	×	×	×
Запись начальная со сбросом	TIR	×	×	×
Запись сброса	TR	×	×	×
Запись запроса	TQ	×	×	×
Запись разъединяющая	TD			×
Запись ждущая	TW	×	×	×
Запись начальная в режиме диалога	TIV	×	×	×
Запись, продолженная в режиме диалога	TTV	×	×	×
Запись начальная в прозрачном режиме	TIX	×	×	×
Запись начальная в прозрачном режиме со сбросом	TIXR	×	×	
Запись блока в прозрачном режиме начальная	TIE	×	×	×
Запись в прозрачном режиме продолженная	TTX	×	×	×
Запись в прозрачном режиме продолженная со сбросом	TTXR	×	×	×
Запись блока в прозрачном режиме продолженная	TTE	×	×	×
Запись в прозрачном диалоговом режиме начальная	TIVX	×	×	×
Запись в прозрачном диалоговом режиме продолженная	TTVX	×	×	×
Запись с соединением	TC			×
Запись разрывающая	TB			×

В регистр 15 после выполнения макрокоманды TWAIT будет занесено смещение адреса ЕСВ от начала списка адресов блоков ЕСВ.

Операции ЧТЕНИЕ и ЗАПИСЬ для локальной станции ЕС-7920

Операция	Код	Операция	Код
Чтение начальное	TI	Чтение буфера с позиции	TBP
Чтение модифицированное	TM	Запись начальная	TI
Чтение модифицированное позиции	TMP	Запись со стиранием	TS
Чтение буфера	TB	Запись со стиранием незащищенная	TUS

13.6. Регистрация ошибок

13.6.1. LERB — построить блок регистрации ошибок в канале связи

Формат макрокоманды:

[метка] LERB число [{[порог-1],[порог-2],[порог-3],[порог-4]]}...]

где число — задает число каналов связи (от 1 до 255), для которых строятся блоки LERB;

порог-1 — задает число передач (от 1 до 255), после которого содержимое пороговых счетчиков прибавляется к значениям суммирующих счетчиков и затем сбрасывается в нуль, по умолчанию принимается равным 255;

порог-2 — задает пороговое значение счетчика ошибок в данных, после которого печатается содержимое пороговых счетчиков, сбрасываемых в нуль; можно задавать от 1 до величины «порог-1»; по умолчанию принимается равным min (10, «порог-1»);

порог-3 — задает пороговое значение счетчика ошибок ТРЕБУЕТСЯ ВМЕШАТЕЛЬСТВО, после которого печатается содержимое пороговых счетчиков и они сбрасываются в нуль; можно задавать от 1 до величины «порог-1»; по умолчанию принимается равным min (5, «порог-1»);

порог-4 — задает пороговое значение счетчика ошибок тайм-аута, после которого печатается содержимое пороговых счетчиков и они сбрасываются в нуль. Можно задавать от 1 до величины «порог-1»; по умолчанию принимается равным min (5, «порог-1»).

13.6.2. LERPRT — распечатать содержимое счетчиков на консоли оператора

Макрокоманда LERPRT выдает оператору ЭВМ сообщение, в котором указывается содержимое счетчиков ошибок. Формат:

[метка] LERPRT адрес-dcb [, номер]
[.ид-консоли] [CLEAR={YES|NO}]

где адрес-dcb — адрес блока DCB;

номер — относительный номер канала связи;

ид-консоли (для варианта ОС ЕС с обеспечением нескольких консолей) — адрес байта, в котором находится идентификатор консоли;

CLEAR={YES|NO} — YES указывает, что после распечатки все счетчики должны быть установлены в ноль.

13.7. Оперативная проверка устройств

Средства оперативной проверки устройств в БТМД позволяют запрашивать при помощи сообщений определенного формата оперативную проверку устройств.

13.7.1. Формат тестового сообщения для стартстопного режима

99999	XX	ТИП	АДРЕС АП	АДРЕС КОМПОНЕНТА	ТЕКСТ	Символ конца сообщения
Длина 5	2	1	1 или 2	1 или 2	<i>n</i>	1

где 99999 — идентифицирует запрос на тест;

XX — тип теста:

01 — коммутация сообщений;

05 — сравнение;

06 — полный набор графических символов;

08 — писать по адресу строки (АП-61, АП-62, АП-63, АП-64);

09 — запрос адреса (АП-61, АП-62, АП-63, АП-64);

ТИП — тип АП:

0 — АП-70 со средствами прерывания;

2 — АП-1;

4 — АП-70;

6 — АП-61, АП-62, АП-63, АП-64;

АДРЕС АП — указывает адрес АП (в шестнадцатеричном коде):

однобайтовый

АП-70 в режиме управления станцией — символ выборки;

АП-70 без режима управления станцией — пробел;

АП-1 — символ выборки;

двухбайтовый

АП-61 — АП-64; адрес устройства управления кодируется следующим образом:

Фактический адрес

Код

0100000

01

0100001

02

1111111

96

АДРЕС КОМПОНЕНТА:

АП-61—АП-64

Адрес компонента кодируется следующим образом:

Фактический адрес	Код	
1 000000	01	Индикаторы и пишущая
1 000001	02	машинка (АП-63, АП-64)
1 011 000	25	Индикатор (АП-61, АП-62)
1 011 001	26	Пишущая машинка (АП-61)
1 011 010	27	

Символ конца сообщения — КП (для АП-1 и АП-70) или КТ (для АП-61 и АП-63).

13.7.2. Форматы запроса тестов для синхронных каналов связи (кроме ЕС-7920)

Тип теста 00 и 01

НТ%	X	Y	N	АДРЕС	ТЕКСТ
2	2	2	1	0—9	

Тип теста 06 и выше

НТ%	X	Y	N	АДРЕС	КТ
2	2	2	1	0—9	1

где НТ% — идентифицирует запрос теста;

X — определяет тип теста (00, 01, 06, 07, 08, 09);

Y — задает число посылок (01—99) тестового сообщения; для АП-2 и АП-4, если X=00, то Y должен быть равен 01;

N — указывает длину в байтах (0—9) поля АДРЕС;

АДРЕС — содержит адрес станции или ее компонента. Для каналов связи в режиме подчинения поле АДРЕС содержит адресную последовательность, состоящую из удвоенного символа адреса АП и удвоенного символа адреса компонента;

ТЕКСТ — данные с обрамляющими символами, посылаемые для типов тестов 01. Для непрозрачного режима символы данных должны обрамляться символами НТ и КТ, для прозрачного — символами АР1 НТ и АР1 КТ.

Типы тестов:

00 — запрашивающая станция вслед за запросом теста посылает тестовое сообщение или несколько тестовых сообщений в зависимости от значения Y (от 1 до 99). Если запрашивающей станцией является АП-2 или АП-4, то тестовое сообщение посылается в составе запроса теста, а в поле Y должна стоять единица;

01 — запрашивающая станция посылает запрос на тест, включающий тестовое сообщение. Станцией, принимающей запрос на тест, может быть только ЭВМ. Станция, при-

- нявшая запрос на тест, посылает обратно тестовое сообщение столько раз, сколько указано в поле Y;
- 06 — запрос сообщения из символов латинского алфавита и цифр. Этот запрос может быть выдан с ЭВМ или АП-4;
- 07 — запрос сообщения из символов латинского алфавита и цифр. Это сообщение посылается на пишущую машину АП-2. Может быть также послано к ЭВМ или АП-4;
- 08 — запрос сообщения из символов латинского алфавита и цифр. Это сообщение посылается на перфоленточное устройство АП-2. Может быть послано к ЭВМ или АП-4;
- 09 — запрос сообщения из символов латинского алфавита и цифр. Это сообщение посылается одновременно на перфоленточное устройство и пишущую машинку АП-2. Может быть послано к ЭВМ или АП-4.

13.7.3. Формат тестового сообщения для локальных дисплеев ЕС-7920

НЗ%	/	НТ	Х	Y	N	АДРЕС
2	1	1	2	2	1	3

- где НЗ% — идентифицирует запрос теста;
- НТ — управляющий символ начала текста;
- Х — определяет тип теста (50, 51, 52, 54, 55);
- Y — задает число повторений теста;
- N — указывает длину в байтах поля АДРЕС; N должно быть равно 3;

АДРЕС — трехзначный шестнадцатеричный адрес устройства ввода-вывода.

Типы тестов:

- 50 — проверяется отображение или печать (40 символов в строке) всех алфавитно-цифровых символов, а также действие бита звукового сигнала в СУЗ и функции символа атрибута (ЕС-7920-00; ЕС-7920-01);
- 51 — проверяется каждая позиция устройства отображения ЕС-7927—00, действие бита звукового сигнала в СУЗ; при тестировании печатающего устройства печатается 40 символов в строке;
- 52 — проверяется каждая позиция устройства отображения ЕС-7927—01, действие бита звукового сигнала в СУЗ; при тестировании печатающего устройства печатается 80 символов в строке;
- 53 — проверяется действие указаний НАЧАЛО ПОЛЯ (НП), УСТАНОВИТЬ АДРЕС БУФЕРА (УАБ) и др.; проверяется действие бита звукового сигнала в СУЗ; символы отображаются с повышенной яркостью; при тестиро-

вании печатающего устройства печатается 64 символа в строке;

54 — проверяется печатающее устройство комплексов ЕС-7920—00, ЕС-7920—01;

55 — проверяется печатающее устройство комплексов ЕС-7920—00, ЕС-7920—01, проверяются действия указаний НОВАЯ СТРОКА (НС), конец сообщения (КС).

13.7.4. Формат тестового сообщения для удаленных дисплеев ЕС-7920

НЗ%	/	НТ	Х	У	Н	АДРЕС АП	КТ
2	1	1	2	2	1	0 или 4	1

где НЗ% / — идентифицирует запрос теста;

НТ — управляющий символ начала текста;

Х — определяет тип теста (56, 57, 58, 59, 60, 61);

У — задает число повторений теста;

Н — указывает длину в байтах поля АДРЕС АП; для коммутируемых каналов связи Н должно быть равно 0, для некоммутируемых каналов связи Н равно 4;

АДРЕС АП — удвоенные символы адреса АП и адреса компонента;

КТ — символ конца текста.

Типы тестов:

тесты типов 56—61 по функциям аналогичны соответственно тестам типов 50—56 для локальных устройств ЕС-7920.

13.7.5. Макрокоманда ONLTST

Макрокоманда ONLTST посылает запрос на тест для синхронных каналов связи удаленным АП или ЭВМ, в которой под управлением БТМД в настоящий момент выполняется макрокоманда READ TI или READ TC. Удаленным АП можно посылать запрос только с типом 00. Формат:

[метка] ONLTST DECB=адрес-decb, X=тип, Y=число, DCB=адрес-dcb, AREA=адрес-области, TEXT=адрес-текста, LENGTH=длина-текста, ENTRY=адрес-списка, RLN=номер]

где адрес-decb — адрес соответствующего блока DECB;

X=тип — задает тип теста;

Y=число — задает число передач тестового сообщения (от 1 до 99);

DCB=адрес-dcb — адрес соответствующего блока DCB;

AREA=адрес-области — адрес области, в которой БТМД формирует запрос на тест и куда пересылаются тестовые сообщения; для тестов типа 00 и 01 длина этой области должна быть достаточной, чтобы принять тестовое сообщение, для других типов длина этой области должна быть не менее 300 байт;

TEXT=адрес-текста — адрес тестового сообщения для типов 00 и 01; текст обрамляется символами НТ и КТ (для прозрачного режима — символами APIHT и APIKT);

LENGTH=длина-текста — число символов в сообщении, адрес которого определен операндом TEXT;

ENTRY=адрес-списка — адрес абонентского списка (OPENLST, DIALST или BSCLST), состоящего из одного элемента; для коммутируемых каналов связи нужно задавать список вызова;

RLN=номер — задает относительный номер канала связи в группе.

Коды возврата (десятичные):

00 — успешное выполнение;

04 — канал связи занят;

08 — неверный относительный номер канала связи;

12 — неверный тип теста;

16 — элемент абонентского списка заблокирован;

20 — ошибка в канале связи во время открытия группы каналов связи.

13.8. Особенности программирования для различных АП

13.8.1. АП-1 (ЕС-8501)

Программист готовит в буфере для отправки блок данных, заканчивающийся символом КБ. Символ НТ посылается методом доступа и не входит в размер, определяемый в макрокоманде WRITE. После чтения блока данных в основную память помещаются символы НТ и КБ. Число считанных байт данных включает в себя эти символы.

Некоммутируемый канал связи. Работа с АП-1 на некоммутируемом канале связи ведется с использованием списков опроса и выборки (см. 13.1.2).

Опрос начинается макрокомандой READ TI (или READ TIR со сбросом), которая посылает символы опроса, читает ответ и, если он положительный, читает блок данных. Абонентский пункт ждет подтверждения. Макрокоманда READ TIR посылает символы НТ КП КП КП, которые интерпретируются абонентским пунктом как положительное подтверждение и переводят его в режим управления (т. е. производят сброс). В режиме управления абонентский пункт ожидает следующих символов опроса или выборки.

Макрокоманда READ TT (или READ TTR со сбросом) посылает положительный ответ и читает следующий блок данных. Макрокоманда READ TP (или READ TPR со сбросом) посылает отрицательный ответ и повторно читает блок данных.

Выборка начинается макрокомандой WRITE TI (или WRITE TIR со сбросом), которая посылает символы выборки, читает ответ и, если он положительный, посылает символ НТ, блок данных и читает ответ. Макрокоманда WRITE TIR, кроме того, посылает символы КП КП КП, переводящие АП в режим управления (т. е. осуществляющие сброс).

Макрокоманда WRITE TT (или WRITE TTR со сбросом) посылает блок данных и читает ответ.

Макрокоманда WRITE TTV объединяет функции макрокоманд WRITE TT и READ TI.

Макрокоманда WRITE TA сбрасывает АП с положительным подтверждением (т. е. посылает символы HT КП КП КП).

Макрокоманда WRITE TN сбрасывает АП с отрицательным подтверждением (т. е. посылает символы КП КП КП).

Макрокоманда READ TS считывает данные, не помещая их в основную память.

Коммутируемый канал связи. Работа по коммутируемому каналу связи с АП-1 ведется с использованием списков ответа или вызова (см. 13.1.2).

В случае инициативы со стороны АП используется список ответа. Макрокоманда READ TI (или READ TIR со сбросом) посылает символы опроса, читает ответ и, если ответ положительный, читает блок данных. Макрокоманда READ TIR, кроме того, посылает символы HT КП и выключает канал связи (разъединяет ЭВМ и АП). Макрокоманда WRITE TI (или WRITE TIR со сбросом) посылает символы выборки, читает ответ и, если ответ положительный, посылает символ HT и блок данных, затем читает ответ. Макрокоманда WRITE TIR, кроме того, посылает символы КП КП КП и выключает канал связи.

В случае инициативы со стороны ЭВМ используется список вызова. Макрокоманда READ TI (или READ TIR со сбросом) набирает телефонный номер АП, посылает символы выборки, читает ответ и, если ответ положительный, читает блок данных (READ TIR, кроме того, выключает канал связи). Макрокоманда WRITE TI (или WRITE TIR со сбросом) набирает телефонный номер АП, посылает символы выборки, читает ответ и, если ответ положительный, посылает символ HT и блок данных, затем читает ответ (WRITE TIR, кроме того, выключает канал связи).

После установления связи можно выдавать макрокоманды: READ TT (TTR), WRITE TT(TTR), READ TP (TPR), WRITE TA, WRITE TN.

Макрокоманда READ TV (или READ TVR со сбросом), используя список опроса DIALST, посылает символы опроса, читает ответ и, если он положительный, читает блок данных. Эта макрокоманда используется при изменении направления передачи без разрыва соединения.

Макрокоманда WRITE TV (или WRITE TVR со сбросом), используя список выборки DIALST, посылает символы HT КП КП КП (положительное подтверждение) и символы выборки, читает ответ и, если ответ положительный, посылает блок данных и читает ответ. Эта макрокоманда используется при изменении направления передачи без разрыва соединения.

Макрокоманда WRITE TTV объединяет функции макрокоманд WRITE TT и READ TV, используя список опроса.

13.8.2. АП-2 (ЕС-8502) и АП-3 (ЕС-8503) (стартстопные)

Абонентские пункты АП-2 и АП-3 (стартстопные) имеют одинаковый алгоритм работы и используют коммутируемые и некоммутируемые каналы связи.

Чтение производится с пишущей машинки или перфоленты в зависимости от действий оператора. Считываемый или посылаемый блок должен заканчиваться символом КП.

Некоммутируемый канал связи. Функции макрокоманды:

READ TI наблюдает за каналом связи и при появлении данных считывает их в основную память.

READ TS считывает данные, не помещая их в основную память. Используется для очистки канала связи после появления ошибки ПОТЕРЯ ДАННЫХ.

WRITE TI (или WRITE TIR со сбросом) посылает данные абонентскому пункту (WRITE TIR, кроме того, выключает канал связи).

WRITE TN прерывает связь с АП.

WRITE TD используется для возобновления связи с АП (включает канал связи). После этого необходимо в течение 20 с выдать следующую макрокоманду, иначе придется заново выдать макрокоманду WRITE TD.

Коммутируемый канал связи. Для работы по коммутируемому каналу связи используются списки ответа и вызова (см. 13.1.2). Функции макрокоманды:

READ TI включает канал связи, после поступления вызова следит за каналом связи и считывает поступающие данные в основную память.

WRITE TI (или WRITE TIR со сбросом) набирает телефонный номер АП (номер телефона и X'OC'), посылает блок данных (WRITE TIR, кроме того, разъединяет канал связи).

READ TV посылает символ КП, после чего наблюдает за каналом связи и считывает поступающие данные.

READ TS считывает остаток данных, очищая канал связи, причем данные не передаются в основную память.

WRITE TV (WRITE TVR со сбросом) посылает данные абонентскому пункту (WRITE TVR, кроме того, разрывает соединение).

WRITE TN посылает абонентскому пункту символы AP1, КП, которые разрывают соединение и отключают канал связи.

13.8.3. АП-2 (ЕС-8502) синхронный

Синхронный абонентский пункт АП-2 использует некоммутируемые и коммутируемые телефонные и телеграфные каналы связи. Формат блока данных:

НЗ (или НТ) данные КБ (или КТ)

Длина блока — 15, 30 или 45 символов (включая управляющие). Программист должен готовить для передачи блок в указанном формате.

При передаче данных абонентскому пункту в первом блоке данных после символа НЗ или НТ нужно указать двухсимвольную адресную последовательность:

AP2 1 — вывод на пишущую машинку;

AP2 2 — вывод на перфоленту;

AP2 5 — вывод на пишущую машинку и перфоленту.

При приеме данных с перфоленты или пишущей машинки goтoятcя все управляющие символы, за исключением символа КБ, который автоматически формируется устройством управления.

Некоммутируемый канал связи. Функции макрокоманды:

READ TI наблюдает за каналом связи, читает KTM, посылает ответ и читает блок данных.

READ TIQ наблюдает за каналом связи и читает символ KTM.

READ TT посылает положительный ответ и читает блок данных (или КП).

READ TP посылает отрицательный ответ и читает блок данных.

READ TQ читает KTM.

READ TRV посылает символы AP1<, сигнализируя удаленной станции, что центральная ЭВМ приостанавливает прием данных, и затем читает КП или данные. Прием КП свидетельствует о том, что абонентский пункт согласился приостановить передачу данных.

WRITE TI (или WRITE TIR со сбросом) посылает KTM и, если ответ на KTM положительный (ДА-0), посылает данные и читает ответ (WRITE TIV, кроме того, посылает КП).

WRITE TIV посылает KTM, читает ответ и, если ответ положительный (ДА-0), посылает блок данных и читает ответ или данные.

Макрокоманда WRITE TT (или WRITE TTR со сбросом) посылает блок данных и читает ответ (WRITE TTR, кроме того, посылает КП).

Макрокоманда WRITE TTV посылает блок данных и читает ответ или данные.

Макрокоманда WRITE TQ посылает KTM и читает ответ.

Макрокоманда WRITE TR посылает КП, чтобы закончить использование канала связи. Следующей макрокомандой должна быть READ TI(TIR) или WRITE TI(TIR).

Коммутируемый канал связи. Использование абонентских списков описано в п. 13.1.2. Функции макрокоманды:

READ TC (расширенная проверка ИД, список автоматических ответов) читает ИД KTM и затем сравнивает прочитанный ИД с заданными в абонентском списке. Если таковой находится, адрес этого элемента помещается в первое слово списка. Если управляющий байт равен 0, то посылается ИД ДА-0 (или только ДА-0) и читается блок данных. Если управляющий байт равен 1, соединение разрывается, и канальная программа повторяется сначала. Если управляющий байт равен 2, то отмечается нормальное завершение — X'7F'. Если ИД не найден, то при принятии только KTM

отмечается нормальное завершение. Если принята AP1 КП, включается бит 1 в поле DECFLAGS и отмечается нормальное завершение. Если принят неправильный ИД, то команда ЧИТАТЬ ИД КТМ повторяется до семи раз, после чего соединение разрывается, бит 3 в поле DECFLAGS устанавливается в 1, и в блоке ЕСВ отмечается нормальное завершение.

READ TP посылает положительный ответ и читает блок данных.

READ TRV посылает отрицательный ответ и читает блок данных.

WRITE TRV посылает двухсимвольную последовательность AP1<, чтобы приостановить прием данных, затем читает данные или КП.

WRITE TI (список автоматических вызовов SWLST) набирает номер АП, посылает ИД КТМ, читает ИД ДА-0 и, если ответ положительный, посылает блок данных и читает ответ. Если ИД АП не соответствует данному абонентскому списку, в блоке ЕСВ отмечается завершение операции.

WRITE TC (ручной вызов WTLIST) посылает ИД КТМ, читает ИД ДА-0. Если принятая ИД не соответствует ожидаемой, отмечается ошибка.

WRITE TC (расширенная проверка ИД, список автоматических или ручных вызовов SWLST) набирает номер (операнд AD в макрокоманде DFTRMLST) или включает канал связи для ручного соединения (операнд MD в макрокоманде DFTRMLST), посылает ИД КТМ и читает ИД ДА-0 или ИД НЕТ. Если прочитано ДА-0 (без ИД) или ИД ДА-0, отмечается нормальное завершение. В случае ответа НЕТ (без ИД), бит 1 поля DECFLAGS устанавливается в 1 и отмечается нормальное завершение. Если прочитано ИД НЕТ или ИД ДА-0, то в первое слово списка помещается адрес элемента, содержащего этот ИД. В случае ИД НЕТ проверяется управляющий байт данного элемента, и если он равен 0, то бит 1 поля DECFLAGS устанавливается в 1, в противном случае повторно посылается ИД КТМ (или КТМ).

Если ответом является неверный ИД, БТМД повторяет команду ПИСАТЬ ИД КТМ (до семи раз). При неуспешных повторениях соединение разрывается, бит 3 поля DECFLAGS устанавливается в единицу, и в блоке ЕСВ отмечается нормальное завершение. В случае ответа НЕТ (без ИД) устанавливается бит 1 поля DECFLAGS и отмечается нормальное завершение.

Если ответа от удаленной станции нет, то повторяется (до семи раз) команда ПИСАТЬ ИД КТМ, и если повторения оказались неуспешными, соединение разрывается, в поле DECSO устанавливается X'01' и в ЕСВ отмечается код завершения X'41'.

WRITE TT посылает блок данных и читает ответ.

WRITE TTV посылает блок данных и читает ответ или данные.

WRITE TQ посылает КТМ и читает ответ.

WRITE TR посылает КП и читает ответ.

WRITE TB выключает канал связи (т. е. разрывает соединение).

WRITE TD посылает символы AP1 КП, сообщая тем самым удаленной станции о необходимости разрыва соединения, после чего выключает канал связи.

13.8.4. АП-4 (ЕС-8504)

Абонентский пункт АП-4 содержит устройство управления. К нему подключаются устройства ввода-вывода, логическая связь с которыми устанавливается символами опроса или выборки в следующем формате:

АдрАП АдрАП АдрУВВ АдрУВВ КТМ

где АдрАП — символ, являющийся адресом АП;

АдрУВВ — символ, являющийся адресом устройства ввода-вывода;

КТМ — символ КТО ТАМ.

Определение терминальных списков для АП-4 см. в 13.1.2. Формат блока для АП-4

НЗ или НТ	текст	РИ-2	КБ или КТ
-----------------	-------	------	-----------------

где РИ-2 вставляется для вывода на пишущую машинку или печатающее устройство.

Размер блока для АП-4 не должен превышать 132 символов (для вывода на АЦПУ — не более 128 символов).

Функции макрокоманд:

READ TI (список опроса AUTOLST или AUTWLST) автоматически опрашивает каждую станцию и, если от какой-либо станции получен положительный ответ, во входную область считывается индексный байт и затем данные.

READ TT посылает положительный ответ и читает блок данных.

READ TP посылает отрицательный ответ и читает блок данных.

READ TQ читает КТМ.

READ TRV посылает символы AP1<, чтобы приостановить передачу данных от АП, и читает данные.

WRITE TI (TIR) посылает символы выборки и, если ответ на выборку положительный, посылает блок данных и читает ответ. Макрокоманда WRITE TIR, кроме того, посылает КП.

WRITE TIV посылает символы выборки и, если ответ на выборку положительный, посылает блок данных и читает ответ или данные.

WRITE TT (TTR) посылает блок данных и читает ответ. Макрокоманда WRITE TTR, кроме того, посылает КП.

WRITE TTV посылает блок данных и читает ответ или данные.
WRITE TQ посылает KTM и читает ответ.
WRITE TR посылает КП.

13.8.5. АП-5 (ЕС-8505)

Абонентский пункт АП-5 состоит из устройства управления и перфоленточных устройств ввода-вывода.

Формат блока для АП-5:

входной

НТ КА1 КА2 ИД АД ТЕКСТ КТ

выходной

НТ КА1 КА2 РЭ ТЕКСТ КТ

где КА1 — символ для адреса канала 1 (А-0);

КА2 — символ для адреса канала 2 (А-0);

РЭ — разделитель элементов (X'1F');

ИД — идентификатор;

АД — условный адрес программы.

Если канал 1 или 2 не используются, то вместо КА1 или КА2 кодируется X'40'. Для заполнения текста холостыми символами, например после символа «Возврат каретки», используется символ СИН (X'16').

Функции макрокоманд:

WRITE TI посылает блок данных и читает ответ.

WRITE TA посылает символ ДА и читает ответ.

WRITE TN посылает символ НЕТ и читает ответ.

WRITE TQ посылает KTM (указывая, что ответ содержит ошибку) и читает ответ.

READ TI(TIR) посылает КП и читает ответ или данные.

Макрокоманда READ TIR, кроме того, посылает ДА и читает КП или KTM.

READ TT(TTR) посылает ДА и читает данные. Макрокоманда READ TTR, кроме того, посылает ДА и читает КП или KTM.

READ TP(TPR) посылает НЕТ и читает данные. Макрокоманда READ TPR, кроме того, посылает ДА и читает КП или KTM.

13.8.6. АП-61, АП-62, АП-63, АП-64 (ЕС-8561, ЕС-8562, ЕС-8563, ЕС-8564)

Абонентские пункты АП-61 и АП-62 имеют в своем составе устройство управления, один индикатор и пишущую машинку. Абонентские пункты АП-63 и АП-64 являются групповыми и имеют в своем составе устройство управления, несколько (до 16) индикаторов и пишущую машинку.

Абонентский пункт АП-61 имеет две модели: 1 и 2. Модель 1 работает в синхронном режиме, модель 2 — в стартстопном. Абонентский пункт АП-63 может работать как в синхронном, так и в стартстопном режиме. Абонентские пункты АП-62 и АП-64 работают только в стартстопном режиме.

Логическая связь с устройством ввода-вывода осуществляется символами опроса и выборки, которые указываются в макрокоманде DFTRMLST (см. 13.1.2) и состоят из символа адреса АП и символа адреса УВВ. Для синхронных АП-61 и АП-63 эти символы дублируются (см. соответствующее техническое описание АП). Символ общего опроса — X'7F'.

Код операции для устройств ввода-вывода (стартстопный режим) указывается типом макрокоманды в соответствии со следующей таблицей:

Операция	Тип макрокоманды	Код операции
Чтение начальное	TI	X'20'
Чтение буфера	TB	X'30'
Запись начальная	TI	X'40'
Запись по адресу строки	TL	X'50'
Запись с позиции (АП-62, АП-64)	TIX	X'58'
Чтение с позиции (АП-62, АП-64)	TV	X'28'
Запись со стиранием	TS	X'60'
Запись с частичным стиранием (АП-62, АП-64)	TV	X'70'

Для синхронных АП-61 и АП-63 используются только типы макрокоманд TI, TIR, TIV.

Сообщение, посылаемое к АП, имеет формат

НТ Текст КТ

Для стартстопных устройств символ НТ вставляется методом доступа. Для стартстопных устройств первый байт текста может содержать адрес строки, а второй (только для АП-62 и АП-64) — адрес колонки. Адреса строк кодируются следующим образом (в коде КОИ-7):

Строка	Код (КОИ-7)	Строка	Код (КОИ-7)
1	30	9	33
2	31	10	39
3	32	11	3A
4	33	12	3B
5	34	13	3C
6	35	14	3D
7	36	15	3E
8	37	16	3F

Адрес колонок кодируется следующим образом (только для АП-62 и АП-64):

Колонка	Код (КОИ-7)	Колонка	Код (КОИ-7)
1	20
2	21	79	6E
3	22	80	6F

Сообщения, посылаемые к АП, могут быть разбиты на блоки. В этом случае каждый блок, кроме последнего, заканчивается символом КБ. Последний блок заканчивается символом КТ. Защищенные области (только для АП-62 и АП-64) начинаются двухсимвольной последовательностью АР1-Н и заканчиваются символами АР1-1. Для определения начала и конца данных используются символы \neq и !.

Сообщение, принимаемое от АП, имеет формат

НТ Адр УУ Адр УВВ Текст [А11] КБ или КТ

АН, если вставлен перед КБ, свидетельствует о том, что во время передачи данных имела место ошибка и этот блок следует аннулировать.

Макрокоманды для стартстопных устройств

WRITE TI(TIR) посылает НТ, 15 символов КП, символы выборки, код команды ЗАПИСЬ, читает ответ на выборку и, если он положительный, посылает НТ, блок данных и читает ответ. Макрокоманда WRITE TIR, кроме того, посылает НТ и три символа КП.

WRITE TS(TSR) аналогична макрокоманде WRITE TI(TIR), только перед передачей данных полностью стирается экран.

WRITE TV(TVR) (только для АП-62 и АП-64) аналогична макрокоманде WRITE TS(TSR), однако стирается не весь экран, а только незащищенные области экрана.

WRITE TL(TLR) аналогична макрокоманде WRITE TI(TIR), только вместо кода ЗАПИСЬ посылается код ЗАПИСЬ ПО АДРЕСУ СТРОКИ.

WRITE TIX(TIXR) (только для АП-62 и АП-64) аналогична макрокоманде WRITE TI(TIR), только вместо кода ЗАПИСЬ посылается код ЗАПИСЬ С ПОЗИЦИИ.

WRITE TT(TTR) посылает НТ, блок данных и читает ответ. Макрокоманда WRITE TTR, кроме того, посылает НТ и 3 символа КП.

WRITE TA посылает НТ и 3 символа КП.

WRITE TN посылает 3 символа КП.

READ TI(TIR) посылает НТ, 15 символов КП, символы опроса, код команды ЧИТАТЬ; читает ответ на опрос, читает данные. Макрокоманда READ TIR, кроме того, посылает НТ и 3 символа КП.

READ TV(TVR) (только для АП-62 и АП-64) аналогична макрокоманде READ TI(TIR), только вместо кода ЧИТАТЬ посылается код ЧИТАТЬ С ПОЗИЦИИ.

READ TT(TTR) посылает символ ДА и читает блок данных. Макрокоманда READ TTR, кроме того, посылает НТ и 3 символа КП.

READ TB(TBR) аналогична макрокоманде READ TI(TIR), только вместо кода ЧИТАТЬ посылается код ЧИТАТЬ БУФЕР. С помощью этой команды считывается весь экран, в том числе и защищенные поля.

READ TP(TPR) посылает символ НЕТ и читает блок данных. Макрокоманда READ TPR, кроме того, посылает НТ и 3 символа КП.

Макрокоманды для синхронных устройств

Макрокоманды для синхронных АП-61 и АП-63 аналогичны макрокомандам для АП-4 со следующим замечанием.

Для операции ЧТЕНИЕ НАЧАЛЬНОЕ используется макрокоманда READ TI(TIR). Операция ЧТЕНИЕ БУФЕРА выполняется макрокомандой READ TIV, которая выбирает станцию и УВВ, посылает код операции ЧТЕНИЕ БУФЕРА и читает данные. Операции ЗАПИСЬ ПО АДРЕСУ СТРОКИ, ЗАПИСЬ СО СТИРАНИЕМ И ЗАПИСЬ НАЧАЛЬНАЯ выполняются макрокомандами WRITE TI(TIR) или WRITE TIV, где первым байтом данных посылается код операции.

13.8.7. АП-70 (ЕС-8570)

Абонентский пункт АП-70 может работать в нескольких режимах:

- без средств проверки;
- со средствами проверки на некоммутируемых каналах связи;
- со средствами проверки на коммутируемых каналах связи и с управлением передачей;
- со средствами управления станцией.

Максимальный размер строки для вывода на АП-70 — 105 символов. При выводе более длинных сообщений на АП-70 необходимо после символа ВК (ВОЗВРАТ КАРЕТКИ) вставлять символы-заполнители ПУСТО (X'00'), число которых зависит от длины обратного кода каретки (т. е. длины предыдущей строки). Кроме того, следует вставить заполнитель после каждого символа ВШ (ВОЗВРАТ НА ШАГ).

АП-70 без средств проверки

Функции макрокоманды:

READ TI посылает КП, следит за каналом связи и читает НТ и данные.

WRITE TI(TIR) посылает НТ и 25 символов ПУСТО, посылает данные. Макрокоманда WRITE TIR, кроме того, посылает КП.

АП-70 со средствами проверки на некоммутируемых каналах связи

Функции макрокоманды:

READ TI(TIR) посылает КП, следит за каналом связи и читает НТ и данные. Макрокоманда READ TIR, кроме того, посылает НТ КП.

READ TT(TTR) посылает ДА и читает блок данных. Макрокоманда READ TTR, кроме того, посылает НТ КП.

READ TP(TPR) посылает НЕТ и читает блок данных. Макрокоманда READ TPR, кроме того, посылает НТ КП.

WRITE TI(TIR) посылает НТ, 25 символов ПУСТО, блок данных и читает ответ. Макрокоманда WRITE TIR, кроме того, посылает КП.

WRITE TT(TTR) посылает блок данных и читает ответ. Макрокоманда WRITE TTR, кроме того, посылает КП.

WRITE TV(TVR) посылает HT, блок данных и читает ответ. Макрокоманда WRITE TVR, кроме того, посылает КП.

WRITE TTV посылает блок данных, читает ответ, посылает КП, ждет блок данных и читает его.

WRITE TA посылает HT КП для прекращения передачи по каналу связи.

WRITE TN посылает КП, что сигнализирует об ошибке при приеме данных.

АП-70 со средствами проверки на коммутируемых каналах связи.

Функции макрокоманды:

READ TI(TIR) включает канал связи и ждет вызова от АП, после чего читает данные. Макрокоманда READ TIR, кроме того, посылает HT КП и выключает (разъединяет) канал связи.

READ TT (TTR) — см. выше. При TTR в конце операции канал связи выключается.

READ TP(TPR) — см. выше. При TPR в конце операции канал выключается.

READ TV(TVR) посылает КП и ждет приема данных, после чего читает блок данных. Макрокоманда READ TVR, кроме того, посылает HT КП и выключает канал связи.

WRITE TI(TIR) набирает номер АП, посылает 25 символов ПУСТО, посылает HT, данные и читает ответ. Макрокоманда WRITE TIR, кроме того, посылает КП и выключает канал связи.

WRITE TT(TTR) посылает блок данных и читает ответ. Макрокоманда WRITE TTR, кроме того, посылает КП и выключает канал связи.

WRITE TV(TVR) — см. предыдущий раздел. При TVR в конце операции выключает канал связи.

WRITE TTV — см. предыдущий раздел.

WRITE TA посылает HT КП и выключает канал связи.

WRITE TN посылает КП и выключает канал связи.

АП-70 со средствами проверки на коммутируемых каналах связи и с управлением передачи.

Функции макрокоманды:

READ TI(TIR) со списком вызовов набирает номер АП, посылает 25 символов ПУСТО, посылает символы выборки, читает ответ и данные. Макрокоманда READ (TIR), кроме того, посылает HT КП и выключает канал связи.

READ TI(TIR) со списком ответа включает канал связи, посылает 25 символов ПУСТО, посылает символ выборки, читает ответ и данные. Макрокоманда READ TIR, кроме того, посылает HT КП и выключает канал связи.

READ TT(TTR) — см. предыдущий раздел.

READ TP(TPR) — см. предыдущий раздел.

READ TV(TVR) посылает символы выборки и, если ответ по-

ложительный, читает блок данных. Макрокоманда READ TVR, кроме того, посылает HT КП и включает канал связи.

WRITE TI(TIR) со списком вызовов набирает номер АП, посылает 25 символов ПУСТО, посылает HT, данные, читает ответ. Макрокоманда WRITE TIR, кроме того, посылает КП и включает канал связи.

WRITE TI(TIR) со списком ответов включает канал связи, посылает 25 символов ПУСТО, посылает HT, данные и читает ответ. Макрокоманда WRITE TIR, кроме того, посылает КП и выключает канал связи.

WRITE TT(TTR) — см. выше.

WRITE TV(TVR) — см. выше.

WRITE TA — см. выше.

WRITE TN — см. выше.

АП-70 со средствами управления станцией

Функции макрокоманды:

READ TI(TIR) посылает три символа КП, символ опроса, ПРОБЕЛ, читает ответ и, если он положительный, читает блок данных. Макрокоманда READ TIR, кроме того, посылает HT и 3 символа КП.

READ TT(TTR) посылает ДА и читает блок данных. Макрокоманда READ TTR, кроме того, посылает HT и 3 символа КП.

READ TP(TPR) посылает НЕТ и читает блок данных. Макрокоманда READ TPR, кроме того, посылает HT и 3 символа КП.

WRITE TI(TIR) посылает три символа КП, символ выборки, ПРОБЕЛ, читает ответ, посылает HT, данные, читает ответ. Макрокоманда WRITE TIR, кроме того, посылает 3 символа КП.

WRITE TT(TTR) посылает блок данных и читает ответ. Макрокоманда WRITE TTR, кроме того, посылает 3 символа КП.

WRITE TA посылает HT и 3 символа КП.

WRITE TN посылает 3 символа КП.

В случае автоопроса в область данных первым байтом считается индекс, указывающий АП.

13.8.8. Телеграфный аппарат (ТА)

Данные, посылаемые с ТА, могут быть разбиты на блоки. В качестве символа КБ применяется $/(X'3737')$, в качестве символа КП используется $=(X'2F2F')$.

Некоммутируемый канал связи

Функции макрокоманды:

READ TI наблюдает канал связи и читает данные.

READ TT читает данные.

READ TE посылает символ ЛАТ и n символов $X'DF'$ (n равно значению операнда MONDLY в макрокоманде DCB), идентификатор ЭВМ, КТМ, читает идентификатор ТА и данные.

WRITE TI посылает символ ЛАТ и n символов $X'DF'$, 12 символов ЛАТ и данные.

WRITE TT посылает ЛАТ, символы $X'DF'$ и данные.

Коммутируемый канал связи

Функции макрокоманды:

WRITE TI включает канал связи, посылает КТМ, читает идентификатор ТА, читает данные.

WRITE TIE набирает номер ТА, читает идентификатор ТА и данные.

WRITE TI посылает ЛАТ и символы X'DF', 12 символов ЛАТ и данные.

13.8.9. Комплексы ЕС-7920

Общие сведения

Комплексы ЕС-7920 состоят из устройства управления, к которому подключены одно или несколько устройств отображения и печатающих устройств. Комплексы могут подключаться локально к ЭВМ или через канал связи (коммутируемый или некоммутируемый).

С помощью макрокоманд БТМД программист может:

читать модифицированные поля после действий оператора;

читать модифицированные поля;

читать модифицированные поля с позиции;

читать буфер;

читать буфер с позиции;

записать в буфер;

стереть и записать в буфер;

стереть незащищенные поля;

копировать (только для некоммутируемого канала связи).

Формат считываемых данных

а) при чтении модифицированных полей:

ин- декс	ИТ	Адр УУ	Адр УВВ	ИВ	Адрес курсора	УАВ	Адрес буфер- ной памяти	Текст	{КБ} {КТ}
-------------	----	-----------	------------	----	------------------	-----	----------------------------------	-------	--------------

где индекс — включается только в первый блок при установлении соединения по некоммутируемому каналу связи; для локальных ЕС-7920 и удаленных ЕС-7920 на коммутируемых каналах связи это поле отсутствует;

ИТ — символ начала текста; включается в данные, считываемые по каналу связи; для локальных ЕС-7920 это поле отсутствует;

Адр УУ, Адр УВВ — адрес устройства управления и устройства ввода-вывода; включается только в первый блок при установлении соединения по некоммутируемому каналу связи; для локальных ЕС-7920 и удаленных ЕС-7920 на коммутируемых каналах связи это поле отсутствует;

ИВ — идентификатор внимания (см. ниже);

Адрес курсора — двухбайтовый адрес положения курсора. Включается только в первый блок считываемых данных.

УАБ, Адрес буферной памяти — указатель адреса буфера и адрес буферной памяти. Включаются в считываемые данные только в случае форматизированной буферной памяти.

Текст — данные. Это поле отсутствует, если сигнал внимания пришел от фотоселектора;

КБ или КТ — символ конца блока или текста. Включается в данные, считываемые по каналу связи.

Если данные считываются после нажатия оператором клавиши «СТРН ЭКР» или «ПД1—ПД3», то адрес курсора, УАБ, адрес буферной памяти и текст не входят в считываемые данные.

Если нажата клавиша «Вызов теста», то поток данных имеет следующий формат:

Индекс	НЗ	%	/	НТ	текст	КТ	некоммутируемый канал связи
НЗ	%	/	НТ	текст	КТ		коммутируемый канал связи
НЗ	%	/	НТ	текст			локальное подклю- чение

Байты состояния, считываемые по каналу связи, содержатся в блоке следующего формата:

Индекс	НЗ	%	R	НТ	Адр УУ	Адр УВВ	Байты состояния	КТ
--------	----	---	---	----	--------	---------	--------------------	----

Для коммутируемого канала связи поле индекса отсутствует:
б) при чтении буфера:

Индекс	НТ	Адр УУ	Адр УВВ	ИВ	Адрес курсора	НП	Атри- бут	Текст	{КБ} {КТ}
--------	----	-----------	------------	----	------------------	----	--------------	-------	--------------

где индекс, НТ, Адр УУ, Адр УВВ, Адрес курсора — см. п. а);

НП — символ начала поля;

Атрибут — атрибут, характеризующий поле (см. ниже).

НП и атрибут включаются в считываемые данные только в случае форматизированной буферной памяти.

Данные от удаленных комплексов считываются блоками. Длина блока устанавливается аппаратно и может быть равна 16, 32, 64, 128 и 256 байт.

Формат посылаемых данных

НТ	АР2	код	СУЗ	указания и текст	КТ	для каналов связи
			СУЗ	указания и текст		для локального подключения

где НТ — символ начала текста;

AP2 — управляющий символ;

код — задает тип операции для удаленных комплексов:

1 — ПИСАТЬ;

2 — ЧИТАТЬ БУФЕР;

5 — СТЕРЕТЬ И ЗАПИСАТЬ В ПОЗИЦИЮ;

6 — ЧИТАТЬ МОДИФИЦИРОВАННЫЕ ПОЛЯ;

7 — КОПИРОВАТЬ;

? — СТЕРЕТЬ НЕЗАЩИЩЕННЫЕ ПОЛЯ;

СУЗ — символ управления записью (см. ниже);

указания и текст — указания (например, УАБ) и текст (например, адрес буферной памяти, или данные);

КТ — символ конца текста.

Байт ИВ — идентификатор внимания

Причина	Код (ДКОИ)	Код (КОИ-7)	Символ	Передается в ЭВМ
Отсутствие сигнала внимания	60	2D	—	Код ИВ, адрес курсора, УАБ, адрес буфера, текст для каждого модифицированного поля. Символы с кодом X'00' не передаются
Команда СЧТМ выдана для печатающего устройства	E8	59	Y	То же
Клавиша ВВОД	7D	27	'	"
Клавиша ПФ1	F1	31	1	"
Клавиша ПФ2	F2	32	2	"
Клавиша ПФ3	F3	33	3	"
Клавиша ПФ4	F4	34	4	"
Клавиша ПФ5	F5	35	5	"
Клавиша ПФ6	F6	36	6	"
Клавиша ПФ7	F7	37	7	"
Клавиша ПФ8	F8	38	8	"
Клавиша ПФ9	F9	39	9	"
Клавиша ПФ10	7A	3A	:	"
Клавиша ПФ11	7B	23	#	"
Клавиша ПФ12	7C	40	@	"
Внимание фотоселектора	7E	3D	=	Код ИВ и адреса модифицированных полей (включая УАБ)
Клавиша ПД1	6C	25	%	Код ИВ
Клавиша ПД2	6E	3E	>	То же
Клавиша ПД3	6B	2C	>	"
Клавиша СТРН экр	6D	5F	—	"
Клавиша ВЫЗОВ ТЕСТА	F0	30	0	НЗ, %, /, НТ, текст

Формат символа атрибута (в коде ДКОИ)

Биты

Значение

0—1 Дополняются для представления некоторого графического символа

2 0 — незащищенное поле; 1 — защищенное поле

- 2—3 11— автоматический пропуск курсором защищенного поля
 3 0— алфавитно-цифровое поле; 1 — цифровое (автоматически переводит регистр клавиатуры)
 4—5 00— отображаемое, не определяемое фотоселектором; 01— отображаемое, определяемое фотоселектором; 10— повышенной яркости, определяемое фотоселектором; 11— не отображаемое, не печатаемое, не определяемое фотоселектором
 6 Резервируется
 7 0— поле не модифицировано; 1— поле модифицировано оператором

Примечание. Символу-атрибуту, описывающему поле, которое определяется фотоселектором, должны предшествовать три символа «пробел» или три символа ПУС (X'00'). За символом-атрибутом должны следовать символ-определитель и отображаемые данные. Символ-определитель «?» обозначает поле выборки. Символ-определитель «_» (пробел) или ПУС обозначает поле внимания. Поле выборки служит для изменения фотоселектором бита модификации в символе-атрибуте (при этом символ «?» меняется на «>» и наоборот). Поле внимания служит для генерации сигнала внимания.

Формат символа управления записью (СУЗ) (в коде ДКОИ)

Биты	Значение
0—1	Дополняются для представления некоторого графического символа
2—3	Определяют длину печатаемой строки: 00— длина задается указанием НС 01— длина равна 40 знакам 10— длина равна 64 знакам 11— длина равна 80 знакам
4	Бит начала печати
5	Бит звукового сигнала
6	Бит разблокировки клавиатуры
7	Бит сброса разрядов модификации в символах-атрибутах

Примечание. Для операции копирования биты 6 и 7 имеют следующий смысл: 00— копируются только символы-атрибуты; 01— копируются символы-атрибуты и незащищенные алфавитно-цифровые поля (включая X'00'). Вместо алфавитно-цифровых символов защищенных полей передаются символы X'00'; 10— копируются символы-атрибуты и защищенные алфавитно-цифровые поля (включая X'00'). Вместо алфавитно-цифровых символов незащищенных полей передаются символы X'00'; 11— копируется все содержимое буферной памяти.

Адресация буферной памяти

Адрес буферной памяти занимает 11 бит. Старшие 5 бит помещаются в первый байт, младшие 6 бит — во второй байт адреса. Байты адреса слева дополняются битами таким образом, чтобы они представляли графические символы в коде ДКОИ или КОИ-7 (табл. 13.7). Например, правый нижний байт экрана 12×40 имеет десятичный адрес — 479, шестнадцатеричный — 01DF, в коде ДКОИ — X'C75F', в коде КОИ-7 — X'475E'.

Макрокоманды для работы с комплексом ЕС-7920

В табл. 13.8 приведены последовательности макрокоманд для реализации различных функций.

Таблица 13.7

Соответствие кодовых комбинаций бит 2—7 кодам в ЭВМ

Комбинация бит 2—7	Символьное представление	Код ДКОИ	Код КОИ-7	Комбинация бит 2—7	Символьное представление	Код ДКОИ	Код КОИ-7
00	Пробел	40	20	20	-	60	2D
01	A	C1	41	21	/	61	2F
02	B	C2	42	22	S	E2	53
03	C	C3	43	23	T	E3	54
04	D	C4	44	24	U	E4	55
05	E	C5	45	25	V	E5	56
06	F	C6	46	26	W	E6	57
07	G	C7	47	27	X	E7	58
08	H	C8	48	28	Y	E8	59
09	I	C9	49	29	Z	E9	5A
0A	[4A	5B	2A		6A	7C
0B	.	4B	2E	2B	,	6B	2C
0C	<	4C	3C	2C	%	6C	25
0D	(4D	28	2D	—	6D	5F
0E	+	4E	2B	2E	>	6E	3E
0F	!	4F	21	2F	?	6F	3F
10	&	50	26	30	0	F0	30
11	J	D1	4A	31	1	F1	31
12	K	D2	4B	32	2	F2	32
13	L	D3	4C	33	3	F3	33
14	M	D4	4D	34	4	F4	34
15	N	D5	4E	35	5	F5	35
16	O	D6	4F	36	6	F6	36
17	P	D7	50	37	7	F7	37
18	Q	D8	51	38	8	F8	38
19	R	D9	52	39	9	F9	39
1A]	5A	5D	3A	:	7A	3A
1B	^	5B	24	3B	#	7B	23
1C	*_	5C	2A	3C	@	7C	40
1D)	5D	29	3D	▴	7D	27
1E	;	5E	3B	3E	=	7E	3D
1F	┘	5F	5E	3F	.	7F	22

Таблица 13.8

Макрокоманды для комплексов ЕС-7920

Функция	Некоммутируемые каналы	Коммутируемые каналы	Локальное подключение
Чтение модифицированного после действия оператора	READ TI(TT)	READ TI(TC[W])	READ TI
Чтение модифицированного	WRITE TIV(TTV) READ TT	WRITE TIV(TTV) READ TT	READ TM

Функция	Некоммутируемые каналы	Коммутируемые каналы	Локальное подключение
Чтение модифицированного с позиции	WRITE TI(TT) WRITE TTV READ TT	WRITE TI(TT) WRITE TTV READ TT	READ TN
Чтение буфера	WRITE TIV(TTV) READ TT	WRITE TIV(TTV) READ TT	READ TB
Чтение буфера с позиции	WRITE TI(TT) WRITE TTV READ TT	WRITE TI(TT) WRITE TTV READ TT	READ TBP
Запись в буферную память	WRITE TIR или WRITE TI(TT) WRITE TT WRITE TR или WRITE TI(TT) WRITE TTR	WRITE TI(TC) WRITE TT WRITE TR(TRM)	WRITE TI
Стирание и запись в буферную память	WRITE TIR или WRITE TI(TT) WRITE TT WRITE TR или WRITE TI(TT) WRITE TTR	WRITE TI(TC) WRITE TT WRITE TR(TRM)	WRITE TS
Стирание незащищенных полей	WRITE TIR или WRITE TI(TT) WRITE TR	WRITE TI(TC) WRITE TT WRITE TRM	WRITE TUS
Копирование	WRITE TIR или WRITE TI(TT) WRITE TR	—	—

Общий телекоммуникационный метод доступа

Общий телекоммуникационный метод доступа (ОТМД) предоставляет программисту средства для построения программы управления сообщениями и написания прикладных программ. Кроме того, ОТМД дает различные сервисные средства, в том числе средства управления сетью, контрольную точку-рестарт, системный журнал и средства отладки. В ОС ЕС издания 4.1 обеспечиваются мультиплексоры передачи данных МПД-1А, МПД-2, МПД-3, абонентские пункты АП-2, АП-4, АП-61 и АП-63 (стартстопные и синхронные), АП-70, режим ЭВМ—ЭВМ, локальные дисплеи ЕС-7906. В ОС ЕС издания 6.1 дополнительно обеспечиваются локальные и удаленные дисплейные станции ЕС-7920.

14.1. Программа управления сообщениями

Программа управления сообщениями (ПУС) состоит из пяти секций:

- секция активизации и деактивизации;
- секция определения наборов данных;
- секция управления каналами связи;
- секция обработчиков сообщений;
- секция программ пользователя.

Секция активизации и деактивизации должна кодироваться первой. Остальные секции можно кодировать в любом порядке.

14.1.1. Активизация и деактивизация ПУС

INTRO — активизировать ПУС. Эта макрокоманда должна быть первой в секции активизации ПУС. Формат:

```
[метка] INTRO [PROGID=символы] [DISK={NO|YES}] [CPB={целое|0}]
           [CIB={целое|2}] [PRIMARY={имя|SYSCON}] [CONTROL=
           {символ|0}]
           {KEYLEN=целое|UNITSZ=целое}, LNUNITS=целое
           [,MSUNITS={целое|0}] [,MSMAX= {целое
           70 }] [,MSMIN={целое
           50 }
           [,DLQ={элемент|0}] [,USEREG={целое
           0 }] [,INVAL={целое
           0 }
```

$[CPINTVL = \{\text{целое} | 1800\}] [CPRCDS = \{\frac{\text{целое}}{2}\}]$
 $[STARTUP = \{C|CY|W|WY\}[I]] [CKREQS = \{\frac{\text{целое}}{0} \cdot \{[RESTART = \{\frac{\text{целое}}{0}\}]\}$
 $[PASSWRD = \{\text{символы} | 0\}] [CROSSRF = \{\text{целое} | 0\}] [TRACE = \{\text{целое} | 0\}]$
 $[TREXIT = \text{метка}] [DTRACE = \{\text{целое} | 0\}] [OLTEST = \{\text{целое} | 10\}]$
 $[COMWRTE = \{YES | NO\}] [TOPMSG = \{NO | YES\}] [LINETYP = \begin{Bmatrix} BISC \\ STSP \\ BOTH \end{Bmatrix}]$
 $[FEATURE = (\{\frac{\text{NODIAL}}{\text{DIAL}}\}, \{\frac{N0857X}{857X}\}, \{\frac{NOTIMER}{TIMER}\})]$

Если операнды KEYLEN, STARTUP, LNUNITS или CPB (при условии DISK=YES) опущены, то оператор ЭВМ в момент запуска ПУС получит сообщение

nn IED002A SPECIFY TCAM PARAMETERS

после чего он может определить недостающие параметры, введя ключевые слова (см. описание соответствующих операндов) или операндов с необходимыми параметрами.

PROGID = символы — идентификатор ПУС (длиною от 1 до 230 символов), который вставляется в тело программы ПУС;

DISK = {NO | YES} — YES указывает, что на НМД имеется набор данных для очередей сообщений;

CPB = {целое | 0} — задает количество блоков канальных программ (максимум 65535), необходимых для обмена данными между буферами и очередями сообщений на НМД. Ключевое слово — D;

CIB = {целое | 2} — задает максимальное количество блоков команд оператора ЭВМ (CIB). Максимальное значение — 255;

PRIMARY = {имя | SYSCON} — задает имя станции или прикладной программы, используемой как пульт старшего оператора ОТМД. SYSCON — имя пульта ЭВМ. Ключевое слово — P;

CONTROL = {символы | 0} — задает группу символов (от 1 до 8), идентифицирующую команды оператора ОТМД. Ключевое слово — L;

KEYLEN = целое — задает размер блока буфера (от 35 до 255);

UNITSZ = целое — то же, что и для KEYLEN. Ключевое слово — K;

LNUNITS = целое — задает количество блоков буфера (максимум 65535) для построения буферов для сегментов входных и выходных сообщений. Ключевое слово — B;

MSUNITS = {целое | 0} — задает максимальное число блоков буфера (до 65535), одновременно выделяемых набору данных очередей сообщений в основной памяти. Ключевое слово — M. Если используется набор данных очередей сообщений в основной памяти, значение этого параметра должно быть больше нуля;

MSMAX={целое|70} — задает процент заполнения набора данных очередей сообщений, при котором выдается предупреждающее сообщение. Ключевое слово — X. Максимальное значение — 100;

MSMIN={целое|50} — задает процент от указанного операндом MSUNITS количества используемых в наборе данных очередей сообщений блоков, ниже которого бит 8 в каждой записи об ошибке устанавливается в 1. Ключевое слово — Y. Максимальное значение — 99;

DLQ={элемент|0} — задает имя архивной очереди сообщений, куда посылаются сообщения с неверным кодом назначения. Ключевое слово — Q;

USEREG={целое|0} — количество сохраняемых регистров (максимально 10), начиная с регистра 2, если в обработчике сообщений встречаются команды программиста;

INTVAL={целое|0} — задает продолжительность в секундах (максимум 65535) системного интервала. Ключевое слово — I;

CPINTVL={целое|1800} — задает максимальное число секунд (минимум 30, максимум 65535) между контрольными точками. Ключевое слово — V;

CPRCDS={целое|2} — задает число сохраняемых контрольных точек. Ключевое слово — E. Максимальное значение — 75, минимальное — 2;

STARTUP={C|CY|W|WY}{I} — задает тип запуска ОТМД после прекращения работы или сбоя системы; C — после быстрого прекращения работы или прекращения работы с обслуживанием очередей должен выполняться рестарт I типа, а после системного сбоя — рестарт II типа со сканированием очередей сообщений; CY — после быстрого прекращения работы, прекращения работы с обслуживанием очередей или после системного сбоя должен выполняться рестарт I типа; W — после быстрого прекращения работы или прекращения работы с обслуживанием очередей должен выполняться рестарт III типа, а после системного сбоя — рестарт II типа со сканированием очередей; WY — после быстрого прекращения работы, прекращения работы с обслуживанием очередей должен выполняться рестарт III типа, а после системного сбоя — рестарт II типа без сканирования очередей сообщений; I — в запись контрольной точки включается состояние каждого списка приглашений. Ключевое слово — S;

CKREQS={целое|0} — задает максимальное количество очередей назначения, используемых одновременно прикладными программами и макрокомандой CKREQ. Ключевое слово — R;

RESTART={целое|0} — задает номер записи (максимум 255) контрольной точки, используемой для рестарта ПУС. Ключевое слово — N;

PASSWRD={символы|0} — задает пароль (от 1 до 8 символов) ПУС для прикладных программ. Ключевое слово — W;

CROSSRF={целое|0} — задает число элементов (максимум 65535) в таблице перекрестных ссылок. Ключевое слово — F;

TRACE={целое|0} — задает число элементов (максимум 65535) в таблице трассировки ввода-вывода ОТМД. Ключевое слово — T;

TREXIT=метка — указывает точку входа программы пользователя, которой передается управление после заполнения всех элементов таблицы трассировки ввода-вывода ОТМД. Адрес таблицы трассировки передается в регистре 0;

DTRACE={целое|0} — задает число элементов (максимум 65535) в таблице трассировки диспетчера ОТМД. Ключевое слово — A;

OLTEST={целое|18} — задает число блоков по 1024 байта (максимум 255) в основной памяти для работы программ оперативной проверки. Ключевое слово — O;

COMWRTE={YES|NO} — YES — указывает запуск задачи регистрации сервисных средств. Ключевое слово — G;

TOPMSG={NO|YES} — YES — задает вывод на пульт старшего оператора ОТМД сообщения IEA001 всякий раз, когда опрашиваемая станция не отвечает. Ключевое слово — H;

LINETYP={BISC|STSP|BOTH} — указывает тип каналов связи: BISC — все каналы связи синхронного типа; STSP — все каналы связи стартстопного типа; BOTH — используются оба типа каналов связи; для локальных дисплеев кодируется STSP или BOTH;

FEATURE=({NODIAL}, {NO857X}, {NOTIMER})
(DIAL, 857X, TIMER)

задает дополнительные возможности:

DIAL — используются коммутируемые каналы связи;

857X — используется ЕС-8570 со средствами прерывания от АП;

TIMER — используется хотя бы одна из следующих возможностей: контрольная точка; интервал; вызов по номеру; очереди в основной памяти; очереди на повторно используемом НМД.

После макрокоманды INTRO программист может включить команды, проверяющие в регистре 15 код возврата.

Коды возврата (десятичные):

4 — ОТМД уже имеется в системе;

8 — не хватает основной памяти для построения: пула буферов, пула свободных СРВ, таблицы трассировки подзадач, таблицы трассировки ввода-вывода, таблицы перекрестных ссылок;

12 — не хватает основной памяти для создания временной рабочей таблицы, используемой для сортировки таблицы имен АП;

16 — ошибка в определении АП (в операнде ALTDEST или в макрокомандах TERMINAL или TPROCESS);

20 — отсутствует пульт старшего оператора ОТМД.

OPEN — открыть наборы данных. Макрокоманда OPEN инициализирует наборы данных, используемые в ПУС. Сначала должны открываться наборы данных для очередей сообщений на НМД, затем — набор данных контрольной точки

[метка] OPEN (метка-dcb [, ({ {OUTPUT
INOUT
INPUT } [IDLE] }],...)
[, { MF=L
MF=(E, метка-списка) }]

метка — метка макрокоманды OPEN. Должна кодироваться, если MF=L;

метка-dcb — метка соответствующей макрокоманды DCB;

{OUTPUT|INOUT|INPUT} — указывает тип набора данных:

OUTPUT — выходной набор данных. Кодировается для системного журнала или для группы каналов связи, если ни одна из станций на этих каналах не может вводить данные;

INOUT — входной и одновременно выходной набор данных. Кодировается для набора данных очередей сообщений на НМД, контрольной точки и группы каналов связи;

INPUT — входной набор данных. Кодировается для группы каналов связи, на которых ни одна станция не может принимать данные.

IDLE — указывает, что во время инициализации набора данных группы каналов связи ни один из каналов связи данной группы не активизируется.

MF=L — указывает построение только списка параметров;

MF=(E, метка-списка) — указывает метку макрокоманды OPEN в списковой форме.

READY — завершить инициализацию ПУС. Эта макрокоманда кодируется между макрокомандами OPEN и CLOSE в секции активизации и деактивизации ПУС. Она задает адреса программ пользователя для построения приветственных сообщений и сообщений о рестарте. Выход на программу построения приветственного сообщения производится при начальном запуске ПУС и при каждом рестарте I типа, выход на программу построения сообщений о рестарте производится при рестарте II или III типа. Программе пользователя в регистре I передается адрес списка параметров из двух слов: первое содержит адрес элемента абонентской таблицы, второе — адрес резервных полей для станции назначения. Адрес построенного сообщения возвращается в регистре 15. Первым байтом сообщения является байт длины сообщения. Максимальная длина сообщения — 255 байт.

[метка] READY [GMMSG=программа] [,RMSG=программа]

GMMSG=программа — указывает метку программы построения приветственного сообщения;

RMSG = программа — указывает метку программы построения сообщений о рестарте.

CLOSE — **деактивизировать наборы данных**. Макрокоманда **CLOSE** должна кодироваться непосредственно за макрокомандой **READY**.

[метка] **CLOSE** (метка-dcb,...)

метка-dcb — указывает метку блока **DCB**. Первыми должны закрываться наборы данных групп каналов связи, потом — набор данных контрольной точки и в конце — набор данных очередей сообщений на **НМД**.

14.1.2. Определение наборов данных

Макрокоманда DCB для группы каналов связи. Формат:

метка **DCB** **DSORG**=TX, **MACRF**=(G, P) [,INTVL={целое|0}]
[,CPRI={R|E|S}] **DDNAME**=имя-dd [,EXLST=адрес-списка]
[,BUFIN={целое|1}] [,BUFOUT={целое|2}] [,BUFMAX={целое|2}]
[,BUFSIZE=целое], **INVLIST**=(имя-списка,,, имя-списка,,,,),
MH=метка-mh [,PCI= $\left(\left\{ \begin{smallmatrix} N \\ R \\ A \end{smallmatrix} \right\}, \left\{ \begin{smallmatrix} N \\ R \\ A \end{smallmatrix} \right\} \right)$] [,RESERVE=
= $\left(\left\{ \begin{smallmatrix} \text{целое-1} \\ 0 \end{smallmatrix} \right\} \left[, \left\{ \begin{smallmatrix} \text{целое-2} \\ 0 \end{smallmatrix} \right\} \right] \right)$]
,TRANS=таблица, SCT=таблица

DSORG=TX — идентифицирует набор данных группы каналов связи;

MACRF=(G, P) — задает тип макрокоманд, осуществляющих доступ к набору данных;

INTVL= {целое|0} — задержка в секундах между каждым прохождением списка приглашений. Максимальное значение — 255;

CPRI= {R|E|S} — задает относительный приоритет: R — более высокий приоритет приема, E — равные приоритеты приема и передачи, S — более высокий приоритет передачи;

DDNAME=имя-dd — указывает имя соответствующего оператора **DD**;

EXLST=адрес-списка — указывает адрес списка выходов на программы пользователя;

BUFIN= {целое|1} — задает число буферов для операций приема каждому каналу из группы. Максимальное значение — 15;

BUFOUT= {целое|2} — задает число буферов для операций отправки каждому каналу из группы. Максимальное значение — 15; минимальное — 2;

BUFMAX= {целое|2} — задает максимальное число буферов, одновременно используемых каждым каналом группы. Минимальное значение — 2, максимальное — 15;

BUFSIZE=целое — задает размер буферов в байтах для любого канала группы. Минимальное значение — 36, максимальное — 65535;

INVLST=(имя-списка ,, имя-списка,,...) — указывает имена (максимум 255 имен) списков приглашений для каналов группы; метка-mh — указывает адрес обработчика сообщений для группы каналов связи;

$$PCI = \left(\left\{ \begin{matrix} N \\ R \\ \underline{A} \end{matrix} \right\}, \left\{ \begin{matrix} N \\ R \\ \underline{A} \end{matrix} \right\} \right)$$

задает режим использования программно управляемого прерывания: N — программно управляемое прерывание не происходит ни во время приема, ни во время отправки; R — программно управляемое прерывание происходит во время заполнения или опорожнения каждого последующего буфера; буфер освобождается, но новый буфер на его место не назначается; A — то же, что и R, но на место освобожденного буфера назначается новый;

$$RESERVE = \left(\left\{ \begin{matrix} \text{целое-1} \\ \underline{0} \end{matrix} \right\} \left[, \left\{ \begin{matrix} \text{целое-2} \\ \underline{0} \end{matrix} \right\} \right] \right)$$

— «целое-1» задает число байт, зарезервированных в буфере, получающем первый входной сегмент каждого сообщения для информации, вставляемой макрокомандами DATETIME и SEQUENCE, «целое-2» задает число байт, зарезервированных во всех буферах, кроме первого. Максимальное значение — 255;

TRANS=таблица — указывает таблицу трансляции для группы каналов связи: DKOI — код передачи в ДКОИ; KOIL — перевод из ДКОИ в латинскую половину КОИ-7; KOIR — перевод из ДКОИ в русскую половину КОИ-7; MTKL — перевод из ДКОИ в латинскую часть МТК-2; MTKR — перевод из ДКОИ в русскую часть МТК-2;

SCT=таблица — указывает метку таблицы специальных символов: CHXG — синхронный режим передачи; 8563 — ЕС-8561 и ЕС-8563 (стартстопный режим); 8570 — ЕС-8570; MTKL — телеграфный аппарат с кодом передачи МТК-2; DKOI — ЕС-7906 и ЕС-7920 (локальный).

Макрокоманда DCB для очередей сообщений и контрольной точки. Формат:

метка DCB DSORG=TQ, MACRF=(G, P), DDNAME=имя-dd,

$$OPTCD=\{L|R|C\} [EXLST=\text{адрес-списка}] [THRESH=\left\{ \begin{matrix} \text{целое} \\ \underline{95} \end{matrix} \right\}]$$

DSORG=TQ — определяет набор данных для очередей сообщений или контрольной точки;

MACRF=(G, P) — определяет тип макрокоманд для доступа к набору данных;

DDNAME=имя-dd — указывает имя соответствующего оператора dd;

OPTCD={L|R|C} — L — указывает, что набор данных располагается на однократно используемом НМД; R — указывает, что набор данных располагается на повторно используемом НМД; C — указывает, что это набор данных для контрольной точки;

EXLST=адрес-списка — указывает адрес списка выходов для программ пользователя;

THRESH={целое|95} — задает процент использованных записей на однократно используемом НМД, после которого работа по обслуживанию очередей прекращается. По умолчанию принимается 95, максимальное значение 100. Для контрольной точки не кодируется.

Макрокоманда DCB для системного журнала. Формат:

метка DCB DSORG=PS, MACRF=W, DDNAME=имя-dd [,BLKSIZE=размер-ключа] [,RECFM=F] [,NCP=целое] [,SYNAD=адрес]

DSORG=PS — указывает последовательную организацию набора данных;

MACRF=W — указывает, что для записи используется макрокоманда WRITE;

DDNAME=имя-dd — указывает имя соответствующего оператора DD;

BLKSIZE=размер-ключа — задается та же величина, что и в операнде KEYLEN макрокоманды INTRO;

RECFM=F — указывает, что используется фиксированный формат записей;

NCP=целое — указывает максимальное число блоков в буфере;

SYNAD=адрес — указывает адрес программы обработки ошибок, заданной пользователем.

Выход на программу ABEND пользователя. Если макрокоманда OPEN не может открыть набор данных, а в списке выходов (EXLST) предусмотрен выход на программу ABEND пользователя (старший байт в списке выхода равен X'0E'), то программа пользователя получает управление, причем в младшем байте регистра 0 содержится код ошибки, а в младшем байте регистра 1 указываются возможные варианты действий программы ABEND:

Код ошибки

Значение

01	Нет памяти для построения DEB
02	Разный тип станций в группе каналов связи
03	Устройство не обеспечивается в ОТМД
04	Неправильный блок DCB или адрес в операторе DD
05	Неверный тип станции в блоке UCB или адрес в операторе DD
06	Неверные характеристики устройств или адрес в операторе DD
07	Нет памяти для построения LCB
08	Нет памяти для построения LCB для коммутируемого канала связи
09	Синхронный режим не соответствует блоку UCB
0A	Не закодировано DISK=YES в макрокоманде INTRO
0B	Размер ключа в DCB не соответствует указанному в макрокоманде INTRO
0C	Не все тома расположены на однотипных устройствах
0D	Неверное поле OPTCD в DCB
0E	Нет памяти для построения DEB
0F	Нет памяти для построения LCB
10	Неверно проинициализирован набор данных очередей сообщений

11)	Неверные адреса UCB
12	Неверно задан размер заголовка
13	Не определен набор данных очередей сообщений
14	При открытии каналов связи обнаружен неоткрытый набор данных очередей сообщений
15	Неверный относительный номер канала связи
16	Неверные характеристики канала связи
17	Не хватает основной памяти для ПУС
18	Не указан список приглашений

Код в регистре 1

Действия программы ABEND

03	а) выдать макрокоманду ABEND или поместить в младший байт регистра 15 код X'02', что приведет к аварийному завершению ПУС; б) поместить в младший байт регистра 15 код X'00'. Это передает управление следующей команде ПУС
07	а) то же, что и а) для кода 03 б) то же, что и б) для кода 03 в) поместить в младший байт регистра 15 код X'01'. В этом случае открывается группа каналов связи, кроме тех каналов, где встретились несовместимые станции

PCB — построить блок связи с прикладной программой.
Формат:

метка PCB MH=метка-mh, BUFSIZE=размер [,BUFIN={число|2}]
[BUFOUT={число|2}] [,RESERVE={целое-1, целое-2}]

MH=метка-mh — указывает адрес обработчика сообщений для прикладной программы;

BUFSIZE=размер — задает размер буферов для сообщений, минимальное значение — 36, максимальное — 65535;

BUFIN={число|2} — задает начальное число буферов, куда передаются данные из рабочей области PUT или WRITE; минимальное значение — 2, максимальное — 15;

BUFOUT={число|2} — задает начальное число буферов, которые заполняются при выполнении макрокоманд GET или READ; минимальное значение — 2, максимальное — 15;

RESERVE={целое-1, целое-2} — задает число резервируемых в буферах байт (см. макрокоманду DCB).

14.1.3. Управление каналами связи

TTABLE — начать абонентскую таблицу. Макрокоманда TTABLE должна быть первой в секции управления каналами связи. Формат:

[метка] TTABLE LAST=метка [,MAXLEN=целое]

LAST=метка — указывает последний элемент абонентской таблицы;

MAXLEN=целое — задает максимальную длину (от 1 до 8) имени элемента абонентской таблицы. По умолчанию принимается равной длине имени последнего элемента.

OPTION — зарезервировать поля. Макрокоманда OPTION описывает поля для станции, ее компонента, канала связи или при-

кладной программы. Одна или несколько макрокоманд **OPTION** выдаются перед макрокомандами **TERMINAL**, **TIIST** или **TPROCESS**. Является необязательной. Формат:

метка **OPTION** тип-размер

тип-размер — указывает тип и размер резервного поля в стандартном формате языка Ассемблера для обозначения полей без коэффициента кратности (например, **AL1**, **F**, **CL6**).

TERMINAL — создать элемент абонентской таблицы. Макрокоманды **TERMINAL** для станций на канале кодируются вместе, группы макрокоманд **TERMINAL** должны кодироваться в порядке возрастания относительного номера канала связи для группы каналов связи.

метка **TERMINAL** **QBY**={**T**|**L**}, **DCB**=метка-dcb, **RLN**=целое,
TERM=тип, **QUEUES**=вид [**DIALNO**= {цифры|**NONE**}]
[**ADDR**=символы][**LEVEL**= (целое, . . .)] [**CLOCK**=время]
[**CINTVL**=целое][**BUFSIZE**=целое][**ALTDEST**=элемент]
[**BFDELAY**=целое][**NTBLKSZ**= (размер-блока,
размер-подблока)]
[**TBLKSZ**=целое][**OPDATA**= (данные, . . .) [**SECTERM**=
= {**YES**|**NO**}]
[**(COMP**= {**YES**|**NO**})][**UTERM**= {**YES**|**NO**}]

метка — указывает имя станции;

QBY={**T**|**L**} — указывает тип очереди выходных сообщений:

T — очередь для станции; **L** — очередь для канала связи;

DCB=метка-dcb — указывает адрес соответствующего блока **DCB**;

RLN=целое — указывает относительный номер канала связи; максимальное значение — 255;

TERM=тип — указывает тип АП или режим работы:

7906 — алфавитно-цифровой дисплей **EC-7066**;

8561 — **EC-8561** — в стартстопном режиме;

8563 — **EC-8563** в стартстопном режиме;

857A — **EC-8570** без средств проверки на выделенном канале связи;

857D — **EC-8570** со средствами проверки на многопунктовом канале связи;

857E — **EC-8570** со средствами проверки и управления передачей на коммутируемом канале связи;

857F — **EC-8570** со средствами проверки на выделенном канале связи;

857G — **EC-8570** со средствами проверки на коммутируемом канале связи;

857X — **EC-8570** со средствами прерывания;

CHX1 — синхронный режим на некоммутируемом двухпунктовом канале связи;

CHX2 — синхронный режим на коммутируемом канале связи;

CHX3 — синхронный режим на некоммутируемом многопунктовом канале связи;

TA	— телеграфный аппарат с кодом передачи МТК-2;
792C	— устройство управления удаленного комплекса ЕС-7920. Используется в случае общего опроса всех индикаторов;
792R	— устройство отображения или печатающее устройство, входящее в состав удаленного группового комплекса;
792S	— устройство отображения, входящее в состав удаленного одиночного комплекса;
792L	— локальное устройство отображения или печатающее устройство;
QUEUES	= вид — указывает организацию очереди сообщений:
DR	— на повторно используемом НМД;
DN	— на однократно используемом НМД;
MO	— в основной памяти;
MR	— в основной памяти с перезаписью на повторно используемый НМД;
MN	— в основной памяти с перезаписью на однократно используемый НМД.

DIALNO = {цифры|NONE} — указывает тип канала связи (коммутируемый или некоммутируемый), «цифры» — телефонный номер станции; NONE означает отсутствие средств автовызова. Если операнд опущен, то станция считается некоммутируемой; опускается, если закодировано UTERM=YES;

ADDR = символы — указывает символы адресации в шестнадцатеричном представлении кода передачи; для стартстопных АП-61 и АП-63 за символами адресации должен следовать байт, указывающий тип команды записи: X'40' — запись с позиции курсора; X'50' — запись по адресу строки (адреса строк для стартстопных АП-61 и АП-63, приведенные в п. 13.8.6, заносятся в первый байт заголовка посылаемого сообщения); X'60' — запись со стиранием; для телеграфных аппаратов должен указываться символ КТМ (X'32'); для синхронных АП последним символом должен быть символ КТМ (в коде КОИ-7 — X'05');

LEVEL = (целое, ...) — определяет в возрастающем порядке допустимые приоритеты в заголовках сообщений для данной станции. Максимальное значение — 255;

CLOCK = время — указывает время суток, когда ЭВМ должна соединиться с коммутируемой станцией (две первые цифры — часы, вторые — минуты);

CINTVL = целое — задает временной интервал (в секундах), через который ЭВМ должна инициировать связь с коммутируемой станцией; максимальное значение — 65535;

BUFSIZE = целое — задает размер буфера в байтах для выходных сообщений для данной станции; подавляет значение, указанное в макрокоманде DCB для группы каналов связи, максимальное значение — 65535; минимальное — 37;

ALTDEST = элемент — указывает запасной пункт назначений для сообщений из очереди на повторно используемом диске;

BFDELAY = целое — задает временной интервал (в секундах) между посылками отдельных блоков сообщений; максимальное значение — 65535;

NTBLSZ = (размер блока, размер подблока) — задает размеры блоков (заканчивающихся символом КБ) и подблоков (заканчивающихся символом РИ1 в непрозрачном режиме); максимальное значение: для блока — 65535; для подблока — 255;

TBLSZ = целое — задает размер блока в прозрачном режиме; максимальное значение — 65535;

OPDATA = (данные, ...) — задает фактически начальные данные для резервных полей, определенных макрокомандой **OPTION** (см. макрокоманду **OPTION**). Если поле не используется для данной станции, то его отсутствие указывается запятой;

SECTERM = {YES|NO} — YES указывает, что данная станция может являться пультом ОТМД;

COMP = {YES|NO} — YES указывает, что данная макрокоманда описывает компонент станции, которая в свою очередь описывается другой макрокомандой;

UTERM = {YES|NO} — YES указывает, что данная макрокоманда описывает канал связи; NO — описывает АП или его компонент. YES указывается в тех случаях, когда на коммутируемом канале связи нельзя однозначно идентифицировать вызывающий АП.

TLIST — определить элемент каскадного списка или списка распределений. Макрокоманда **TLIST** определяет список станций или прикладных программ. Если сообщение содержит в качестве кода назначения имя каскадного списка, то сообщение помещается в очередь к тому элементу списка, у которого эта очередь наименьшая. Если кодом назначения является имя списка распределения, то сообщение поочередно посылается каждой станции или прикладной программе данного списка. Формат:

метка **TLIST TYPE** = {D|C}, **LIST** = (элемент, элемент, ...)

TYPE = {D|C} — D указывает список распределения, C — каскадный список;

LIST = (элемент, элемент, ...) — указывает элементы списка; максимальное число элементов, которое может быть задано несколькими макрокомандами **TLIST** — 32767.

TPROCESS — определить прикладную программу. Макрокоманда **TPROCESS** включает в абонентскую таблицу имя очереди для прикладной программы. Для каждой очереди назначения, к которой прикладная программа обращается макрокомандами **GET** или **READ**, и для каждого элемента, к которому обращается макрокоманда **PUT** или **WRITE**, нужно кодировать отдельные макрокоманды **TPROCESS**

метка **TPROCESS** **PCB** = метка-pcb[, **QUEUES** = вид]
[**ALTDEST** = элемент] [, **CKPTSYN** = {YES|NO}]
[, **SECTERM** = {YES|NO}] [, **RECDL** = ограничитель]
[, **LEVEL** = (целое, ...)] [, **OPDATA** = (данные, ...)]
[, **SECURE** = {YES|NO}]

метка — метка должна совпадать со значением параметра QNAME оператора DD для прикладной программы;

PCB=метка-pcb — указывает блок PCB. Все макрокоманды TPROCESS, выданные для одной и той же прикладной программы, должны указывать один и тот же блок PCB;

QUEUES=вид — указывает организацию очереди сообщений для данной прикладной программы (для GET или READ) (см. макрокоманды TERMINAL).

ALTDEST=элемент — указывает запасной пункт назначения (для GET или READ) для сообщений на повторно используемом НМД или пункт назначения (для PUT или WRITE), которому прикладная программа посылает ответы на команды оператора, выданные прикладной программой;

CKPTSYN={YES|NO} — YES означает, что во время рестарта очередь назначения не очищается;

SECTERM={YES|NO} — YES означает, что прикладная программа может рассматриваться как пульт оператора ОТМД;

RECDL=ограничитель — задает однобайтовый шестнадцатеричный код, используемый в качестве ограничителя записи; распознается макрокомандой GET в прикладной программе;

LEVEL=(целое, ...) — задает допустимые приоритеты в заголовках сообщений в порядке возрастания; максимальное значение — 255;

OPDATA=(данные, ...) — задает фактически начальные данные для резервных полей, определенных макрокомандой OPTION (см. макрокоманда OPTION).

SECURE={YES|NO} (издание 6.1) — YES указывает, что очередь защищается оператором ЭВМ, который получает сообщение всякий раз, когда прикладная программа делает попытку открыть очередь. В сообщении указывается имя задания, в рамках которого работает прикладная программа. Оператор может разрешить или запретить доступ к очереди, выдав соответствующий ответ.

LOGTYPE — инициировать средства записи в системный журнал. Макрокоманда LOGTYPE необходима, если в системный журнал записываются полные сообщения. Макрокоманда LOGTYPE не должна быть последней макрокомандой абонентской таблицы. Формат:

метка LOGTYPE метка-dcb, BUFSIZE=размер, QUEUES=вид

метка — должна совпадать с меткой макрокоманды LOG;
метка-dcb — указывает блок DCB для набора данных системного журнала;

BUFSIZE=размер — задает размер буферов для сообщений, записываемых в системный журнал, минимальное значение — 37, максимальное — 65535;

QUEUES=вид — указывает организацию очереди сообщений, ожидающих записи в системный журнал (см. макрокоманды TERMINAL).

INVLIST — создать список приглашений. Макрокоманда **INVLIST** кодируется после макрокоманд, определяющих абонентскую таблицу.

метка **INVLIST ORDER** = (элемент, ...) [**EOT** = код] [**CPUID** = адрес]
[**MASTER** = {**YES**|**NO**}]

метка — определяет список приглашений для канала связи и должна совпадать с меткой, заданной в операнде **INVLIST** макрокоманды **DCB** для группы каналов связи, включающей данный канал;

ORDER = (элемент, ...) — задает элементы списка приглашений. Элементы состоят из имени станции, индикатора (+ или —) и символов приглашения в шестнадцатеричном виде в коде передачи. Индикатор — (минус) означает, что данная станция опрашиваться не будет. Можно указать максимум 200 элементов. Станции опрашиваются в том порядке, в котором заданы элементы в списке операнда **ORDER**. Символы приглашения могут быть опущены для АП, не имеющих идентификационных последовательностей и работающих в режиме соперничества. Для дисплея **ЕС-7566** символом приглашения должен быть **X'02'** (чтение ручного ввода) или **X'06'** (чтение буфера), для старто-стопных АП-61 и АП-63 последним символом приглашения должен быть **X'20'**, для локальной станции **ЕС-7920** символы приглашения имеют вид **X'02'** (чтение буфера) или **X'06'** (чтение модифицированных полей);

EOT = код — шестнадцатеричный код символа **EOT** (в КОИ-7 **EOT** = 04), указывается только для синхронных АП;

CPUID = адрес — указывает адрес ячейки, содержащей идентификатор ЭВМ. Идентификатор начинается байтом, указывающим длину последовательности, за которым следует идентификационная последовательность в коде передачи;

MASTER = {**YES**|**NO**} (издание 6.1) — **YES** указывает, что в ситуации соперничества ОТМД будет пытаться захватить канал связи.

14.1.4. Обработчик сообщений

Обработчик сообщений состоит из двух групп макрокоманд: входной и выходной. Входная группа обрабатывает все сообщения, поступающие из каналов связи или от прикладной программы в ПУС. Выходная группа обрабатывает сообщения, посылаемые по каналам связи или прикладным программам. Каждая группа состоит из трех подгрупп: подгруппы заголовков, обрабатывающей заголовки сообщений; подгруппы буферов, обрабатывающей каждый сегмент сообщения; подгруппы сообщений, работающей после приема или отправки полного сообщения.

STARTMH — начало обработчика сообщений. Макрокоманда **STARTMH** должна быть первой макрокомандой каждого обработчика сообщений. Формат:

метка STARTMH LC = {IN|OUT} [{STOP} = { YES
CONT} = { (резервное-поле, переключатель) }]]

[, CONV = { YES
NO } { (резервное-поле, переключатель) }]]

[, LOGICAL = { (резервное-поле)
(резервное-поле1, переключатель, резервное-поле2) }]]

[, BREG = { целое
1 }]]

LC = {IN|OUT} — OUT указывает, что из входных сообщений удаляются управляющие символы HT и KB (стартстопные станции) и символы H3, HT, KB, KT (синхронные станции). Символ RI не рассматривается ПУС как символ управления передачей и не удаляется;

STOP = {YES| (резервное-поле, переключатель)} — YES безусловно прекращает передачу в случае ошибки; резервное поле — метка однобайтового резервного поля, определенного макрокомандой OPTION; переключатель — десятичное или шестнадцатеричное (в виде X'hh') число; передача прекращается в случае ошибки, если некоторые из бит переключателя включены в резервном поле;

CONT = {YES| (резервное-поле, переключатель)} — YES указывает, что в случае ошибки передача должна быть безусловно продолжена; передача продолжается в случае ошибки, если некоторые из бит переключателя включены в резервном поле; резервное поле и переключатель — см. выше описание операнда STOP;

CONV = {YES| (резервное-поле, переключатель) | NO} — YES указывает режим диалога, т. е. блок, введенный станцией, рассматривается как законченное сообщение; NO указывает, что ни режим диалога, ни обработка по приему KB не используются. Эти режимы используются, если некоторые из бит, заданных параметром «переключатель», включены в резервном поле; описание параметров «резервное поле» и «переключатель» см. выше в описании операнда STOP;

LOGICAL = { (резервное-поле)
(резервное-поле-1, переключатель, резервное-поле-2) }

— «резервное-поле» определяет 4-байтовое резервное поле, в котором старший байт указывает ошибку, а младшие три байта являются адресом подпрограммы, которой безусловно передается управление для каждого блока при обнаружении логических ошибок во входных сообщениях; параметр «резервное-поле-1» задает адрес подпрограммы пользователя, которой передается управление, если некоторые из бит, заданных переключателем, включены в однобайтовом «резервном-поле-2».

При входе в подпрограмму пользователя общие регистры содержат следующую информацию:

регистр 1 — адрес 4-байтового резервного поля, содержащего однобайтовый указатель ошибки и адрес подпрограммы пользователя;

регистр 4 — адрес LCB;
регистр 6 — адрес последнего буфера в блоке данных;
регистр 8 — адрес SCB;
регистр 13 — адрес области сохранения;
регистр 14 — адрес возврата;
регистр 15 — адрес точки входа в подпрограмму пользователя:

$BREG = \left\{ \begin{matrix} \text{целое} \\ 1 \end{matrix} \right\}$ — указывает число базовых регистров для

МН. Регистры назначаются с регистра 12 в сторону уменьшения; максимальное число — 11.

INHDR — начать входную подгруппу заголовка. Макрокоманда INHDR должна быть первой во входной группе. Формат:

[метка] INHDR [PATH=(резервное-поле, переключатель)]

PATH=(резервное-поле, переключатель) — «резервное-поле» указывает метку однобайтового поля, определенного макрокомандой OPTION; переключатель — десятичное или шестнадцатеричное (в виде X'hh') число. Подгруппа выполняется, если некоторые из бит переключателя включены в резервном поле. Если операнд опущен, то подгруппа выполняется всегда.

INBUF — начать входную подгруппу буферов. Макрокоманда INBUF указывает начало входной подгруппы буферов и является необязательной. Формат:

[метка] INBUF [PATH=(резервное-поле, переключатель)]

PATH=(резервное-поле, переключатель) — (см. макрокоманду INHDR).

INMSG — начать входную подгруппу сообщений. Макрокоманда INMSG указывает начало входной подгруппы сообщений. Является необязательной. Формат:

[метка] INMSG [PATH=(резервное-поле, переключатель)]

PATH=(резервное-поле, переключатель) — (см. макрокоманду INHDR).

INEND — закончить входную группу. Макрокоманда INEND указывает конец входной группы. Формат:

[метка] INEND

Операнды отсутствуют.

OUTHDR — начать выходную подгруппу заголовка. Макрокоманда OUTHDR указывает начало выходной подгруппы заголовка. Является необязательной. Формат:

[метка] OUTHDR [PATH=(резервное-поле, переключатель)]

PATH=(резервное-поле, переключатель) — (см. макрокоманду INHDR).

OUTBUF — начать выходную подгруппу буферов. Макрокоманда OUTBUF указывает начало выходной подгруппы буферов. Является необязательной. Формат:

[метка] OUTBUF [PATH=(резервное-поле, переключатель)]

PATH=(резервное-поле, переключатель) — (см. макрокоманду **INHDR**).

OUTMSG — начать выходную подгруппу сообщений. Макрокоманда **OUTMSG** указывает начало выходной подгруппы сообщений. Является необязательной. Формат:

[метка] **OUTMSG** [**PATH**=(резервное-поле, переключатель)]

PATH=(резервное-поле, переключатель) — (см. макрокоманду **INHDR**).

OUTEND — закончить выходную группу. Макрокоманда **OUTEND** указывает конец выходной группы. Должна быть последней макрокомандой в группе. Формат:

[метка] **OUTEND**

Операнды отсутствуют.

CANCELMG — аннулировать сообщение. Макрокоманда **CANCELMG** аннулирует сообщения безусловно или при возникновении определенной ошибки, является необязательной и если кодируется, то должна быть первой во входной подгруппе сообщений. Недопустима в других подгруппах. Формат:

[метка] **CANCELMG** [маска] [**CONNECT**={**AND**|**OR**}]

маска — указывает 5-байтовую маску, используемую для проверки записи об ошибке в каждом сообщении. Пропуск маски или указание маски с нулями приводит к безусловному выполнению макрокоманды; формат десятичный или шестнадцатеричный. В последнем случае используются символы **X** (при этом нужно кодировать старшие нули) или **XL5** (старшие нули могут быть опущены), максимальное значение — 16 777 215;

CONNECT={**AND**|**OR**} — **AND** указывает, что макрокоманда будет выполняться, если все биты, указанные в маске, включены в записи об ошибке; **OR** указывает, что макрокоманда будет выполняться, если хотя бы один из бит, указанных в маске, включен в записи об ошибке.

CHECKPT — дополнить запись контрольной точки. Макрокоманда **CHECKPT** производит дополняющую запись контрольной точки для резервных полей. Не может кодироваться в выходных подгруппах заголовков и буферов. Формат:

[метка] **CHECKPT**

Операнды отсутствуют.

CODE — транслировать данные. Макрокоманда **CODE** транслирует сегмент сообщения, проверяет команды оператора, кодируется в любом месте входных и выходных подгрупп заголовков и буферов (не допускается в других подгруппах). Формат:

[метка] **CODE** [{метка-таблицы|**NONE**|(регистр)}]

метка-таблицы — указывает таблицу трансляции, кодируется, как указано в описании операнда **TRANS** макрокоманды **DCB** (см. макрокоманду **DCB**);

NONE — сообщение не транслируется; данный параметр используется для проверки команд оператора;

регистр — указывает регистр (десятичное число от 2 до 11), предварительно загруженный адресом используемой таблицы.

COUNTER — подсчет числа сообщений или сегментов. Макрокоманда **COUNTER** кодируется во входных или выходных подгруппах заголовков или буферов. Формат:

[метка] **COUNTER** резервное-поле

резервное-поле — определяет двоичный счетчик с максимальным значением 65535, определяемый макрокомандой **OPTION**.

CUTOFF — определить максимальную длину входных сообщений. Макрокоманда **CUTOFF** задает максимальную длину входных сообщений и кодируется только во входной подгруппе буферов. Формат:

[метка] **CUTOFF** целое

целое — десятичное или шестнадцатеричное (в виде X'' или XLn'') число, задающее максимальное число символов во входном сообщении. Максимальное значение — 65535.

DATETIME — установить дату и время. Макрокоманда **DATETIME** вставляет дату и (или) время в заголовок входного сообщения; кодируется только во входной или выходной подгруппах заголовков. Формат:

[метка] **DATETIME** [DATA={YES|NO}][TIME={YES|NO}]

DATA={YES|NO} — NO указывает, что дату вставлять в заголовок не нужно;

TIME={YES|NO} — NO указывает, что время вставлять в заголовок не нужно.

ERRORMSG — послать сообщение об ошибке. Макрокоманда **ERRORMSG** посылает сообщение о возникшей ошибке; кодируется только во входной или выходной подгруппах сообщений.

[метка] **ERRORMSG** [маска][CONNECT={AND|OR}]
[.DEST={имя-назначения|резервное-поле|
|ORIGIN|DESTIN}]
[.DATA={сообщение|метка-поля}]
[.EXIT=адрес]

маска — десятичное или шестнадцатеричное (вида X'' или XL5'') число, указывающее, какого типа ошибки будут приводить к появлению сообщения об ошибке;

CONNECT={AND|OR} — AND указывает, что макрокоманда выполняется только в том случае, если все биты записи об ошибке, указанные в маске, установлены в единицу; OR указывает, что макрокоманда выполняется, если хотя бы один бит записи об ошибке, из указанных в маске, установлен в единицу. Макрокоманда всегда выполняется, если маска опущена или нулевая;

DEST={имя-назначения|резервное-поле|ORIGIN|DESTIN}

указывает пульт назначения сообщения об ошибке: имя-назначения — указывает имя элемента в абонентской таблице; окаймляется символами С" или CLn"; резервное-поле — указывает метку резервного поля, определенного макрокомандой OPTION, в котором находится имя элемента абонентской таблицы; ORIGIN — указывает, что сообщение об ошибке посылается станции-источнику ошибки; DESTIN — указывает, что сообщение об ошибке посылается станции, указанной в заголовке сбойного сообщения;

DATA = {сообщение|метка-поля} — сообщение — сообщение об ошибке; указывается с окаймляющими символами С" или CLn"; метка-поля — метка поля в основной памяти; в первом байт-поле — счетчик числа символов в сообщении, в последующих байтах — само сообщение; максимальная длина сообщения — 255 символов;

EXIT = адрес — указывает адрес программы пользователя, завершающей обработку сообщения об ошибке.

FORWARD — **направить сообщение в очередь.** Макрокоманда FORWARD направляет сообщения в очередь; кодируется в каждой входной подгруппе заголовков. Формат:

[метка] FORWARD [DEST = {имя-назначения|резервное-поле| (число) **|PUT|REG (число)}] [,EOA = символ] [,EXIT = адрес]

DEST = {имя-назначения|резервное-поле| (число) **|PUT|REG (число)} — указывает пункт назначения сообщения: имя-назначения — указывает имя элемента в абонентской таблице, окаймляется символами С" или CLn"; резервное-поле — указывает метку резервного поля, определенного макрокомандой OPTION, в котором содержится имя элемента абонентской таблицы; (число) — число символов (максимум 8) в каждом из имен назначений в заголовке сообщения; ** — указывает, что в заголовке сообщения имена назначений имеют разную длину и отделяются друг от друга пробелами (необходимо также кодировать операнд EOA); PUT — указывает для сообщений из прикладной программы, что назначения указываются пользователем в рабочей области прикладной программы; REG (число) — указывает регистр, содержащий адрес поля, в котором находится имя назначения;

EOA = символ — указывает символ или строку символов, используемую для ограничения поля назначения в заголовке, в символическом (С", CLn") или шестнадцатеричном представлении;

EXIT = адрес — указывает имя программы пользователя, которая получает управление в случае неверного назначения.

HOLD — **задержать передачу сообщений.** Макрокоманда HOLD приостанавливает передачу сообщений, кодируется во входных подгруппах заголовков и сообщений и в выходной подгруппе сообщений. Формат:

[метка] HOLD [маска] [,RELEASE] [,INTVL = число] [,CONNECT = {AND|OR}]

маска-десятичное или шестнадцатеричное (вида X" или XL5") число, указывающее, какого типа ошибки будут вызывать выполнение макрокоманды;

RELEASE — указывает, что передача приостанавливается до тех пор, пока не будут выданы команда оператора RESMXMIT или макрокоманда в прикладной программе MRELEASE;

INTVL = число — указывает число секунд, на которое приостанавливается передача, в десятичном или шестнадцатеричном (X" или XLn") формате;

CONNECT = {AND|OR} — AND указывает, что макрокоманда выполняется, если все биты, указанные в маске, в записи об ошибке установлены в единицу; OR — указывает, что макрокоманда выполняется, если хотя бы один бит, из указанных в маске, в записи об ошибке установлен в единицу. Макрокоманда всегда выполняется, если маска опущена или нулевая.

INITIATE — послать сегмент. Макрокоманда INITIATE инициирует послыку сегментов сообщений по назначениям, не дожидаясь поступления полного сообщения; кодируется во входной подгруппе заголовков. Формат:

[метка] INITIATE [ключ [,BLANK={символ|NO|YES}]]

ключ — указывает символ или строку символов (от 1 до 8) в символьном (C", CLn" или без окаймляющих символов) или шестнадцатеричном (X" или XLn") представлении, которые сравниваются со следующим непустым полем; макрокоманда INITIATE выполняется при сравнении или если операнд «ключ» опущен;

BLANK = {символ|NO|YES} — YES указывает, что пробелы в строке символов заголовка игнорируются; NO — указывает, что пробелы в строке символов заголовка не игнорируются: символ — указывает единичный символ в символьном (C" или CL1") или шестнадцатеричном (X" или XL1") представлении, который игнорируется в строке символов заголовка.

LOCK — установить блокировку. Макрокоманда LOCK поддерживает соединение между станцией и прикладной программой; кодируется во входной подгруппе заголовка. Формат:

[метка] LOCK [EXTEND
MESSAGE] [ключ [,BLANK={YES|NO|символ}]]

{EXTEND
MESSAGE} указывает тип режима блокировки: EXTEND указывает, что станция будет находиться в режиме блокировки до тех пор, пока она не будет иметь сообщений для передачи или пока не будет выполнена макрокоманда UNLOCK; MESSAGE указывает, что станция переводится в режим блокировки на время передачи сообщения и ответа на него;

ключ — указывает символ или строку символов (от 1 до 8) в символьном (C", CLn" или без окаймляющих символов) или шестнадцатеричном (X" или XLn") представлении, которые срав-

ниваются со следующим непустым полем; при сравнении или если операнд «ключ» опущен, макрокоманда LOCK выполняется;

BLANK={YES|NO|символ}—(см. макрокоманду INITIATE).

LOCOPT—получить адрес резервного поля. Макрокоманда LOCOPT получает адрес любого резервного поля; кодируется во входных и выходных подгруппах заголовков и буферов. Формат:

[метка] LOCOPT резервное-поле,{(регистр)|(15)}

резервное-поле — указывает метку резервного поля, определенную в макрокоманде OPTION (см. макрокоманду OPTION);

регистр — указывает номер регистра (со 2 по 11 или 15), куда помещается адрес резервного поля; если поле не найдено, то в данный регистр помещается X'FF', а в регистр 15 — X'04', а в случае указания регистра 15 он будет содержать нули.

LOG—послать сообщение в системный журнал. Макрокоманда LOG заносит в системный журнал сообщения или сегменты сообщений; кодируется в любой подгруппе. Формат:

[метка] LOG {адрес-dcb | адрес-типа}

адрес-dcb — указывает адрес блока управления данными для записи в журнал (используется в подгруппах заголовков и буферов);

адрес-типа — адрес элемента системного журнала в абонентской таблице (используется в подгруппах сообщений).

Если адрес DCB или адрес элемента задан неверно, то в регистр 15 помещается код X'04'.

MSGEDIT—отредактировать сообщение. Макрокоманда MSGEDIT вставляет, удаляет или замещает символы в сообщениях, кодируется во входных или выходных подгруппах буферов и заголовков. Формат:

[метка] MSGEDIT ((группа 1), (группа 2),...)[BLANK={символ|NO|YES}]

((группа1), (группа2),...) — каждая группа определяет одну операцию включения или удаления и состоит из четырех операндов: функций, данных, AT и TO, разделяемых запятыми. Можно указать таким образом до 31 группы. Операнды в группе имеют следующие форматы:

Операнд функций

{I|R[A][T]} — I задает функцию включения, включаемые данные указываются операндом данных; R задает функцию удаления, удаляются данные, указанные в операндах AT и TO; они замещаются данными, указываемыми операндом данных; если операнд AT или TO не указывает данных, то удаляется один байт данных, начиная с текущего положения указателя сканирования; A — задает, что удаление и замещение начинается с первого символа строки, указанной операндом AT; если A опущено, то удаление и замещение начинается с символа, следующего за последним символом в строке, указанной операндом AT; T — задает, что последним должен быть удален последний символ в строке, определенной операндом TO; если T опущено, то послед-

ним удаляется символ, предшествующий строке, указанной операндом TO,

Операнд данных

[символы|(шестнадцатеричный,п)|DELIMIT|CONTRACT]

— символы — от 1 до 8 букв, цифр или специальных символов (кроме пробела) в символьном (C" или CLn") или шестнадцатеричном представлении (X" или XLn"); символы включаются в сообщение или заменяют удаляемые данные; (шестнадцатеричный, п) — указывает, что данный шестнадцатеричный символ включается или заменяет удаляемый символ п раз; DELIMIT — указывает, что в случае операции удаления строка символов, заданная операндами AT и TO, замещается символом, указанным в операнде RECDL макрокоманды TPROCESS, метка которой содержится в поле назначения. Очередь назначения должна указываться макрокомандой FORWARD перед макрокомандой MSGEDIT, CONTRACT — указывает, что после удаления данных символы в буфере сдвигаются.

Операнд AT

[символы|смещение|(число, резервное-поле)|SCAN] — указывает место, с которого должно производиться включение или удаление данных; символы — от 1 до 8 букв, цифр или специальных символов (кроме пробела) в символьном (C" или CLn") или шестнадцатеричном представлении; смещение — целое десятичное число (максимум 65535), указывающее число байт за префиксом буфера, после которого включаются или удаляются данные; число — в десятичном или шестнадцатеричном (X" или XLn") представлении, указывающее число байт (максимум 65535), через которые в заголовок и в текст вставляются данные; резервное-поле — метка резервного поля, определенного операндом OPDATA макрокоманды TERMINAL или TPROCESS; SCAN — указывает, что включение или удаление должно начинаться с символа, следующего за байтом, на который в данный момент установлен указатель сканирования (используется в подгруппах заголовка).

Операнд TO

[символы|смещение|SCAN|(счетчик)|(0)] — указывает конец удаляемой строки символов; символы — от 1 до 8 байт в символьном (C" или CLn") или шестнадцатеричном (X" или XLn") представлении; смещение — десятичное целое число (максимум 65535), указывающее смещение от начала сегмента (показывает конец удаляемой строки); SCAN — указывает, что последним удаляемым символом является байт, на который установлен указатель сканирования; (счетчик) — указывает количество удаляемых байт; по умолчанию принимается значение 0;

BLANK={символ|NO|YES} — YES указывает, что символ пробела игнорируется; NO указывает, что символ пробела трактуется как данные; символ — символ в символическом (C", CLn" или без окаймляющих символов) или шестнадцатеричном (X" или XLn") представлении.

MSGFORM — сформировать блок сообщения. Макрокоманда MSGFORM вставляет в выходные сообщения символы управления передачей данных; кодируется только в выходной подгруппе заголовка. Формат:

[метка] MSGFORM [BLOCK=число] [SUBBLCK=число] [SENDTRP=
{ YES }
{ NO }]

BLOCK=число — указывает число байт (максимум 65535) в десятичном или шестнадцатеричном (X" или XLn") представлении в каждом выходном блоке; если этот операнд опущен, то используется значение операндов NTBLKSZ или TBLKSZ макрокоманды TERMINAL для станции назначения;

SUBBLCK=число — указывает величину подблока (максимум 255) в десятичном или шестнадцатеричном (X" или XL1") представлении; после каждой группы с указанным числом байт в сообщение вставляется управляющий символ PI1;

SENDTRP={YES|NO} — YES указывает, что сообщение посылается в прозрачном режиме, NO указывает, что сообщение посылается в непрозрачном режиме.

MSGGEN — сгенерировать сообщение. Макрокоманда MSGGEN генерирует сообщение, не устанавливаемое в очередь; кодируется во входной или выходной подгруппе сообщений. Формат:

[метка] MSGGEN [маска], {сообщение|метка-поля}
[CONNECT={AND|OR}] [CODE={метка-таблицы|NO}]

маска — указывает 5-байтовую комбинацию в десятичном или шестнадцатеричном (X" или XLn") представлении для проверки записи об ошибке;

сообщение — указывает сообщение в символьном (C" или CLn") или шестнадцатеричном (X" или XLn") представлении, которое посылается источнику или пункту назначения в зависимости от того, где закодирована макрокоманда — во входной или выходной подгруппе сообщений; максимальная длина сообщения — 24 байта; сообщение должно включать все символы управления передачей, включая КП для стартстопных станций;

метка-поля — указывает адрес поля, в котором первый байт содержит число оставшихся байтов в поле, которое занимает сообщение;

CONNECT={AND|OR} — AND указывает, что макрокоманда будет выполняться, если все биты, указанные в маске, в записи об ошибке установлены в единицу; OR указывает, что макрокоманда выполняется, если хотя бы один бит из указанных в маске, в записи об ошибке установлен в единицу;

CODE={метка-таблицы|NO} — NO указывает, что сообщение не транслируется, метка-таблицы — указывает адрес таблицы, используемой для трансляции сообщения; если операнд опущен, то используется таблица, указанная в макрокоманде DCB, если в макрокоманде DCB таблица не указана, то трансляция не производится.

MSGLIMIT — ограничить число сообщений. Макрокоманда MSGLIMIT ограничивает число сообщений в течение одной передачи для станции на некоммутируемом канале, кодируется во входной и выходной подгруппах заголовков. Формат:

[метка] MSGLIMIT {число|резервное-поле}

число — указывает число сообщений (максимум 255) в десятичном или шестнадцатеричном (X" или XLn") представлении;

резервное-поле — имя однобайтового резервного поля, определенного макрокомандой OPTION.

MSGTYPE — определить группу условного выполнения. Макрокоманда MSGTYPE управляет прохождением заголовка через обработчик сообщений, кодируется во входной или выходной подгруппах заголовков. Формат:

[метка] MSGTYPE [символы [,BLANK={YES|NO|символ}]]

символы — от 1 до 8 символов в символьном (C", CLn" или без окаймляющих символов) или шестнадцатеричном представлении; если эти символы совпадают с полем в заголовке сообщения, то будут выполняться только те команды в подгруппе, которые находятся между данной и следующей макрокомандой MSGTYPE; если символы не соответствуют полю заголовка, то управление передается следующей макрокоманде MSGTYPE в подгруппе или, если ее нет — следующей разграничивающей макрокоманде; если операнд опущен, то обрабатываются заголовки сообщений, не обработанные предыдущей макрокомандой MSGTYPE с непустым операндом «символы»;

BLANK={YES|NO|символ} — YES указывает, что символ пробела игнорируется; NO указывает, что символ пробела трактуется как данные; символ — символ в символическом (C", CLn" или без окаймляющих символов) или шестнадцатеричном (X" или XLn") представлении, игнорируемый макрокомандой MSGTYPE.

ORIGIN — проверить поле источника. Макрокоманда ORIGIN проверяет поле источника в заголовке сообщения и, если оно неверное, устанавливает бит в записи об ошибке; кодируется во входной подгруппе заголовков. Макрокоманда ORIGIN должна кодироваться после макрокоманды CODE (если используется код передачи, отличный от ДКОИ). Формат:

[метка] ORIGIN [число|X'FF']

число — указывает число символов в десятичном или шестнадцатеричном (X" или XLn") представлении в поле источника (внутренние пробелы игнорируются), максимальное значение — 8;

X'FF' — указывает, что поле источника имеет переменную длину и считается законченным при появлении очередного пробела.

PATH — изменить путь сообщения. Макрокоманда PATH изменяет байт переключения пути, кодируется в подгруппах заголовков и буферов. Формат:

[метка] PATH переключатель, резервное-поле [символы [BLANK = { $\begin{matrix} \text{YES} \\ \text{NO} \\ \text{символ} \end{matrix}$ }]]

переключатель — задает переключатель, записываемый в байт резервного поля; указывает число от 0 до 255 в десятичном или шестнадцатеричном (X" или XL1") представлении;

резервное-поле — указывает адрес резервного поля, определенного макрокомандой OPTION;

символы — от 1 до 8 символов в символьном (C", CLn" или без окаймляющих символов) или шестнадцатеричном представлении, которые сравниваются с полем заголовка; если сравнение успешное или если операнд опущен, то макрокоманда PATH выполняется;

BLANK = {YES|NO|символ} — YES указывает, что символ пробела игнорируется; NO указывает, что символ пробела трактуется как данные; символ — символ в символическом (C", CL1" или без окаймляющих символов) или шестнадцатеричном (X" или XLn") представлении, игнорируемый макрокомандой PATH.

PRIORITY — задать приоритет. Макрокоманда PRIORITY указывает приоритет обработки сообщений и кодируется во входной подгруппе заголовков. Формат:

[метка] PRIORITY [число] [символы [BLANK = {YES|NO|символ}]]

число — десятичное число, указывающее приоритет; если данный операнд опущен, то предполагается, что приоритет содержится в следующем байте, не являющемся пробелом и расположенном за текущим положением указателя сканирования;

символы — от 1 до 8 символов в символьном (C", CLn" или без окаймляющих символов) или шестнадцатеричном (X" или XLn") представлении, которые сравниваются с полем заголовка; если сравнение успешное или если операнд опущен, то макрокоманда PRIORITY выполняется;

BLANK = {YES|NO|символ} — YES указывает, что символ пробела игнорируется, NO указывает, что символ пробела трактуется как данные; символ — символ в символическом (C", CL1" или без окаймляющих символов) или шестнадцатеричном (X" или XLn") представлении, игнорируемый макрокомандой PRIORITY.

REDIRECT — поставить в дополнительную очередь. Макрокоманда REDIRECT ставит в очередь сообщения, в которых обнаружены ошибки, указанные операндом «маска»; кодируется в подгруппах сообщений. Формат:

[метка] REDIRECT [маска] [CONNECT = {AND|OR}]
[DEST = {имя-назначения|резервное-поле|ORIGIN}]

маска — указывает 5-байтовую комбинацию в десятичном или шестнадцатеричном (X" или XL5") представлении для проверки записи об ошибке;

CONNECT={AND|OR} — AND указывает, что макрокоманда выполняется, если все биты, указанные в маске, в записи об ошибке установлены в единицу; OR указывает, что макрокоманда выполняется, если хотя бы один из бит, указанных в маске, установлен в единицу в записи об ошибке;

DEST={имя-назначения|резервное-поле|ORIGIN} — указывает дополнительное назначение:

имя-назначения — имя элемента в абонентской таблице в символьном (C" или CLn") или шестнадцатеричном (X" или XLn") представлении;

резервное-поле — метка резервного поля, определенного макрокомандой OPTION;

ORIGIN — указывает, что сообщение возвращается источнику.

SCREEN — модифицировать вывод на экран. Макрокоманда SCREEN указывает тип операции записи на экран; кодируется в выходной подгруппе заголовка. Формат:

[метка] SCREEN [WRE|WLA|WDC|EAU][,символы][,BLANK=
={YES|NO|символ}][,RETRV={YES|NO}]

[WRE|WLA|WDC|EAU] — указывает тип операции записи для вывода на экран дисплея: WRE указывает ЗАПИСЬ СО СТИРАНИЕМ, WLA указывает ЗАПИСЬ ПО АДРЕСУ СТРОКИ (ЕС-7906, стартовые АП-61 — АП-64) (при этом первым символом в сообщении должен быть адрес строки), WDC указывает ЗАПИСЬ С ПОЗИЦИИ КУРСОРА (ЕС-7906) или ЗАПИСЬ С ТЕКУЩЕЙ ПОЗИЦИИ (ЕС-7920), EAU указывает СТЕРЕТЬ ВСЕ НЕЗАЩИЩЕННОЕ (ЕС-7920);

символы — от 1 до 8 символов в символьном (C" или CLn") или шестнадцатеричном (X" или XLn") представлении, которые сравниваются с полем заголовка; если сравнение успешное или если операнд опущен, то макрокоманда SCREEN выполняется;

BLANK={YES|NO|символ} — YES указывает, что символ пробела игнорируется; NO указывает, что символ пробела трактуется как данные; символ — символ в символическом (C", CL1" или без окаймливающих символов) или в шестнадцатеричном представлении (X" или XL1"), игнорируемый макрокомандой SCREEN;

RETRV={YES|NO} — YES указывает, что код команды, подготовленный в выходном потоке данных для удаленного ЕС-7920, используется для формирования соответствующей канальной программы для локального комплекса ЕС-7920. Выходной поток данных не изменяется и его необходимо отредактировать перед выводом на локальный дисплей.

Форматы считываемых и посылаемых записей для комплексов ЕС-7920 приведены в 13.8.9.

Макрокоманда SCREEN, если не указать тип операции записи, возвращает в регистре 15 код, указывающий тип операции записи по умолчанию:

Операция	Код ДКОИ	Код КОИ-7
WDC	X'A0'	X'40'
WLA	X'B0'	X'50'
WRE	X'E0'	X'60'

SEQUENCE — проверить или вставить номер сообщения. Макрокоманда SEQUENCE проверяет входной порядковый номер входного сообщения или вставляет выходной порядковый номер в выходное сообщение; кодируется в подгруппах заголовков.

[метка] SEQUENCE

Операнды отсутствуют. Для включения номера сообщения необходимо с помощью операнда RESERVE макрокоманды DCB зарезервировать 5 байт в заголовке сообщения.

Коды возврата (десятичные) в регистре 15:

- 4 — не хватает зарезервированной памяти для включения номера в выходное сообщение;
- входной порядковый номер меньше ожидаемого;
- 8 — неверный или слишком большой входной порядковый номер;
- 12 — неизвестен источник входных сообщений.

Рестарты ОТМД типов II и III сохраняют непрерывность порядковых номеров.

SETEOF — указать конец файла. Макрокоманда SETEOF устанавливает в префиксе буфера бит, указывающий конец файла; кодируется в выходной подгруппе заголовка. Формат:

[метка] SETEOF [символы [BLANK={YES|NO|символ}]]

символы — от 1 до 8 символов в символьном (C" или CLn") или шестнадцатеричном (X" или XLn") представлении, которые сравниваются с полем заголовка; если сравнение успешное или если операнд опущен, то макрокоманда SETEOF выполняется;

BLANK={YES|NO|символ} — YES указывает, что символ пробела игнорируется, NO указывает, что символ пробела трактуется как данные; «символ» — символ в символическом (C", CL1" или без окаймляющих символов) или в шестнадцатеричном (X" или XL1") представлении, игнорируемый макрокомандой SETEOF.

SETSCAN — установить указатель сканирования. Макрокоманда SETSCAN перемещает указатель сканирования, определяет адрес указателя сканирования; кодируется в подгруппах заголовков. Формат:

[метка] SETSCAN { $\begin{matrix} \text{символы} \\ \text{целое} \end{matrix} \} \text{ [BLANK={YES|NO|символ}] }$
 $\text{[POINT={BACK|FORWARD}]} \text{ [MOVE={RETURN|KEEP}] }$
 $\text{[RESULT={{(регистр)|(15)}}]}$

символы — указывает строку от 1 до 8 символов в символьном (C", CLn" или без окаймляющих символов) или шестнадцатеричном (X" или XLn") представлении, на которую необходимо установить указатель сканирования (если MOVE=KEEP), или адрес последнего символа, который нужно поместить в регистр, указываемый операндом RESULT (если MOVE=RETURN)); в случае MOVE=RETURN указатель сканирования должен находиться в текущем буфере;

целое — десятичное число пропускаемых символов (максимум 65535); если 0, то в регистре возвращается адрес указателя сканирования;

BLANK={YES|NO|символ} — YES указывает, что символ пробела при пропуске игнорируется; NO указывает, что символ пробела трактуется как данные; символ — символ в символьном (C", CL1" или без окаймляющих символов) или шестнадцатеричном (X", XLn") представлении, игнорируемый при перемещении указателя сканирования;

POINT={BACK|FORWARD} — FORWARD указывает, что указатель сканирования перемещается вперед, BACK — назад; если указано BACK, то указатель сканирования не может быть передвинут за пределы буфера;

MOVE={RETURN|KEEP} — KEEP указывает, что указатель сканирования перемещается к новой позиции; RETURN указывает, что указатель сканирования не перемещается, вместо этого в регистре возвращается адрес последнего символа в заданной строке;

RESULT={ (регистр) | (15) } — указывает общий регистр (от 2 до 11 или 15), в который помещается адрес последнего символа отыскиваемой строки.

Коды возврата (десятичные) в регистре 15:

00 — не найдена строка (MOVE=RETURN, RESULT=(15));

04 — не найдена строка (MOVE=RETURN, RESULT≠(15));
указатель сканирования выходит за пределы буфера (POINT=BACK).

TERRSET — установить бит ошибки. Макрокоманда TERRSET устанавливает бит ошибки (бит 20) в записи об ошибке; кодируется в подгруппах заголовков и буферов. Формат:

[метка] TERRSET.

Операнды отсутствуют.

UNLOCK — снять блокировку. Макрокоманда UNLOCK выводит станцию из режима блокировки; кодируется в подгруппах заголовков и сообщений. Формат:

[метка] UNLOCK [символы [,BLANK={YES|NO|символ}]]

символы — от 1 до 8 символов в символьном (C", CLn" или без окаймляющих символов) или шестнадцатеричном (X" или XLn") представлении, которые сравниваются с полем заголовка;

если сравнение успешное или если операнд опущен, макрокоманда UNLOCK выполняется;

BLANK={YES|NO|символ} — YES указывает, что символ пробела игнорируется; NO указывает, что символ пробела трактуется как данные; символ — символ в символьном (C", CLп" или без окаймляющих символов) или шестнадцатеричном (X" или XL1") представлении, игнорируемый макрокомандой UNLOCK.

14.1.5. Программы пользователя в ПУС

Программы пользователя в ПУС могут быть открытыми и закрытыми. Открытая подпрограмма может включаться в подгруппы заголовков и буферов обработчика сообщений и использовать макрокоманды ОТМД. Регистр 12 является базовым для открытой программы, в регистре 13 содержится адрес области сохранения. Содержимое регистра 13 необходимо восстановить, если оно было изменено, перед возвратом из открытой подпрограммы.

Закрытая подпрограмма включается в ПУС как программная секция, доступ к которой осуществляется, как указано в макрокомандах INTRO, STARTMH, DCB, READY, ERRORMSG или FORWARD. В закрытой подпрограмме нельзя использовать макрокоманды ОТМД.

14.2. Прикладные программы ОТМД

Прикладные программы выполняются асинхронно с ПУС как отдельные задачи или подзадачи обычно в другом разделе или зоне. Для передачи сообщений используются макрокоманды GET — PUT или READ — WRITE последовательного метода доступа. Соответствующий оператор DD имеет операнд QNAME, указывающий очередь, с которой работает прикладная программа.

14.2.1. Макрокоманда DCB в прикладной программе

В прикладной программе определяются входной и выходной наборы данных. Входной набор данных содержит сообщения, посылаемые к прикладной программе из очереди назначения, создаваемой макрокомандой TPROCESS. Выходной набор данных составляют сообщения, посылаемые из прикладной программы. Для каждого из этих наборов данных требуется макрокоманда DCB.

метка DCB DSORG=PS,MACRF= $\left\{ \begin{array}{l} G\{M|L\}[T] \\ R\{P \\ P\{M|L\} \\ W \end{array} \right\}$,DDNAME=имя-dd

[,BLKSIZE=число][,BUFL=число][,LRECL=число]

[,RECFM= $\left\{ \begin{array}{l} F \\ V \\ U \end{array} \right\} [B]]$][,OPTCD=[W][U][C] [,EODAD=адрес]

[,SYNAD=адрес][,EXLST=адрес]

DSORG=PS — указывает последовательную организацию набора данных;

MACRF=код — указывает тип используемых макрокоманд:

G — (входной набор данных) — GET;

R — (входной набор данных) — READ;

P — (выходной набор данных) — PUT;

W — (выходной набор данных) — WRITE;

M — режим пересылки;

L — режим указания;

T — использование POINT с GET;

P — использование POINT с READ;

DDNAME=имя-dd — указывает имя соответствующего оператора DD;

BLKSIZE=число — указывает размер в байтах (максимум 32760) рабочей области прикладной программы (может быть опущен для выходного набора данных, если не используется режим указания);

BUFL=число — указывает размер в байтах (от 37 до 65535) буферов ПУС;

LRECL=число — указывает размер записи плюс размер резервных полей (максимум 32760); кодируется обязательно, если указано RECFM=F;

RECFM=код — указывает формат рабочих блоков;

F — фиксированный формат;

V — переменный формат; B — заблокированный формат, но число записей в блоке равно единице;

U — неопределенный формат;

OPTCD=[W][U][C] — W указывает, что в 8-байтовое поле источника в рабочей области ОТМД должен поместить имя источника или пробелы, если источник неизвестен; U указывает, что обрабатываемый блок является не записью, а сообщением или сегментом сообщения; если U опущен, то предполагается, что рабочий блок является записью; C указывает, что для описания рабочего блока используется поле позиции размером в 1 байт:

X'F1' Первый сегмент сообщения

X'40' Промежуточный сегмент сообщения

X'F2' Последний сегмент сообщения

X'F3' Полное сообщение

X'F5' Первая часть сообщения, конец записи

X'F4' Промежуточная часть сообщения, конец записи

X'F6' Последняя часть сообщения, конец записи

X'F7' Полное сообщение, конец записи

EODAD=адрес — (только для входного набора данных) указывает адрес подпрограммы, получающей управление при обнаружении указателя конца файла в заголовке сообщения;

SYNAD=адрес — указывает адрес подпрограммы, получающей управление, если рабочий блок больше рабочей области и не указано OPTCD=C (для входного набора данных) или если

поле позиции или имя в поле назначения указаны неверно (для выходного набора данных);

EXLST=адрес — указывает адрес списка выходов; элемент с байтом управления X'05' является элементом выхода DCB; элемент с байтом X'0F' указывает адрес подпрограммы, инициирующей установку контрольных точек ОС; регистр 1 при входе в подпрограмму выхода содержит адрес DCB.

14.2.2. Активизация и деактивизация сопряжения с прикладной программой

Макрокоманды OPEN и CLOSE активизируют и деактивируют блоки DCB. Формат:

[метка] { OPEN
CLOSE } (метка-dcb, ...) [MF={L (E, адрес-списка)}]

метка-DCB — указывает адрес активизируемого DCB;
адрес-списка — указывает адрес макрокоманды OPEN в списковой форме.

Макрокоманда MCPCLOSE инициирует прекращение работы системы телеобработки и является необязательной. Формат:

[метка] MCPCLOSE {QUICK|FLUSH} [PASSWRD=пароль]

{QUICK|FLUSH} — QUICK указывает быстрое завершение работы системы телеобработки после передачи любого текущего сообщения (сообщения в очередях не передаются); FLUSH означает постепенное завершение работы системы (передаются все сообщения, поставленные в очередь);

PASSWRD=пароль — указывает пароль (от 1 до 8 символов), который нужно указывать, если в макрокоманде INTRO закодирован операнд PASSWRD.

Коды возврата (десятичные):

- 0 — успешное выполнение макрокоманды;
- 12 — ОТМД не работает;
- 20 — неправильный или неопределенный пароль.

14.2.3. Формат резервных полей в рабочей области

4 или 8 байт	1 байт	8 байт	
SAM-префикс	Поле позиции	Поле источника или поле назначения	

Начало рабочей области

Начало рабочего блока

SAM-префикс — присутствует, если RECFM=V или VB. Если V, то размер префикса равен 4 байтам (первые 2 байта — это размер рабочего блока+4, вторые 2 байта — двоичные нули). Если VB и MACRF=R или W, то размер префикса равен 8 байтам (первые 2 байта — это размер рабочего блока+8, вторые 2 бай-

та — двоичные нули, третьи 2 байта — размер рабочего блока + 4 последние 2 байта — двоичные нули);

поле позиции — см. описание операнда OPTCD=C в макрокоманде DCB (см. 14.2.1).

14.2.4. Макрокоманды доступа к данным

Макрокоманда GET. Макрокоманда GET получает рабочие блоки из ПУС. Если она выдана после начала быстрого завершения, то управление передается по адресу в EODAD. Формат:

[метка] GET метка-dcb [,метка-области]

метка-dcb — указывает адрес блока DCB;

метка-области — адрес области, куда помещается рабочий блок; может быть опущен, если в DCB задан режим указания; тогда область получает ОТМД и адрес области помещается после первой макрокоманды GET в регистр 1. Код возврата 12 (десятичный), если GET следует за POINT, говорит о том, что сообщение не получено.

Макрокоманда PUT. Макрокоманда PUT передает рабочие блоки в ПУС. Формат:

[метка] PUT метка-dcb [,метка-области]

Операнды имеют тот же смысл, что и для GET (см. выше).

Код возврата X'04' говорит о том, что макрокоманда PUT не выполнялась ввиду быстрого завершения работы системы телеобработки.

Код возврата X'10' говорит о том, что переполнен набор данных очередей сообщений.

Макрокоманда READ. Макрокоманда READ читает рабочий блок из ПУС в область прикладной программы. Формат:

[метка] READ метка-decb,SF, метка-dcb, метка-области
[,длина|'S'][,MF={L|(E, метка-списка))}]

метка-decb — указывает метку создаваемого блока DECB;

метка-dcb — указывает адрес блока DCB;

метка-области — указывает адрес рабочей области;

[длина|'S'] указывает длину рабочего блока плюс длину резервных полей (максимум 32760); кодируется только для рабочих блоков неопределенного формата; 'S' означает, что длина берется из поля LRECL блока DCB;

метка-списка — указывает адрес макрокоманды READ в списковой форме.

Макрокоманда WRITE. Макрокоманда WRITE посылает рабочие блоки в ПУС. Формат:

[метка] WRITE метка-decb, SF, метка-dcb, метка-области [,длина|'S']

Операнды макрокоманды WRITE имеют тот же смысл, что и для макрокоманды READ (см. выше).

Коды завершения (для макрокоманд READ и WRITE). Коды завершения в поле ECB блока DCB следующие:

7F000000	Нормальное завершение (READ и WRITE)
70000000	В ПУС работает SETEOF. Рабочая область не содержит рабочего блока (READ)
5C000000	Переполнен набор данных очереди назначения сообщений (WRITE)
5E000000	Начато быстрое завершение ОТМД. Запрос отвергнут (WRITE)
58000000	Ошибка в последовательности рабочих блоков (WRITE)
54000000	Недоступно значение сообщения (WRITE)
52000000	Переполнена рабочая память (READ)
50000000	Сообщение не найдено (POINT-READ)
40000000	Данные в очереди предварительного чтения
02000000	Конец очереди (нет SETEOF и нет данных в ПУС)
01000000	Очередь предварительного чтения пуста, очередь назначения не пуста

Макрокоманда CHECK. Макрокоманда CHECK переводит программу в состояние ожидания, пока не завершится операция ввода-вывода. В случае ошибки управление передается подпрограмме, адрес которой указан операндом SYNAD макрокоманды DCB. Если операнд опущен, то в регистре 15 в случае ошибки возвращается код X'08'. Формат:

[метка] CHECK метка-dcb

метка dcb — указывает адрес соответствующего блока DECB.

Макрокоманда TPDAT. Макрокоманда TPDAT указывает, нужно ли удалять ограничители записи из данных, считываемых прикладной программой. Формат:

[метка] TPDAT DCB = { метка
(регистр) }, DELETE = {YES|NO}

DCB = {метка| (регистр)} — указывает блок DCB, описывающий очередь сообщений для прикладной программы;

DELETE = {YES|NO} — YES указывает, что необходимо удалять ограничители записи.

14.2.5. Выход на подпрограмму SYNAD

Программа SYNAD получает управление, если поле позиции содержит неправильную величину или если указано неверное имя назначения.

Содержимое регистров на входе в подпрограмму SYNAD имеет следующее значение:

Регистр	Бит	Значение
0	8—31	Адрес блока DECB (BSAM) или адрес индикаторов состояния (QSAM)
1	0	1, если ошибка в GET или READ
	1	1, если ошибка в PUT или WRITE
	4	1, если неверное имя назначения (PUT или WRITE)

2—13	8—31	Адрес блока DCB
	8—31	Содержимое то же, что и до выдачи макрокоманды
14	8—31	Адрес возврата
15	8—31	Адрес подпрограммы SYNAD

14.2.6. Макрокоманды управления системой телеобработки

Макрокоманда TСОРУ. Макрокоманда TСОРУ копирует содержимое элемента абонентской таблицы и связанных с ним резервных полей в рабочую область. Формат:

[метка] TСОРУ имя-станции, метка-области

имя-станции — указывает имя станции;

метка-области — указывает адрес рабочей области, куда копируется информация.

Коды возврата (десятичные):

0 — успешное выполнение;

8 — ОТМД не работает;

12 — не открыт блок DCB в прикладной программе;

32 — в макрокоманде TСОРУ указано неправильное имя станции.

Макрокоманда ICОРУ. Макрокоманда ICОРУ копирует содержимое списка приглашений в рабочую область. Формат:

[метка] ICОРУ имя-группы, относительный-номер, метка-области

имя-группы — указывает имя группы каналов связи;

относительный-номер — указывает относительный номер канала связи в группе (максимум 255);

метка-области — указывает адрес рабочей области, куда копируется информация.

Формат списка приглашений:

Управление списком	X'nn'	X'aa'	X'll'		A	P	X'mm'		P		X'nn'
	0	1	2	3(0-5	6	7)	4	5(0-6	7)	6	7

Байты (в скобках биты)

Активные элементы	символы опроса	индекс	X'FE'
	l—1	1 байт	1 байт

Неактивные элементы	символы опроса	индекс
------------------------	----------------	--------

X'nn' — общее число элементов в списке опроса;

X'aa' — число активных элементов в списке опроса;

X'll' — длина каждого элемента (включая индекс);

Биты 0—5 в байте 3 являются управляющими битами ОТМД;

Бит 6 (А), если установлен в 1, указывает, что данный список является активным.

Бит 7 (Р), если установлен в 1, указывает, что используется автоопрос.

Байты 4—7 для устройств без буферов содержат нули или адрес идентификатора ЭВМ. Для устройств с буферами байт 4 (X'mm') указывает число активных устройств в канале связи, байт 5 содержит X'01' (Р), если канал связи доступен для автоопроса; байт 6 не используется, а байт 7 содержит общее число абонентских устройств на канале связи.

Ограничитель X'FE' указывает конец списка активных элементов. Для синхронных устройств перед ограничителем X'FE' должны быть символы КП.

Коды возврата (десятичные):

- 0 — успешное выполнение;
- 4 — неправильный относительный номер канала связи;
- 8 — ОТМД не работает;
- 12 — не открыт блок DCB в прикладной программе;
- 32 — неверный адрес DCB.

Макрокоманда QCOPY. Макрокоманда QCOPY копирует содержимое блока QCB (блока управления очередью назначения) и относительных приоритетов QCB в рабочую область. Размер рабочей области должен быть не менее 68—28п байт, где п — число различных приоритетов, указанное в операнде LEVEL макрокоманды TERMINAL. Формат:

[метка] QCOPY имя-станции, метка-области

имя-станции — указывает имя элемента абонентской таблицы; метка-области — указывает адрес рабочей области, куда копируется информация.

Коды возврата (десятичные):

- 0 — успешное выполнение;
- 4 — неправильный тип элемента таблицы;
- 8 — ОТМД не работает;
- 12 — не открыт блок DCB в прикладной программе;
- 32 — неправильное имя станции;
- 36 — не определен список приглашений.

Макрокоманда TCHNG. Макрокоманда TCHNG помещает указанные данные в элементы таблицы и резервные поля. Формат:

[метка] TCHNG имя-станции, метка-области[,PASSWRD=символ]

имя-станции — указывает имя элемента абонентской таблицы; содержимое которого заменяется новыми данными;

метка-области — указывает адрес рабочей области, из которой берется новая информация;

PASSWRD=символ — указывает пароль

Коды возврата (десятичные):

- 0 — успешное выполнение;
- 8 — ОТМД не работает;

- 12 — не открыт блок DCB;
- 20 — неправильный пароль;
- 32 — неправильное имя станции.

Макрокоманда ICHNG. Макрокоманда ICHNG изменяет список приглашений. Формат:

[метка] ICHNG имя-группы, относительный-номер,
 { метка-области } [,PASSWRD=символ]
 { ACT
 DEACT }

имя-группы — указывает имя группы каналов связи;
 относительный-номер — указывает относительный номер канала связи в группе (максимум 255);

{ метка-области }
 { ACT
 DEACT }

указывает тип модификации: метка-области — указывает адрес рабочей области, содержащей новый список опроса, ACT активизирует все элементы списка, DEACT деактивизирует все элементы списка;

PASSWRD=символ — указывает пароль.

Коды возврата (десятичные):

- 0 — успешное выполнение;
- 1 — не открыт DCB для группы каналов связи;
- 4 — неправильное имя группы;
- 8 — неправильный относительный номер;
- 12 — ОТМД не работает;
- 20 — неправильный пароль;
- 32 — неправильное имя группы.

Макрокоманда MRELEASE. Макрокоманда MRELEASE активизирует передачу сообщений приостановленной станции. Формат:

[метка] MRELEASE имя-станции[,PASSWRD=символ]

имя-станции — указывает имя станции, которой необходимо передать приготовленные для нее сообщения;

PASSWRD=символ — указывает пароль.

Коды возврата (десятичные):

- 0 — успешное выполнение;
- 1 — не открыт блок DCB;
- 4 — станция уже принимает сообщения;
- 12 — ОТМД не работает;
- 20 — неправильный пароль;
- 24 — имя станции не является одиночным элементом в абонентской таблице или используются очереди только в основной памяти;
- 32 — неправильное имя станции.

14.2.7. Возврат сообщений из очереди

Возврат сообщений из очереди осуществляется с помощью комбинации макрокоманд POINT и GET или READ. Нельзя вернуть сообщение из очереди, находящейся только в основной памяти. Формат:

[метка] POINT метка-dcb, адрес

метка-dcb — указывает блок DCB;

адрес — указывает адрес поля, необходимого макрокоманде POINT. Это поле содержит три смежных поля:

8-байтовое поле, содержащее имя станции;

поле, содержащее I(X'C9') или O(X'D6'). I указывает входной порядковый номер, O — выходной порядковый номер;

2-байтовое поле, содержащее двоичный порядковый номер.

Коды возврата (десятичные):

0 — успешное выполнение;

4 — сообщения с указанным порядковым номером нет в очереди назначения;

8 — неправильный пункт назначения;

12 — очередь назначения находится только в основной памяти.

14.2.8. Координация контрольных точек макрокомандой СКРЕQ

Контрольные точки ПУС могут быть скоординированы с контрольными точками прикладной программы с помощью макрокоманды СКРЕQ. Формат:

[метка] СКРЕQ

Поле операндов пусто.

Коды возврата (десятичные):

0 — успешное выполнение;

4 — запись контрольной точки не произведена.

14.3. Оперативное управление

14.3.1. Общие сведения

Для контроля и управления сетью телеобработки оператор сети имеет в своем распоряжении набор команд, которые могут вводиться с консоли ЭВМ, удаленной станции или из прикладной программы.

Команда оператора имеет следующий формат:

Идентификатор	Операция	Операнды	Окончание
---------------	----------	----------	-----------

Каждое поле отделяется друг от друга по крайней мере одним пробелом.

Идентификатор — от одного до восьми символов, идентифицирующих данное сообщение как команду оператора. Задается операндом CONTROL макрокоманды INTRO. Это поле не используется, если команды вводятся с консоли ЭВМ.

Операция — один из следующих типов операций: VARY, MODIFY, HALT, DISPLAY, HOLD, RELEASE.

Операнды — один или несколько операндов, разделенных запятыми.

Окончание — поле окончания содержит символ конца сообщения. Для ввода команд с удаленной станции перед символом конца сообщения должен быть символ «новая строка».

Далее при описании команд оператора поле «окончание» описываться не будет.

14.3.2. Команды оператора

Команда DISPLAY. Команда DISPLAY запрашивает необходимую управляющую информацию. Формат:

Идентификатор { DISPLAY } TP, тип $\left[\begin{array}{l} \text{, имя-группы, относительный-номер} \\ \text{, адрес} \\ \text{, имя-станции [, имя-поля, {X|C|D}]} \end{array} \right]$

тип — указывает тип запрашиваемых данных:

ACT — запрашивается перечень всех активных элементов списка приглашений. Кодировются также «имя-группы», «относительный-номер» или «адрес»;

ADDR — запрашивает относительный номер канала связи. Кодировается также «имя-станции»;

INACT — запрашивает список неактивных элементов в списке приглашений. Кодировются также «имя-группы», «относительный номер» или «адрес»;

INTER — запрашивает имена всех приостановленных станций;

LINE — запрашивает поле состояния и записи ошибки для данного канала связи. Кодировются также «имя-группы», «относительный-номер» или «адрес»;

LIST — запрашивается состояние списка приглашений для канала связи. Кодировются также «имя-группы», «относительный-номер» или «адрес»;

OPTION — запрашивает резервное поле для станции. Кодировются также «имя-станции», «имя-поля» и одно из трех X, C или D. X задает шестнадцатеричный формат, C — символьный, D — десятичный;

FRITERM — запрашивает имя станции, являющейся в данный момент пультом старшего оператора ОТМД;

QUEUE — запрашивает поля блока управления очередью. Кодировается также «имя-станции»;

SECTERM—запрашивает имена пультов оператора ОТМД;
TERM—запрашивает состояние станции. Кодировается также «имя-станции»;
имя-группы — указывает имя группы каналов связи;
относительный-номер — указывает относительный номер канала связи в группе;
адрес — указывает трехзначный шестнадцатеричный адрес начала связи;
имя-станции — указывает имя станции;
имя-поля — указывает имя резервного поля, связанного с данной станцией.

Возможные ответы на тип LINE в поле LNSTAT:

BS Синхронный канал связи
CM Канал связи в режиме управления
CR Операция продолжается или сбрасывается
DL Коммутируемый канал связи
IM Получение сообщения в режиме INITIATE
LF Канал связи свободен
MS Генерация сообщения (старт)
NR Отрицательный ответ на опрос
OC Канал связи остановлен оператором
RC Произведен повторный вызов
RV Канал связи в режиме приема
SD Канал связи в режиме передачи
TB КП из буферизованного АП
TR Трассировка ввода-вывода

Возможные ответы на тип LINE в поле ERR:

ABR Неисправность синхронной станции
CDC Ошибка при подключении или отключении
CHR Ошибка канала ЭВМ
CUR Ошибка устройства управления
CUT Ошибка при сохранении
FMT Ошибка формата
FWD Ошибка пересылки
HDR Неполный заголовок
HDW Ошибка аппаратуры
INV Неверный идентификатор станции
ISB Неверные буферы
LER Ошибка в канале связи
LST Потеря данных
MAX Достигнут максимальный размер основной памяти
MIN Достигнут минимальный размер памяти
MNS Сообщение не послано или не получено
NOP Станция не работает
NTS Система разделения времени не предусмотрена
OLT Оперативная проверка устройств не предусмотрена
ORG Неверный источник
SEL Ошибка выборки
SQH Слишком большой порядковый номер
SQL Слишком маленький порядковый номер
TER Ошибка АП
TXT Ошибка при передаче текста
UNX Особый случай в устройстве
USE Ошибка пользователя

Возможные ответы на тип QUEUE в поле «состояние»:

SNDDUF Передача к АП с буфером
 NONEON Биты состояния не установлены
 BUFRD Очередь для станции с буфером
 DELAY Задержка в очереди
 RDPRI Чтение имеет приоритет
 TSSES Работает система с разделением времени
 TWELVE Задержка вызова

Команда HALT. Команда HALT инициирует завершение работы. Формат:

Идентификатор $\left\{ \begin{matrix} \text{HALT} \\ Z \end{matrix} \right\}$ TP, $\left\{ \begin{matrix} \text{QUICK} \\ \text{FLUSH} \end{matrix} \right\}$

QUICK — инициирует быстрое завершение;

FLUSH — инициирует завершение с обслуживанием очередей.

Команда HOLD. Команда HOLD приостанавливает передачу к указанной станции. Формат:

Идентификатор $\left\{ \begin{matrix} \text{HOLD} \\ H \end{matrix} \right\}$ TP=имя-станции

имя-станции — имя останавливаемой станции.

Команда MODIFY. Команда MODIFY модифицирует состояние канала, станции или полей. Формат:

Идентификатор $\left\{ \begin{matrix} \text{MODIFY} \\ F \end{matrix} \right\}$ $\left\{ \begin{matrix} [\text{имя-процедуры.}] \text{идентификатор} \\ \text{имя-задания} \\ \text{ключ-данные} \end{matrix} \right\}$

имя-процедуры — имя процедуры, с помощью которой запущен ОТМД;

идентификатор — идентификатор, задаваемый во время запуска ОТМД;

имя-задания — имя задания запуска ОТМД из системного входного потока;

ключ — ключевое слово, задающее тип команды MODIFY. «Ключ» и «данные» имеют следующие значения:

AUTOPOLL = $\left\{ \begin{matrix} \text{имя-группы, относительный-номер} \\ \text{адрес} \end{matrix} \right\}, \left\{ \begin{matrix} \text{ON} \\ \text{OFF} \end{matrix} \right\}$

AUTOPOLL переключает канал связи с автоопроса на программный опрос (OFF) и наоборот (ON).

DEBUG = $\left\{ \begin{matrix} L \\ D \end{matrix} \right\}, \left\{ \begin{matrix} \text{IEDQFE10} \\ \text{IEDQFE20} \\ \text{IEDQFE30} \end{matrix} \right\}$

DEBUG активизирует и деактивизирует сервисную программу ОТМД, записывающую таблицу трассировки диспетчера подзадач, прерываний ввода-вывода:

L — активизировать регистрацию;

D — прекратить регистрацию;

IEDQFE10 запускает или останавливает запись таблицы трассировки подзадач;

IEDQFE20 запускает или останавливает запись таблицы трассировки ввода-вывода;

IEFDQFE30 запускает или прекращает дамп буферов;

INTENSE = $\left\{ \begin{array}{l} \text{LINE, } \left\{ \begin{array}{l} \text{имя-группы, относительный-номер} \\ \text{адрес} \end{array} \right\} \\ \text{TERM, имя-станции} \end{array} \right\}, \text{обозначение} \left[\cdot, \left\{ \begin{array}{l} \text{счет-} \\ \text{чик} \end{array} \right\} \underline{15} \right] \right\}$

INTENSE создает записи о временных ошибках в канале связи. Поле «обозначение» кодируется одним из следующих:

BO	Ошибка в канале ЭВМ	IR	Требуется вмешательство
CR	Команда отвергнута	LD	Потеря данных
DC	Ошибка в данных	OR	Перепополнение
ED	Ошибка в устройстве	TO	Таймаут
IM	Интенсивный режим	UE	Особый случай в устройстве

счетчик — десятичное число от 1 до 15, задающее количество записей, определенных в поле «обозначение»

INTERVAL = $\left\{ \begin{array}{l} \text{SYSTEM[данные]} \\ \text{POLL, имя-станции, данные} \end{array} \right\}$

INTERVAL изменяет задержку опроса;

SYSTEM задает системный интервал, определенный в макрокоманде INTRO;

POLL задает задержку опроса для данной станции;

данные — задержка опроса (максимальное значение 65535 для SYSTEM и 255 для POLL);

OPERATOR = $\left\{ \begin{array}{l} \text{имя-станции} \\ \text{SYSCON} \end{array} \right\}$

OPERATOR назначает пульт старшего оператора ОТМД;

имя-станции — имя станции, которая становится пультом старшего оператора ОТМД;

SYSCON — пультом старшего оператора ОТМД назначается консоль ЭВМ.

OPT = имя-станции, имя-резервного-поля, данные

OPT включает в резервное поле данные. Данные могут okay-ляться в символы C" или X".

TRACE = $\left\{ \begin{array}{l} \text{имя-группы, относительный-номер} \\ \text{адрес} \end{array} \right\}, \left\{ \begin{array}{l} \text{ON} \\ \text{OFF} \end{array} \right\}$

TRACE активизирует (ON) или останавливает (OFF) трассировку прерываний ввода-вывода в канале связи.

Команда RELEASE. Команда RELEASE освобождает станции, остановленные ранее командой HOLD. Формат:

Идентификатор $\left\{ \begin{array}{l} \text{RELEASE} \\ \text{A} \end{array} \right\}$ TP = имя-станции

Команда VARY. Команда VARY активизирует или деактивирует станции или канал связи. Формат:

Идентификатор $\left\{ \begin{array}{l} \text{VARY} \\ \text{V} \end{array} \right\} \left\{ \begin{array}{l} \text{имя-станции} \\ \text{имя-группы, } \left\{ \begin{array}{l} \text{ALL} \\ \text{относительный-номер} \end{array} \right\} \\ \text{имя-группы} \\ \text{адрес} \end{array} \right\}, \left\{ \begin{array}{l} \text{ONTR} \\ \text{OFFTR} \end{array} \right\}, \left[\begin{array}{l} \text{B} \\ \text{E} \\ \text{C} \\ \text{I} \end{array} \right]$

ONTR — активизирует станцию или группу;

OFFTP — деактивирует станцию или группу;

В — указывает, что некоммутируемые станции активизируются или деактивируются как для послыки, так и для приема сообщений. Кодировается с операндом «имя-станции»;

Е — указывает, что активизируется или деактивируется элемент списка приглашений. Кодировается с операндом «имя-станции»;

С — указывает, что передача останавливается в конце сообщения. Кодировается с операндом OFFTP;

I — указывает, что передача останавливается немедленно.

14.4. Средства отладки

14.4.1. Программа распечатки трассировки (IEDQXB)

Программа распечатки COMEDIT выполняется как отдельное задание, которое выглядит следующим образом:

```
//имя      JOB      ...  
//         EXEC     PGM=IEDQXB[,PARM='параметры']  
//SYSPRINT DD      SYSOUT=A  
//SYSUT1   DD      операнды, описывающие входной набор  
                        данных
```

'параметры' — параметры, управляющие работой программы:

STCB — распечатывается таблица трассировки диспетчера подзадач;

IOTR — распечатывается таблица прерываний ввода-вывода;

BUFF — распечатываются буферы;

BLOCK=hhmmddd — определяет начальную точку (только для набора данных на магнитной ленте), с которой распечатываются записи: hh — часы; mm — минуты; ddd — день года;

LINECNT= $\begin{Bmatrix} \text{xx} \\ 60 \end{Bmatrix}$ — задает число строк (до 99) на странице.

14.4.2. Программа распечатки набора данных очередей сообщений (IEDQXC)

Программа IEDQXC распечатывает набор данных очередей сообщений по номерам записей или по очередям. В одном задании можно распечатать до пяти очередей. Задание на распечатку имеет следующий вид:

```
//имя      JOB      ...  
//         EXEC     PGM=IEDQXC[,PARM='Q=параметры']  
//DISKQnn  DD      DSN=имя, DISP=OLD, UNIT=тип,  
                        VOL=SER=метка  
//SYSPRINT DD      UNIT=00F
```

Каждый экстенст набора данных очередей сообщений определяется отдельным оператором DD с именами DISKQnn, где nn — номер экстенста.

параметры — параметры, управляющие распечаткой:

DMP — распечатываются все сообщения по порядковым номерам;

xxx,DMP — последовательно распечатываются все сообщения, xxx — трехзначное десятичное число, указывающее общее число очередей;

xxx,ALL — последовательно распечатываются все сообщения в соответствии с их порядком в очереди, xxx — общее число очередей;

XXX, $p_1p_1p_1, p_2p_2p_2, \dots, p_5p_5p_5$ — распечатываются все сообщения для номеров очередей от $p_1p_1p_1$ до $p_5p_5p_5$ (до пяти очередей);

PART — позволяет распечатать часть записей (издание 6.1).
Оператору ЭВМ выдается сообщение IED426D, на которое он может ответить:

aaaaaa,bbbbbb — указать в шестнадцатеричном формате нижнюю и верхнюю границы;

ALL — распечатать все остальные записи;

END — закончить распечатку.

14.5. Запись об ошибке

С каждым обрабатываемым сообщением ОТМД связывает пятибайтовую запись об ошибке, в которой отмечаются различные ненормальные ситуации. Эти биты могут проверяться некоторыми макрокомандами обработчика сообщений. Значение каждого бита следующее:

- 0 Ошибка в заголовке
- 1 Неверное поле источника
- 2 Резервируется
- 3 Неправильный или слишком большой номер сообщения
- 4 Номер сообщения меньше ожидаемого
- 5 Резервируется
- 6 Не хватает буферов
- 7 Ошибка при выполнении макрокоманды CUTOFF
Неверный ответ на выборку
- 8 Очередь в основной памяти содержит меньшее число сообщений, чем это указано операндом MSMIN в INTRO
- 9 Очередь в основной памяти содержит большее число сообщений, чем это указано операндом MSMAX в INTRO
- 10 Резервируется
- 11 Резервируется
- 12 Резервируется
- 13 От синхронной станции получен символ аннулирования
- 14 Неверный код назначения
- 15 Резервируется
- 16 Потеряно входное сообщение из-за нехватки основной памяти
- 17 Неверный идентификатор станции
- 18 Станция не работает
- 19 Резервируется
- 20 Ошибка пользователя
- 21 Неверный формат сообщения для синхронного АП

- 22 Ошибка оборудования. Превышено пороговое значение для входных сообщений в очереди в основной памяти
- 23 Особый случай
- 24 Ошибка во время приглашения или выборки
- 25 Ошибка во время передачи данных
- 26 Ошибка во время соединения или разъединения
- 27 Сбой в АП
- 28 Резервируется
- 29 Ошибка в устройстве управления
- 30 Ошибка в канале
- 31 Неопределенная ошибка
- 32 Команда отвергнута
- 33 Требуется вмешательство
- 34 Ошибка на шин-К
- 35 Ошибка оборудования
- 36 Ошибка в данных
- 37 Перегрузка в канале
- 38 Потеря данных
- 39 Таймаут

Программы обслуживания ОС ЕС

15.1. Автономные программы обслуживания

15.1.1. Общие сведения

Задания на выполнение автономных программ обслуживания содержат набор управляющих и функциональных операторов, имеющих формат

Имя	Операция	Операнды	Комментарии
-----	----------	----------	-------------

и расположенных в следующей последовательности:

```

[имя] JOB
[имя] MSG TODEV=xxxx, TOADDR=ууу
      ⋮
      ⋮ } Функциональные операторы
[имя] END
  
```

где оператор JOB — начало задания;

оператор MSG определяет периферийное устройство для вывода сообщений программы обслуживания: TODEV=xxxx — тип устройства, TOADDR=ууу — адрес устройства;

функциональные операторы предназначены для описания запрашиваемых функций программ обслуживания;

оператор END — конец задания.

Для автономных программ обслуживания операторы задания подготавливаются в формате 80-позиционной перфокарты, учитывая следующие правила:

для полей оператора отводятся позиции с 1-й по 71-ю; если поля оператора занимают более 71 позиции, то оператор можно продолжить на несколько перфокарт, указывая символ продолжения в 72-й позиции и начиная продолжение на следующей перфокарте с 16-й позиции;

если оператору присваивается имя, то оно указывается с 1-й позиции;

поле операции заполняется, начиная с любой, кроме 1-й, позиции, при условии, что если указано имя оператора, то между полем имени и полем операции должен быть хотя бы один пробел;

среди операторов не допускаются пустые перфокарты; оператор для продолжения на следующую перфокарту может быть прерван или в 71-й позиции, или после любой запятой; имя оператора и операция должны быть указаны на первой перфокарте.

После загрузки автономной программы обслуживания в основную память следует определить устройство ввода управляющих операторов программы, которые могут помещаться следом за самой программой (за последней перфокартой или за последней записью программы на магнитной ленте) или вводиться с отдельного устройства ввода. Для задания устройства ввода надо выполнить следующие действия:

нажать клавишу ВНИМАНИЕ (ВН) на основной консоли, в ответ будет напечатано сообщение:

DEFINE INPUT DEVICE;

ввести с основной консоли сообщение:

INPUT=xxx CUU

где xxxx— типовое имя устройства ввода;

С— адрес канала ввода-вывода;

UU— адрес устройства ввода.

Допустимыми типовыми именами являются 6012 и 5010.

15.1.2. Возможности и функциональные операторы автономных программ обслуживания

Программа IBCDASDI. Инициализация тома прямого доступа задается функциональными операторами: DADEF, VLD, VTOCD, IPLTXT, LASTCARD (операторы должны располагаться в указанной последовательности).

Оператор DADEF предназначен для определения тома прямого доступа, подлежащего инициализации. Формат:

```
[имя] DADEF  TODEV=xxxx
              TOADDR=yyy
              {IPL=YES}
              {VOLID=регистрационный }
              {VOLID=SCRATCH           }
              [FLAGTEST=NO]
              [PASSES=n]
              [BYPASS=YES]
```

где TODEV=xxxx — определяет типовое имя устройства прямого доступа;

TOADDR=yyy — определяет адрес устройства прямого доступа в вычислительной системе;

IPL=YES — указывает, что на том прямого доступа должна быть записана программа начальной загрузки;

VOLID=регистрационный — определяет регистрационный номер тома прямого доступа, который будет инициализироваться не впервые;

VOLID=SCRATCH — указывает, что регистрационный номер инициализируемого тома не проверяется;

FLAGTEST=NO — используется при инициализации с проверкой дорожек на дефектность и указывает, что проверка дорожек должна производиться независимо от того, были ли они ранее отмечены как дефектные;

PASSES=n — используется при инициализации с проверкой дорожек на дефектность и указывает, что дорожка должна проверяться n раз ($n=1-255$);

BYPASS=YES — используется при инициализации без проверки дорожек на дефектность и указывает, что проверка дорожек не должна выполняться.

Оператор **VLD** предназначен для определения метки тома и дополнительных меток. Формат:

```
[имя] VLD NEWVOLID=регистрационный
      { VOLPASS=1 }
      { VOLPASS=0 }
      [OWNERID=xxxxxxxxxx]
      [ADDLABEL=n]
```

где **NEWVOLID**=регистрационный — указывает присваиваемый регистрационный номер тома (содержит от 1 до 6 символов);

VOLPASS=1 — указывает, что бит разграничения доступа к тому должен устанавливаться в 1;

VOLPASS=0 — указывает, что бит разграничения доступа к тому должен устанавливаться в 0 (это же значение бита устанавливается, если операнд **VOLPASS** не указывается);

OWNERID=xxxxxxxxxx — используется для указания информации о владельце тома (от 1 до 10 символов);

ADDLABEL=n — используется для указания числа дополнительных меток ($n=1-7$).

Оператор **VT OCD** предназначен для определения расположения оглавления тома прямого доступа и длины оглавления тома. Формат:

```
[имя] VT OCD STRTADR=nnnnn
      EXTENT=nnnn
```

где **STRTADR=nnnnn** — определяет адрес дорожки, с которой должно начинаться оглавление тома (для указания адреса используется десятичное число);

EXTENT=nnnn — задает длину оглавления тома в дорожках (для указания количества дорожек используется десятичное число; максимальное количество меток наборов данных (DSCB) на одной дорожке для НМД с объемом пакетов дисков 7,25 Мбайт, 29 Мбайт и 100 Мбайт соответственно равно 16, 25 и 39).

Оператор **IPLTXT** предназначен для указания, что на том прямого доступа должна быть записана программа начальной загрузки, которая на перфокартах размещается за оператором. В этом операторе только в поле операции указывается **IPLTXT**, а остальные поля оператора не заполняются. Если программа на-

чальной загрузки входит в задание, то в операторе END, помещаемом за программой, поле операции должно начинаться во второй колонке перфокарты.

Оператор LASTCARD предназначен для указания, что за заданием или набором заданий для программы IBCDASDI следуют другие операторы. Он необходим только в этом случае и размещается за последним оператором END. В операторе LASTCARD заполняется только поле операции с указанием LASTCARD.

Проверка качества дорожки тома прямого доступа и замена дефектной дорожки на запасную задается функциональным оператором GETALT. Формат:

```
[имя] GETALT TODEV=xxxx
          TOADDR=yyy
          TRACK=cccchhh
          VOLID=регистрационный
          [FLAGTEST=NO]
          [PASSES=n]
          [BYPASS=YES]
```

где TODEV=xxxx — определяет типовое имя устройства прямого доступа;

TOADDR=yyy — определяет адрес периферийного устройства в вычислительной системе;

TRACK=cccchhh — определяет адрес дорожек, для которых запрашивается запасная (альтернативная) дорожка (cccc — номер цилиндра, hhhh — номер дорожки на цилиндре);

VOLID=регистрационный — определяет регистрационный номер тома, на котором требуется замена дефектной дорожки на запасную;

FLAGTEST=NO — указывает, что перед заменой проверка качества заменяемой дорожки должна производиться независимо от того, была ли она ранее отмечена как дефектная;

PASSES=n — указывает, что дорожка должна проверяться перед заменой n раз (n=1—255) (без указания этого операнда проверка выполняется один раз);

BYPASS=YES — указывает, что проверка дорожки не должна выполняться (без указания операнда запасная дорожка назначается только в том случае, если указанная дорожка дефектная).

Программа IBCDMPRS. Выполнение дампа (копирование) тома прямого доступа задается функциональными операторами DUMP и VDRL (применяется для частичного дампа).

Оператор DUMP предназначен для определения исходного тома прямого доступа и тома, на который выполняется дамп (копия). Формат:

```
[имя] DUMP FROMDEV=xxxx
          FROMADDR=yyy
          TODEV=xxxx
          TOADDR=yyy
          [VOLID=регистрационный]
```

где FROMDEV=xxxx — определяет типовое имя периферийного устройства, с носителя данных которого создается копия;

FROMADDR=ууу — определяет адрес того же периферийного устройства в вычислительной системе;

TODEV=xxxx — определяет типовое имя периферийного устройства, на носителе данных которого создается дамп;

TOADDR=ууу — определяет адрес того же устройства в вычислительной системе;

VOLID=регистрационный[,per] — определяет регистрационные номера томов, на которых создается дамп.

Оператор VDRL предназначен для указания границ частично дампа тома. Если оператор VDRL задается, то он должен следовать за оператором DUMP. Формат:

```
[имя] VDRL BEGIN=nnnnn  
[END=nnnnn]
```

где BEGIN=nnnnn — определяет адрес дорожки, с которой будет выполняться частичный дамп;

END=nnnnn — определяет адрес последней дорожки, включаемой в частичный дамп (при выполнении дампа одной дорожки значения операндов END и BEGIN совпадают).

Выполнение операции, обратной операции DUMP, т. е. восстановление данных на том прямого доступа в исходном виде, задается функциональным оператором RESTORE. Формат:

```
[имя] RESTORE FROMDEV=xxxx  
FROMADDR=ууу  
TODEV=xxxx  
TOADDR=ууу  
VOLID=регистрационный
```

где FROMDEV=xxxx — определяет типовое имя периферийного устройства, с носителя данных которого восстанавливаются данные;

FROMADDR=ууу — определяет адрес того же периферийного устройства в вычислительной системе;

TODEV=xxxx — определяет типовое имя периферийного устройства, на носитель данных которого восстанавливаются данные;

TOADDR=ууу — определяет адрес того же периферийного устройства в вычислительной системе;

VOLID=регистрационный — определяет регистрационный номер тома прямого доступа, на который восстанавливаются данные.

15.2. Программы обслуживания наборов данных

15.2.1. Общие сведения

Задания на выполнение программ обслуживания наборов данных содержат набор операторов языка управления заданиями и управляющих операторов программы, расположенных в следующей последовательности:

```

JOB ...
//[имя] EXEC PGM=имя программы
//SYSPRINT DD ...
//имяDD1 DD ...
. . .
//имяDDK DD ...
//SYSIN DD ...

```

Операторы DD для описания наборов данных

Оператор DD для набора данных с управляющими операторами программы обслуживания

/*

Операторы DD языка управления заданиями, необходимые для каждой конкретной программы, приводятся далее в сведениях о соответствующей программе.

Управляющие операторы программ имеют формат

Имя	Операция	Операнды	Комментарии
-----	----------	----------	-------------

и подготавливаются в формате 80-позиционной перфокарты по следующим правилам:

для полей оператора отводятся позиции с 1-й по 71-ю; если оператор занимает более 71 позиции, то его можно продолжить на несколько перфокарт, указывая символ продолжения в 72-й позиции и начиная продолжение на следующей перфокарте с 16-й позиции;

имя оператора и операция должны быть указаны на первой перфокарте;

операнд может быть прерван для продолжения после любой запятой, а комментарий — в любом месте;

комментарии могут быть на перфокарте, содержащей все операнды или только часть их (в последнем случае нельзя продолжать комментарии на следующую перфокарту без операндов).

В описаниях управляющих операторов программ и значений поля PARM оператора EXEC используются обозначения:

квадратные скобки [] — показывают, что заключенный в них элемент необязательный (при этом скобки не кодируются);

фигурные скобки {} — показывают, что нужно выбрать один из элементов, указанных в них (при этом скобки не кодируются);

подчеркивание — показывает, что по умолчанию принимается подчеркнутый элемент, если он не указывается в операторе;

многоточие показывает, что за последним указанным элементом могут следовать один или несколько аналогичных элементов;

точка, запятая, знак равенства, круглые скобки и апостроф — рассматриваются только в качестве разделителей.

15.2.2. Возможности и управляющие операторы программ обслуживания наборов данных

Программа IEBCOMPR. Выполняет сравнение наборов данных. Организация наборов данных последовательная, библиотечная.

Формат записей наборов данных: фиксированный, переменный и неопределенный.

Для выполнения программы используются операторы языка управления заданиями JOB, EXEC, DD с именами SYSPRINT, SYSIN, SYSUT1 (определяет один из входных сравниваемых наборов данных), SYSUT2 (определяет другой входной сравниваемый набор данных) и управляющие операторы программы: COMPARE, EXITS, LABELS.

Оператор COMPARE предназначен для задания типа организации набора данных и используется, если в задании предусмотрены операторы EXITS и LABELS или если сравниваются библиотечные наборы данных. Формат:

$$[\text{имя}] \text{ COMPARE TYPORG} = \begin{Bmatrix} \text{PS} \\ \text{PO} \end{Bmatrix}$$

где TYPORG=PS — указывает, что входные наборы данных последовательные;

TYPORG=PO — указывает, что входные наборы библиотечные.

Оператор EXITS предназначен для указания вызываемой программы, необходимой для дополнительной обработки. Формат:

$$[\text{имя}] \text{ EXITS } \begin{Bmatrix} \text{INHDR} = \text{cccccccc} \\ \text{INTLR} = \text{cccccccc} \\ \text{ERROR} = \text{cccccccc} \\ \text{PRECOMP} = \text{cccccccc} \end{Bmatrix}$$

где INHDR=cccccccc — указывает имя программы обработки входных начальных меток пользователя ЭВМ;

INTLR=cccccccc — указывает имя программы обработки входных конечных меток пользователя ЭВМ;

ERROR=cccccccc — указывает имя программы обработки ошибок, которой должно передаваться управление после каждого несравнения;

PRECOMP=cccccccc — указывает имя программы обработки логических записей одного или обоих входных наборов данных.

Оператор LABELS предназначен для указания условия обработки программой IEBCOMPR меток пользователя ЭВМ как данных. Формат:

$$[\text{имя}] \text{ LABELS DATA} = \begin{Bmatrix} \text{YES} \\ \text{NO} \\ \text{ALL} \\ \text{ONLY} \end{Bmatrix}$$

где YES — указывает, что все метки пользователя ЭВМ программой обработки меток пользователя ЭВМ обрабатываются как данные (обработка меток как данных завершается в соответствии с кодом возврата);

NO — указывает, что метки пользователя ЭВМ не обрабатываются как данные;

ALL — указывает, что метки пользователя ЭВМ обрабатываются как данные независимо от кода возврата. Код возврата

16 приводит к завершению обработки группы меток пользователя ЭВМ и прекращению выполнения пункта задания;

ONLY — указывает, что только начальные метки пользователя ЭВМ обрабатываются как данные независимо от кода возврата.

Коды возврата программы:

00 — успешное выполнение задания,

08 — возникло несравнение; обработка продолжается;

12 — выполнение пункта задания прекращается;

16 — вызываемая программа передала IEBCOMPR код возврата, равный 16; выполнение пункта задания прекращается.

Программа IEBCOPY. Выполняет копирование библиотечных наборов данных. Обеспечиваются следующие возможности:

создание копии наборов данных с одновременным увеличением размера внешней памяти для набора данных;

разгрузка набора данных и загрузка разгруженного с помощью программы набора данных (обеспечено в ОС ЕС с издания 6.0);

выборка разделов из набора данных при выполнении копирования, разгрузки и загрузки;

замена одинаково поименованных разделов в наборах данных (исключая функцию разгрузки);

переименование выбранных разделов;

исключение указанных разделов набора данных при копировании, загрузке или разгрузке;

сжатие библиотечного набора данных в распределенной для него памяти на томе прямого доступа (исключая разгруженные наборы данных);

указание количества неиспользованных блоков справочника и неиспользованных дорожек в копии набора данных.

Для выполнения программы используются операторы языка управления заданиями JOB, EXEC, DD с именами SYSPRINT, SYSIN; операторы DD для описания входных и выходных наборов данных; операторы DD с именами SYSUT3 и SYSUT4 (определяют вспомогательные наборы данных) и управляющие операторы программы: COPY, SELECT и EXCLUDE.

В поле PARM оператора EXEC можно задать в байтах размер буфера программы:

PARM='SIZE=nnnnnnnn[k]'

где nnnnnnnn — число, содержащее от одной до восьми цифр;

K — указывает, что число умножается на 1024.

Оператор COPY предназначен для указания (в сочетании с операторами DD для входного и выходного наборов данных) функции, которая будет задана этим оператором COPY: копирование, загрузка или разгрузка. Формат:

[имя] COPY OUTDD=имя DD

$$\left[\text{INDD} = \left\{ \begin{array}{l} \text{имя DD1} [\text{имя DD2}] \dots \\ \text{имя DD1} [\text{имя DD2}] [(\text{имя DD3,R})] \dots \\ ((\text{имя DD1,R}) [\text{имя DD2}] \dots) \end{array} \right\} \right]$$

[LIST=NO]

где OUTDD=имяDD — определяет имя оператора DD для выходного библиотечного набора данных;

INDD — определяет имена операторов DD для входных библиотечных наборов данных (операнд INDD может быть на отдельной перфокарте, в этом случае другие операнды на этой перфокарте не указываются);

имяDD1 — определяет имя оператора DD для входного набора данных. Если определено несколько имен операторов DD (имяDD2, имяDD3 и т. д.) для копирования или загрузки, то порядок их следования определяет обработку наборов данных в той же последовательности. Параметр R необязательный и указывает, что все разделы, копируемые или загружаемые из указанного входного набора данных, должны заменить одинаково поименованные разделы в выходном библиотечном наборе данных. Когда определен параметр R, параметр имя DD и соответствующий ему параметр R должны быть заключены в круглые скобки; если он определяется с первым параметром имяDD1 в поле операнда INDD, то все операнды, за исключением ключевого слова INDD=, заключаются во вторую пару круглых скобок.

LIST=NO — определяет, что имена обрабатываемых разделов не следует выводить на печать.

Оператор SELECT предназначен для указания разделов, которые должны быть выбраны для копирования, загрузки или разгрузки в выходной набор данных. Оператор также используется для указания переименования или замены определенных разделов в выходном наборе данных. Формат:

$$[\text{имя}] \text{ SELECT MEMBER} = \left\{ \begin{array}{l} \text{имя1} [\text{имя2}] \dots \\ [(\text{имя1} [\text{имя2}, \text{R}] \dots)] \\ [(\text{имя1} [(\text{имя2}, \text{новое-имя} [\text{R}])])] \\ ((\text{имя1}, \text{новое-имя} [\text{R}]) [\text{имя2}]) \\ ((\text{имя1}, \text{R})) \end{array} \right\}$$

где MEMBER — определяет список основных и дополнительных имен разделов, выбираемых из входного набора данных;

имя — указывает имя раздела, копируемого в выходной набор данных (в одном шаге копирования имена разделов не должны повторяться);

новое-имя — указывает новое имя для выбираемого раздела (если это имя уже есть в выходном наборе данных, то раздел копируется, только если определена замена);

R — указывает, что данный раздел должен заменить раздел с таким же именем в выходном библиотечном наборе данных (параметр R нельзя указывать при разгрузке).

Оператор EXCLUDE предназначен для указания разделов, которые должны быть исключены из операции копирования, загрузки или разгрузки. Формат:

$$[\text{имя}] \text{ EXCLUDE MEMBER} = [(\text{имя1} [\text{имя2} \dots])]]$$

где MEMBER — определяет список основных и дополнительных имен разделов во входном наборе данных, которые не должны

копироваться, загружаться или разгружаться в выходной набор данных (имя раздела, определенное в одном шаге копирования, не должно повторяться).

Коды возврата программы:

00 — успешное выполнение задания;

04,08 — имеются ошибки, которые можно устранить; выполнение задания продолжается;

12 — обнаружена ошибка; выполнение задания прекращается.

Программа IEBDG. Предназначена для создания набора тестовых данных для использования, например, при отладке программ. Для построения тестовых данных используются стандартные шаблоны, предусмотренные в программе, и шаблоны, обеспечиваемые программистом. Могут быть созданы наборы данных с последовательной, индексно-последовательной и библиотечной организацией.

Для выполнения программы используются операторы языка управления заданиями JOB, EXEC, DD с именами SYSIN, SYSPRINT, оператор DD для описания входного набора данных, оператор DD для описания выходного набора данных и управляющие операторы программы: DSD, FD, CREATE, REPEAT, END.

Оператор DSD предназначен для указания исходного (входного) набора данных и создаваемого (выходного) набора данных. Формат:

```
[имя] DSD OUTPUT=(имяDD)
      [INPUT=(имяDD,...)]
```

где OUTPUT=(имяDD) — указывает имя оператора DD, определяющего выходной набор данных;

INPUT=(имяDD,...) — указывает одно или несколько имен операторов DD, определяющих входные наборы данных (нельзя в операнде INPUT указывать имя оператора DD SYSIN).

Оператор FD предназначен для определения содержания поля записи и используется только в том случае, если в последующем операторе CREATE определено имя поля. Формат:

```
[имя] FD NAME=имя
      LENGTH=длина
      [STARTLOC=начальное-положение-байта]
      { FILL= { 'символ' }
        { X'цифры' } }
      [FORMAT=шаблон[, CHARACTER=символ]]
      { PICTURE=длина, { 'символы' }
        { P'число' }
        { V'число' } }
      [SIGN=знак]
      [ACTION=действие]
      [INDEX=число[, CYCLE=число] [, RANGE=число]]
      [INPUT=имяDD]
      [FROMLOC=число]
```

где NAME=имя — указывает имя поля, определяемого в операторе FD;

LENGTH=длина — указывает в байтах длину определяемого поля (для записей переменной длины нужно резервировать 4 байта для дескриптора записи);

STARTLOC=начальное-положение-байта—определяет начальное положение поля записи;

FILL='символ' — указывает символ ДКОИ, которым заполняется каждый байт определяемого поля;

FILL=X'цифры' — указывает две шестнадцатеричные цифры, которыми заполняется каждый байт определяемого поля;

FORMAT=шаблон — определяет вид стандартного шаблона формата поля: AN — алфавитно-цифровой, ZD — зонированный десятичный, PD — упакованный десятичный, CO — упорядоченная последовательность символов, BI — двоичный, AL — алфавитный, RA — случайная числовая последовательность;

CHARACTER=символ — используется только с операндом FORMAT и определяет первый символ поля записи;

$$\text{PICTURE} = \text{'длина'}, \left\{ \begin{array}{l} \text{'символы'} \\ \text{P'число'} \\ \text{B'число'} \end{array} \right\}$$

определяет шаблон, обеспечиваемый программистом ('длина' — указывает количество символов в шаблоне; 'символы' — определяет строку символов ДКОИ, которые помещаются в указанном поле и выравниваются по левой границе; P'число' — указывает десятичное число, которое преобразуется в упакованный десятичный формат; B'число' — указывает десятичное число, которое преобразуется в двоичное и помещается в заданное поле);

SIGN=знак — указывает математический знак (+ или —) и используется, когда определяется упакованное или двоичное поле;

ACTION=действие — указывает действие, которое будет выполнено при включении поля в выходную запись, и может принимать следующие значения: SL — сдвиг влево, SR — сдвиг вправо, TL — ограничение слева, TR — ограничение справа, RO — вращение, WV — волна, FX — фиксация, RP — колебание;

INDEX=число — указывает приращение значения поля для каждой последующей записи или группы записей;

CYCLE=число — указывает количество выходных записей в группах записей;

RANGE=число — определяет максимальное значение поля;

INPUT=имяDD — указывает имя оператора DD для выходного набора данных;

FROMLOC=число — указывает позицию поля во входной записи.

Оператор CREATE предназначен для определения содержимого выходных записей. Формат:

```

[имя] CREATE      [QUANTITY=число]
                   [ FILL= { 'символ'
                             X'цифры' } ]
                   [ INPUT={имяDD
                             SYSIN[(cccc)] } ]
                   [ PICTURE=длина,начало, { 'символы'
                                                P'число'
                                                B'число' } ]
                   [ NAME={ {имя
                             (имя1,имя2,...)
                             (имя1,COPY=число,имя2,имя3,...) } } ]
                   [EXIT=имя-программы]

```

где QUANTITY=число — определяет количество записей, создаваемых оператором CREATE;

FILL — см. в операторе FD;

INPUT — определяет входной набор данных, записи которого используются при построении выходных записей;

имяDD — указывает имя оператора DD для входного набора данных;

SYSIN[(cccc)] — указывает, что набор данных SYSIN содержит, кроме управляющих операторов программы, записи, которые используются при построении выходных записей;

PICTURE — указывает длину, первый байт и содержание шаблона, обеспечиваемого программистом (см. также оператор FD);

NAME — указывает имя поля записи, которое включается в используемую выходную запись;

(имя1,...) — указывает имя или список имен полей, которые включаются в выходную запись;

COPY=число — указывает, что имена всех полей во внутренних скобках трактуются как группа и включаются указанное число раз в каждую выходную запись, создаваемую оператором CREATE;

EXIT=имя-программы — определяет имя программы пользователя ЭВМ, которая получает управление от программы IEBDG до занесения каждой выходной записи в выходной набор данных.

Оператор REPEAT предназначен для указания числа повторных использований оператора или группы операторов CREATE при создании выходных записей (оператор REPEAT предшествует операторам CREATE). Формат:

```
[имя] REPEAT QUANTITY=число[,CREATE=число]
```

где QUNTITY — указывает число повторных использований группы операторов CREATE (число не более 65535);

CREATE=число — указывает количество последующих операторов CREATE, включенных в группу.

Оператор END предназначен для указания конца набора управляющих операторов программы. Формат:

```
[имя] END
```

Коды возврата программы:

00 — успешное завершение задания;

- 04 — вызываемая программа возвращает программе IEBDG код 16; выполнение задания прекращается по требованию программиста;
- 08 — ошибка при обработке набора управляющих операторов программы; построение данных невозможно;
- 12 — ошибка при обработке входного или выходного набора данных; выполнение задания прекращается;
- 16 — ошибка, при которой выполнение задания прекращается.

Программа IEEDIT. Формирует выходной набор данных, содержащий набор заданий, который может быть использован в качестве входного потока заданий.

Для выполнения программы используются операторы языка управления заданиями JOB, EXEC, DD с именами SYSPRINT, SYSIN и SYSUT1, SYSUT2 (определяют соответственно входной и выходной последовательные наборы данных) и управляющий оператор программы EDIT.

Оператор EDIT предназначен для указания пунктов задания из входного набора данных, включаемых в выходной набор данных. Формат:

```
[имя] EDIT      [START=имя-задания]
                  [ TYPE= { POSITION
                           INCLUDE
                           EXCLUDE } ]
                  [ STEPNAME= ( { имя
                                имя-имя } [, { имя
                                                имя-имя } ] , ... ) ]
                  [NOPRINT]
```

где START=имя-задания — указывает имя задания во входном наборе данных;

TYPE=POSITION — указывает, что выходные данные состоят из оператора JOB, пункта задания, указанного в ключевом слове STEPNAME, и всех остальных пунктов задания, которые следуют за этим пунктом;

TYPE=INCLUDE — указывает, что выходной набор данных должен содержать оператор JOB и все пункты этого задания, приведенные в операнде STEPNAME;

TYPE=EXCLUDE — указывает, что входной набор данных должен содержать оператор JOB и все пункты этого задания, за исключением пунктов, приведенных в ключевом слове STEPNAME;

STEPNAME=имя — указывает имя пункта задания, начиная с которого пункты задания пересылаются в выходной набор данных (если определено TYPE=POSITION);

$$\text{STEPNAME} = \left(\left\{ \begin{array}{c} \text{имя} \\ \text{имя-имя} \end{array} \right\} \left[, \left\{ \begin{array}{c} \text{имя} \\ \text{имя-имя} \end{array} \right\} \right] , \dots \right)$$

указывает имена пунктов задания, которые должны быть включены в операцию копирования, если указано TYPE=INCLUDE, или исключены, если указано TYPE=EXCLUDE;

NOPRINT — указывает, что не осуществляется распечатка выходного набора данных.

Коды возврата программы:

- 00 — успешное завершение программы;
- 04 — возникла ошибка, в результате которой выходной набор данных может быть не пригоден для использования, но обработка продолжается;
- 08 — ошибка при обработке входного, выходного или управляющего набора данных, которая вызывает прекращение выполнения пункта задания.

Программа IEBGENER. Обеспечивает следующие возможности:

создание копии последовательного набора данных или раздела библиотечного набора данных;

запись разделов во вновь сформированный или существующий библиотечный набор данных;

редактирование последовательного или библиотечного набора данных;

изменение размера блока набора данных и (или) изменение размера его записи;

формирование меток пользователя ЭВМ для выходных последовательных наборов данных.

Для выполнения программы используются операторы языка управления заданиями JOB, EXEC, DD с именами SYSPRINT, SYSIN и SYSUT1, SYSUT2 (определяют соответственно входной и выходной наборы данных) и управляющие операторы программы: GENERATE, EXITS, LABELS, MEMBER, RECORD.

Оператор GENERATE используется в тех случаях, когда выходной набор данных является библиотечным, должно выполняться редактирование, предусматриваются для выполнения вызываемые программы, указывается обработка меток (оператор всегда первый среди управляющих операторов программы). Формат:

```
[имя] GENERATE [MAXNAME=n]
                  [MAXFLDS=n]
                  [MAXGPS=n]
                  [MAXLITS=n]
```

где MAXNAME=n — указывает число, не меньшее, чем общее количество основных и дополнительных имен в последующих операторах MEMBER;

MAXFLDS=n — указывает число, не меньшее, чем общее количество параметров FIELD в последующих операторах RECORD;

MAXGPS=n — указывает число, не меньшее, чем общее количество параметров IDENT в последующих операторах RECORD;

MAXLITS=n — указывает число, не меньшее, чем общее количество символов, которые содержатся в литерях параметров FIELD последующих операторов RECORD.

Оператор EXITS используется, если предусматриваются выходы на специальные обрабатывающие программы. Формат:

```
[имя] EXITS [INHDR=имя-программы]
              [OUTHDR=имя-программы]
              [INTLR=имя-программы]
              [OUTTLR=имя-программы]
              [KEY=имя-программы]
              [DATA=имя-программы]
              [IOERROR=имя-программы]
              [TOTAL=(имя-программы,размер)]
```

где INHDR=имя-программы — указывает имя программы обработки начальных меток пользователя ЭВМ входного набора данных;

OUTHDR=имя-программы — указывает имя программы, которая строит начальные метки пользователя ЭВМ выходного набора данных;

INTLR=имя-программы — указывает имя программы обработки конечных меток пользователя ЭВМ входного набора данных;

OUTTLR=имя-программы — указывает имя программы обработки конечных меток пользователя ЭВМ выходного набора данных;

KEY=имя-программы — указывает имя программы формирования ключа выходной записи;

DATA=имя-программы — указывает имя программы модификации блока данных;

IOERROR=имя-программы — указывает имя программы обработки ошибки ввода-вывода;

TOTAL=(имя-программы, размер) — указывает имя программы создания итоговых данных и размер в байтах, необходимый для данных этой программы (при кодировании этого операнда необходимо указать параметр OPTCD=T в DCB оператора DD с именем SYSUT2).

Оператор LABELS предназначен для определения обработки меток пользователя ЭВМ. Формат:

```
[имя] LABELS
              DATA= { YES
                       NO
                       ALL
                       ONLY
                       INPUT }
```

где YES — все достоверные метки пользователя ЭВМ обрабатываются как данные;

NO — метки пользователя ЭВМ не обрабатываются как данные;

ALL — метки пользователя ЭВМ обрабатываются как данные независимо от кода возврата;

ONLY — только начальные метки пользователя ЭВМ обрабатываются как данные независимо от кода возврата;

INPUT — выходные данные пользователя ЭВМ вводятся как 80-байтовые записи в управляющем наборе данных (этим записям предшествует оператор RECORD LABELS=n).

Оператор MEMBER предназначен для указания основного и дополнительных имен раздела в выходном библиотечном наборе данных. Формат:

[имя] MEMBER NAME = (имя[, псевдоним]...)

где NAME = (имя[, псевдоним]) — определяет основные и дополнительные имена раздела.

Оператор RECORD предназначен для определения группы записей и указания информации для редактирования. Формат:

[имя] RECORD [IDENT = (длина, 'имя', начало-имени)]
[FIELD = ([длина], { [начало-поля] }, [преобразование],
[литерал'] }, [начало-поля])]
[LABELS = n]

где IDENT = (длина, 'имя', начало-имени) — указывает последнюю запись входной группы записей, для которой в этом операторе задан операнд FIELD или оператор MEMBER;

FIELD = (длина, { [начало-поля] }, [преобразование], [начало-поля]) — определяет поле, подлежащее обработке, и информацию для редактирования;

длина — указывает размер (в байтах) входного поля или литерала, который следует обработать;

начало-поля — указывает начальный байт поля, который будет обрабатываться;

литерал — указывает литерал (максимальная длина 40 байт), который будет помещен в поле, начиная с байта, указанного как начало поля в выходных записях;

преобразование — указывает двухбайтовый код, определяющий тип преобразования, которому будет подвергаться это поле (PZ — из упакованного в распакованный десятичный формат);

ZP — из распакованного в упакованный десятичный формат; HE — из кода КОИ-8 в код ДКОИ;

начало-поля — указывает начало поля в выходных записях;

LABELS = n — указывает, что следующие n записей в управляющем наборе данных рассматриваются как метки пользователя ЭВМ (n = 1—8).

Коды возврата программы:

00 — успешное выполнение;

04 — выдано предупреждающее сообщение, но возможно успешное завершение;

08 — выполнение обработки набора данных было прекращено после запроса обработки начальных меток пользователя ЭВМ;

12 — встретилась ошибка, которую программа не может исправить (выполнение пункта задания прекращается);

16 — вызываемая программа передала код возврата 16 программе IEBGENER (выполнение пункта задания прекращается).

Программа IEBISAM. Обеспечивает копирование, разгрузку, загрузку и вывод на печать индексно-последовательного набора данных.

Для выполнения программы используются операторы языка управления заданиями JOB, EXEC, DD с именем SYSPRINT, операторы DD с именами SYSUT1 и SYSUT2 (определяют соответственно входной и выходной наборы данных). Запрос на выполняемые функции программы указывают в поле PARM оператора EXEC:

$$PARM = \left\{ \begin{array}{l} COPY \\ UNLOAD \\ LOAD \\ PRINTL \\ 'PRINTL[,N][,EXIT=имя-программы]' \end{array} \right\}$$

где COPY — указывает выполнение копирования;

UNLOAD — указывает выполнение разгрузки;

LOAD — указывает выполнение загрузки;

PRINTL — указывает выполнение распечатки с преобразованием каждой записи в шестнадцатеричное представление;

'PRINTL[,N][,EXIT=имя-программы]' — указывает выполнение распечатки (при указании N преобразование записей в шестнадцатеричное представление не производится, при указании EXIT=имя-программы управление передается перед распечаткой каждой записи вызываемой программе, которая должна быть включена либо в библиотеку задания, либо в общую библиотеку).

Коды возврата программы:

00 — успешное завершение;

04 — программе IEBISAM был передан код возврата 04 и 12 вызываемой программой;

08 — обнаружена ошибка; выполнение пункта задания прекращается;

12 — код возврата, отличающийся от 00, 04, 08 и 12, был передан из вызываемой программы программе IEBISAM; выполнение пункта задания завершается;

16 — обнаружена ошибка; выполнение пункта задания прекращается.

Программа IEBTRCH. Предназначена для вывода на печатающее или перфокарточное устройство вывода последовательных или библиотечных наборов данных.

Для выполнения программы используются операторы языка управления заданиями JOB, EXEC, DD с именами SYSPRINT, SYSIN, операторы DD с именами SYSUT1, SYSUT2 (определяют соответственно входной и выходной набор данных) и управляющие операторы программы: PRINT, PUNCH, TITLE, EXITS, MEMBER, RECORD, LABELS.

Оператор PRINT или PUNCH предназначен для указания запроса на распечатку или перфорацию соответственно. Формат:

MAXFLDS=*n* — указывает число, не меньшее общего количества операндов FIELD в последующих операторах RECORD;

MAXGPS=*n* — указывает число, не меньшее общего количества операндов IDENT в последующих операторах RECORD;

MAXLITS=*n* — указывает число, не меньшее общего количества символов, содержащихся в идентификаторах операндов IDENT последующих операторов RECORD;

INITPG=*n* — указывает номер первой страницы (затем страницы нумеруются последовательно);

MAXLINE=*n* — указывает максимальное количество строк на странице (по умолчанию предполагается *n*=60);

CDSEQ=*n* — указывает номер первой перфокарты в колоде (по умолчанию перфокарты не нумеруются);

CDINCR=*n* — указывает приращение для номера перфокарты, которое используется при нумерации (если CDINCR отсутствует, а CDSEQ указано, приращение принимается равным 10).

Оператор TITLE предназначен для указания заголовка, который должен предшествовать данным, выводимым на печать или перфорацию. Формат:

[имя] TITLE ITEM = ('заголовок'[,позиция])

где ITEM = ('заголовок'[,позиция]) — определяет информацию для заголовка;

'заголовок' — указывается в апострофах содержание заголовка, максимальная длина которого 40 байт;

позиция — указывается начальное положение заголовка в выходной записи (по умолчанию подразумевается 1).

Оператор EXITS предназначен для определения вызываемых программ и должен следовать непосредственно после оператора

TITLE или после оператора { PRINT } . Формат:

[имя] EXITS [INHDR=имя-программы]
[INTLR=имя-программы]
[INREC=имя-программы]
[OUTREC=имя-программы]

где INHDR=имя-программы — указывает имя программы обработки начальных меток пользователя ЭВМ;

INTLR=имя-программы — указывает имя программы обработки конечных меток пользователя ЭВМ;

INREC=имя-программы — указывает имя программы для дополнительной обработки каждой записи, прежде чем передать ее программе IEVBTPCH;

OUTREC=имя-программы — указывает имя программы для обработки каждой записи, прежде чем вывести ее на печать или перфорацию.

Оператор MEMBER предназначен для определения разделов, которые должны быть выведены на печать или отперфорированы. Формат:

[имя] MEMBER NAME = { имя-раздела
дополнительное-имя }

где NAME = { имя-раздела
дополнительное-имя } — указывает основное или дополнительное имя раздела.

Оператор RECORD предназначен для определения группы записей, которая должна быть напечатана или перфорирована в формате, определенном программистом. Формат:

[имя] RECORD [IDENT = (длина, 'имя', позиция-ввода)]
[FIELD = (длина, [позиция-ввода], [преобразование]
[позиция-вывода]) ...]

где IDENT — указывает идентификатор последней записи в группе записей, для которых определена обработка в операнде FIELD:

длина — указывает длину (в байтах) поля во входной записи идентификатора (длина не может превышать 8 байт);

'имя' — указывает идентификатор, определяющий последнюю запись в группе записей;

позиция ввода — указывает адрес начала поля во входной записи, которое содержит идентификатор;

FIELD — указывает поле обрабатываемой информации:

длина — указывает длину (в байтах) входного поля, которое должно быть обработано;

позиция ввода — указывает начальный байт обрабатываемого входного поля (по умолчанию подразумевается 1);

преобразование — указывает двухбайтовый код типа преобразования, выполняемого до вывода на печать или перфорацию: PZ — из упакованного в распакованный десятичный формат, XE — из буквенно-цифрового в шестнадцатеричное представление.

Оператор LABELS указывает, следует ли трактовать метки пользователя ЭВМ как данные. Формат:

[имя] LABELS
DATA = { YES
NO
ALL
ONLY }

где YES — все допустимые метки пользователя ЭВМ обрабатываются как данные;

NO — метки пользователя ЭВМ не обрабатываются как данные;

ALL — метки пользователя ЭВМ обрабатываются как данные независимо от кода возврата (если код возврата вызываемой программы равен 16, то после завершения обработки группы меток пользователя ЭВМ прекращается выполнение программы IEVBTPCH);

ONLY — только начальные метки пользователя ЭВМ обрабатываются как данные независимо от кода возврата (выполнение задания прекращается после возврата из программы OPEN).

Коды возврата программы:

00 — успешное завершение;

08 — раздел, определенный для печати, не найден во входном наборе данных (обрабатывается следующий раздел);

- 12 — обнаружены ошибки или вызываемая программа передала код возврата 12 программе IEВРТРСН (выполнение пункта задания прекращается);
- 16 — вызываемая программа передала код возврата 16 программе IEВРТРСН (выполнение пункта задания прекращается).

Программа IEВUPDATE. Предназначена для внесения изменений в последовательный или библиотечный набор данных, содержащий тексты программ на исходном языке. Обеспечивает следующие возможности:

создание и обновление библиотечных наборов данных;
 внесение изменений (заменить, вычеркнуть, добавить) в разделы библиотечного набора данных или в последовательный набор данных;

изменение организации набора данных из последовательной в библиотечную или наоборот.

Для выполнения программы используются операторы языка управления заданиями JOB, EXEC, операторы DD с именами SYSPRINT, SYSIN, операторы DD с именами SYSUT1 и SYSUT2 (определяют соответственно входной и выходной последовательный или библиотечный набор данных) и управляющие операторы программы: ADD, REPL, CHANGE, REPRO, NUMBER, DELETE, LABEL, ALIAS, ENDUP и *операторы данных* (в управляющих операторах, кроме операторов данных, должны указываться символы ./ в позициях 1, 2).

В поле PARM оператора EXEC можно задавать следующую информацию:

$$PARM = \left\{ \begin{array}{c} \text{NEW} \\ \text{MOD} \end{array} \right\}, [\text{прогр1}], [\text{прогр2}]$$

где NEW — указывает, что входные данные определены только в управляющем наборе данных и входной набор данных SYSUT1 не определяется;

MOD — указывает, что определены как управляющий набор данных, так и входной набор данных;

прогр1 — определяет имя вызываемой программы обработки начальной метки пользователя ЭВМ управляющего набора данных;

прогр2 — определяет имя вызываемой программы обработки конечной метки пользователя ЭВМ управляющего набора данных.

Функциональные операторы ADD, REPL, CHANGE и REPRO предназначены для задания функции программы. Формат:

./[имя]	$\left\{ \begin{array}{l} \text{ADD} \\ \text{REPL} \\ \text{CHANGE} \\ \text{REPRO} \end{array} \right\}$	$\left\{ \begin{array}{l} [\text{LIST}=\text{ALL}] \\ [\text{SEQFLD}=\text{ddl}] \\ [\text{NEW}=\left\{ \begin{array}{l} \text{PO} \\ \text{PS} \end{array} \right\}] \\ [\text{MEMBER}=\text{cccccccc}] \\ [\text{COLUMN}=\text{dd}] \\ [\text{UPDATE}=\text{INPLACE}] \\ [\text{INHDR}=\text{cccccccc}] \\ [\text{INTLR}=\text{cccccccc}] \end{array} \right\}$
---------	------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

```
[OUTHDR=ccccccc]
[OUTTLR=ccccccc]
[TOTAL=(имя, размер)]
[NAME=ccccccc]
[LEVEL=hh]
[SSI=hhhhhhh]
```

где ADD — указывает, что раздел или набор данных должен быть добавлен в выходной набор данных;

REPL — указывает, что определенные в операторе входные данные заменяют последовательный набор данных или раздел библиотечного набора данных;

CHANGE — указывает, что выполняется модификация в существующем разделе или наборе данных (использование этого оператора без операторов NUMBER или DELETE и без операторов данных вызывает ошибку);

REPRO — указывает, что раздел или набор данных копируется в новый набор данных;

LIST=ALL — указывает, что на печать выводятся обновленный раздел или набор данных и управляющие операторы программы (по умолчанию на печать выводятся только выполненные модификации и управляющие операторы);

SEQFLD=ddl — указывает номер начальной позиции dd и длину l (не больше 8 байт) для порядковых номеров записей и последовательности операторов данных (dd+l не должно превышать LRECL+1 (по умолчанию принимается SEQFLD=738, т. е. 8-байтовый порядковый номер начинается в позиции 73));

NEW=PO — указывает, что входной последовательный набор данных преобразуется в раздел библиотечного набора данных;

NEW=PS — указывает, что раздел входного библиотечного набора данных преобразуется в последовательный набор данных;

MEMBER=ssssss — присваивает имя разделу, помещаемому в библиотечный набор данных, определяемый оператором DD с именем SYSUT2 (операнд кодируют, только если в SYSUT1 определен последовательный набор данных, в SYSUT2 — библиотечный набор данных и указан операнд NEW=PO);

COLUMN=dd — используется только с оператором CHANGE и указывает начальную позицию поля данных в записи (dd — десятичное число);

UPDATE=INPLACE — указывает, что набор данных модифицируется на месте (следует учесть следующее: оператор DD с именем SYSUT2 не кодируется, в операторе EXEC следует указать параметр MOD, в одном пункте задания допускаются только один оператор CHANGE и один операнд UPDATE=INPLACE, можно указывать только функции замещения и перенумерации меток и модификации начальных меток пользователя ЭВМ);

INHDR=ssssss — указывает имя программы обработки начальных меток пользователя ЭВМ входного набора данных;

INTLR=ssssss — указывает имя программы обработки конечных меток пользователя ЭВМ входного набора данных;

OUTHDR=cccccccc — указывает имя программы обработки начальных меток пользователя ЭВМ выходного набора данных;

OUTTLR=cccccccc — указывает имя программы обработки конечных меток пользователя ЭВМ выходного набора данных;

TOTAL=(имя, размер) — указывает, что до внесения каждой записи в выходной набор данных происходит передача управления вызываемой программе создания итоговых данных (в скобках указывается имя вызываемой программы и размер основной памяти для данных пользователя ЭВМ, который не должен превышать 32 Кбайт и должен быть не менее 2 байт).

Примечания: 1. При указании операнда TOTAL следует указать в операторе DD с именем SYSUT2 параметр OPTCD=T.

2. Операнды INHDR, INTLR, OUTHDR, OUTTLR, TOTAL можно указывать только при обработке последовательных наборов данных.

3. Нельзя одновременно указывать операнды INTLR, OUTHDR, OUTTLR и UPDATE=INPLACE;

NAME=cccccccc — указывает имя раздела, помещаемого в библиотечный набор данных (операнд NAME указывается для каждого входного раздела);

LEVEL=hh — указывает номер текущего обновления в шестнадцатеричном представлении (00 — FF);

SSI=hhhhhhhh — указывает восемь шестнадцатеричных символов индикатора состояния модуля.

Операторы детализации NUMBER, DELETE используются с функциональными операторами для перенумерации или удаления определенных записей. Формат:

./[имя] { NUMBER } [SEQ1=ccccccc] используются с оператором
 { DELETE } [SEQ2=ccccccc] NUMBER или DELETE.
 [SEQ1=ALL]

[NEW1=ccccccc] используются только с оператором
[INCR=ccccccc] NUMBER,
[INSERT=YES] .

где NUMBER — указывает выполнение перенумерации записей в разделе или наборе данных (допускается только один оператор NUMBER с каждым оператором ADD или REPL и несколько операторов NUMBER с каждым оператором CHANGE);

DELETE — используется с оператором CHANGE для удаления записей из раздела или набора данных;

SEQ1=ccccccc — указывает порядковый номер первой записи, с которой должна начаться обработка;

SEQ2=ccccccc — указывает порядковый номер последней записи, на которой должна закончиться обработка;

SEQ1=ALL — определяет перенумерацию записей всего раздела или набора данных;

NEW1=ccccccc — указывает первый порядковый номер, который присваивается новой включаемой записи;

INCR=ccccccc — определяет приращение, на которое увеличивается порядковый номер от записи к записи.

Примечание. Операнды SEQ1, SEQ2, NEW1, INCR указывают десятичное число, содержащее от 1 до 8 символов. Можно указывать только значащую величину, например вместо SEQ1=00000010 может быть записано SEQ1=10;

INSERT=YES — используется, если оператор NUMBER применяется в сочетании с оператором CHANGE и указывает включение записей, которые являются операторами данных и содержат в поле нумерации пробел. Следует учесть следующее: операнд SEQ1 указывает запись во входном наборе данных, после которой следует включить данные, операнд SEQ2 не указывается, нельзя указывать SEQ1=ALL, операнд NEW1 указывает порядковый номер для первого включаемого оператора данных, операнд INCR используется при перенумерации раздела или набора данных начиная с первой включенной записи.

Операторы данных используются с функциональным оператором или в совокупности с функциональным оператором и оператором детализации. Каждый оператор данных представляется в формате 80-позиционной перфокарты и начинается в первой позиции. Оператор данных предназначен для замены записи во входном наборе данных или использования наряду с записями входного набора данных. Оператор данных в сочетании с оператором CHANGE содержит новые данные или данные, предназначенные для замены, а в сочетании с операторами ADD или REPL содержит новые данные для включения в выходной набор данных.

Оператор LABEL указывает, что последующие операторы данных должны трактоваться как метки пользователя ЭВМ выходного набора данных. Формат:

./[имя] LABEL

При выполнении программы допускается не более двух операторов LABEL.

Оператор ENDUP используется для указания конца управляющих операторов программы. Формат:

./[имя] ENDUP

Коды возврата программы:

- 00 — успешное завершение;
- 04 — ошибка в управляющем операторе программы;
- 12 — обнаружена ошибка, и выполнение пункта задания прекращается;
- 16 — из программы обработки меток пользователя ЭВМ получен код возврата 16, и выполнение пункта задания прекращается.

15.3. Системные программы обслуживания

15.3.1. Общие сведения

Задания на выполнение системных программ обслуживания содержат набор операторов языка управления заданиями и уп-

равляющих операторов программы, расположенных в следующей последовательности:

```

//имя      JOB ...
//[имя]     EXEC PGM=имя-программы
//SYSPRINT DD ...
//имяDD1   DD ...
.
.
.
//имяDDK   DD ...
//SYSIN    DD ...
/*

```

Операторы DD языка управления заданиями, необходимые для каждой конкретной программы, описаны далее в сведениях о соответствующей программе. Формат управляющих операторов программ и правила их подготовки те же, что и для управляющих операторов программ обслуживания данных.

15.3.2. Возможности и управляющие операторы системных программ обслуживания

Программа IENATLAS. Предназначена для исправления на томе прямого доступа ошибок в данных путем выделения альтернативной дорожки и пересылки данных на альтернативную дорожку, на которой исправляются дефектные записи.

Для выполнения программы используются следующие операторы языка управления заданиями: операторы JOB, EXEC, DD с именами SYSPRINT, SYSIN, оператор DD с именем SYSUT1 (определяет набор данных, содержащий дефектные записи) и управляющие операторы программы: TRACK, VTOC.

Формат операторов TRACK и VTOC:

```

TRACK=bbbbccccchhhrrkkddd[S]
VTOC=bbbbccccchhhrrkkddd

```

где TRACK — указывает, что альтернативные дорожки предназначены для дорожек, которые не содержат записей оглавления тома;

VTOC — указывает, что альтернативные дорожки предназначены для дорожек, которые содержат записи оглавления тома;

bbbb — содержат нули;

cccc — номер цилиндра, в котором была обнаружена дефектная дорожка;

hhhh — номер дефектной дорожки;

rrkk — номер записи и длина ключа для дефектной записи;

dddd — длина данных в дефектной записи (если используется команда канала ЗАПИСЬ СЧЕТЧИКА, КЛЮЧА и ДАННЫХ (СПЕЦИАЛЬНАЯ), то dddd — длина сегмента записи);

S — указывает, что используется команда канала ЗАПИСЬ СЧЕТЧИКА, КЛЮЧА и ДАННЫХ (СПЕЦИАЛЬНАЯ).

Входные данные непосредственно следуют за управляющим оператором программы и представляются в шестнадцатеричном виде.

Программа IEHDASDR. Предназначена для инициализации томов прямого доступа для НМД типа ЕС-5050, ЕС-5061, ЕС-5066, изменения регистрационного номера инициализированного тома прямого доступа, назначения альтернативных дорожек для указанных дефектных дорожек тома, создания копии тома прямого доступа или распечатки его содержимого, восстановления скопированных данных с магнитной ленты на том прямого доступа, внесения на том прямого доступа записей IPL и программы IPL. Для выполнения программы используются операторы языка управления заданиями JOB, EXEC, DD с именами SYSPRINT, SYSIN, операторы DD для определения каждого НМД и НМЛ, используемого в пункте задания, в виде

```
//имяdd DD UNIT=xxxx,VOLUME=SER=xxxxxx,DISP=OLD
```

и управляющие операторы программы: ANALYZE, FORMAT, LABEL, GETALT, DUMP, RESTORE, PUTIPL.

Оператор EXEC может кодироваться следующим образом:

```
// EXEC PGM=IEHDASDR [ ,PARM='LINECNT=xx'  
                      [ ,PARM='N=n'  
                      [ ,PARM='LINECNT=xx,N=n' ] ]
```

где LINECNT=xx — указывает десятичное число строк на странице в распечатке xx от 1 до 99 (по умолчанию 58);

N=n — указывает десятичное число от 1 до 6 и определяет максимальное число одинаковых функций, которые могут одновременно выполняться программой (по умолчанию одновременно может выполняться до шести функций ANALYZE, FORMAT, DUMP или RESTORE).

Инициализация томов прямого доступа с выполнением анализа качества дорожек задается оператором ANALYZE. Формат:

```
[имя] ANALYZE { TODD=(сущ,...) }  
               { TODD=(имя dd,...) }  
               VTOC=xxxxx  
               EXTENT=xxxxx  
               [NEWVOLID=регистрационный]  
               [IPLDD=имяdd]  
               [FLAGTEST= { YES  
                           NO } ]  
               [PASSES= { n  
                           0 } ]  
               [OWNERID=имя]  
               [PURGE= { YES  
                        NO } ]
```

где TODD=(сущ,...) — указывает физический адрес периферийного устройства, содержащего обрабатываемый том (оператор DD для описания этого устройства кодировать не нужно, а перед выполнением задания указанное устройство должно быть переведено в автономное состояние с помощью команды VARY OFFLINE);

TODD=(имяdd,...) — указывает имя оператора DD, определяющего периферийное устройство с обрабатываемым томом (для каждого имени должен быть закодирован оператор DD);

VTOC=xxxxxx — указывает десятичный относительный адрес дорожки, на которой должно начинаться оглавление тома (оглавление не может занимать нулевую дорожку и первую дорожку, если на томе записан текст программы IPL);

EXTENT=xxxxxx — указывает длину оглавления тома в дорожках;

NEWVOLID=регистрационный — указывает новый регистрационный номер тома (по умолчанию сохраняется прежний регистрационный номер);

IPLDD=имяdd — указывает имя оператора DD, определяющего набор данных, содержащий программу IPL (если текст программы IPL включается во входной поток, то для отделения оператора ANALYZE от текста программы IPL используется оператор IPLTXT);

FLAGTEST — указывает, требуется ли проверка предварительно помеченных дорожек: NO — не проверяются, YES — проверяются;

PASSES=n — указывает, что проверка записи стандартной комбинации бит должна выполняться n раз (n=1—255); PASSES=0 означает, что не выполняются анализ и разметка дорожек, а записываются только оглавление тома и нулевая запись дорожки (для томов НМД типа ЕС-5066 указывается только это значение); PASSES=0 — указывается только для ранее инициализированных томов прямого доступа;

OWNERID=имя — указывает помещаемую в метку идентифицирующую информацию (от 1 до 10 символов в коде ДКОИ, исключая пробел и запятую);

PURGE — указывает режим обработки наборов данных с истекшим сроком хранения (YES — стирание наборов данных определяется ответом оператора: U — все наборы данных могут быть стерты, T — наборы данных не могут быть стерты; NO — если встречается на томе набор данных с истекшим сроком хранения, обработка тома прекращается).

Примечание. Операнд PURGE не применим к наборам данных, защищенным паролем, т. е. на каждый защищенный набор данных оператор ЭВМ должен ответить паролем. Если он не сможет указать соответствующий пароль, то обработка тома прекращается.

Инициализация тома прямого доступа без выполнения анализа качества дорожек задается оператором FORMAT. Формат:

```
[имя] FORMAT TODD=(имяdd,...)
              VTOC=xxxxx
              EXTENT=xxxxx
              [NEWVOLID=регистрационный]
              [IPLDD=имяdd]
              [OWNERID=имя]
              [PURGE={ YES }
                  { NO }]
```

где значения операндов те же, что и для операндов оператора ANALYZE.

Изменение регистрационного номера тома прямого доступа задается оператором LABEL.

Формат:

```
[имя] LABEL TODD={ сии  
имяdd }  
NEWVOLID=регистрационный  
[OWNERID=имя]
```

где значения операндов те же, что и для операндов оператора ANALIZE.

Назначение альтернативных дорожек для указанных дефектных дорожек тома осуществляется оператором GETALT. Формат:

```
[имя] GETALT TODD=имяdd  
TRACK=ccccchhhh
```

где TODD=имяdd — указывает имя оператора DD, определяющего НМД с томом, на котором назначается альтернативная дорожка;

TRACK=ccccchhhh — указывает в шестнадцатеричном представлении номер цилиндра (cccc) и номер дорожки (hhhh), для которой требуется альтернативная (нельзя указывать адрес нулевой или первой дорожки).

Дамп дорожки, группы дорожек или всего тома прямого доступа задается оператором DUMP. Формат:

```
[имя] DUMP FROMDD=имяdd  
TODD=(имяdd,...)  
[CPYVOLID={YES  
NO }]  
[BEGIN=ccccchhhh]  
[END=ccccchhhh]  
[PURGE={YES  
NO }]
```

где FROMDD=имяdd — указывает имя оператора DD, определяющего НМД с томом, с которого должен быть выполнен дамп;

TODD=(имяdd,...) — указывает имя оператора DD, определяющего печатное устройство (SYSPRINT), либо имена операторов DD, определяющих периферийные устройства с томами прямого доступа или томами магнитных лент, на которые выполняется дамп (при задании распечатки тома для НМД типа EC-5066 в операторе DD с именем SYSPRINT должен быть указан операнд SPACE);

CPYVOLID=YES — указывает, что всем томам прямого доступа, на которые выполняется дамп, присваивается регистрационный номер исходного тома; по умолчанию или при указании CPYVOLID=NO все тома, на которые выводится дамп, сохраняют свои регистрационные номера;

BEGIN=ccccchhhh — указывает в шестнадцатеричном представлении номер цилиндра (cccc) и номер первой обрабатываемой дорожки (hhhh) (если BEGIN опущен, обработка начинается с нулевой дорожки);

END=ccccchhhh — указывает в шестнадцатеричном представлении номер цилиндра (cccc) и номер последней обрабатываемой дорожки (hhhh) (если END опущен, то последней обрабатываемой дорожкой является последняя основная дорожка).

Примечание. Для указания обработки одной дорожки в операндах BEGIN и END указывают один и тот же адрес дорожки.

PURGE — указывает режим обработки наборов данных с истекшим сроком хранения (YES — стирание наборов данных определяется ответом оператора: U — все наборы данных могут быть стерты, T — наборы данных не могут быть стерты; NO — если встречается на томе набор данных с истекшим сроком хранения, обработка тома прекращается).

Примечание. Операнд PURGE не применим к наборам данных, защищенным паролем, т. е. на каждый защищенный набор данных оператор ЭВМ должен отвечать паролем. Если он не сможет указать соответствующий пароль, то обработка тома прекращается.

Восстановление на том прямого доступа данных с магнитной ленты, на которую был выполнен дамп, задается оператором RESTORE. Формат:

```
[имя] RESTORE TODD=(имяdd,...)
FROMDD=имяdd
[CPYVOLID={YES}
NO}]
[PURGE={YES}
NO}]
```

где TODD=(имяdd,...) — указывает имена операторов DD, определяющих НМД с томами, на которые восстанавливаются данные;

FROMDD=имяdd — указывает имя оператора DD, который определяет НМЛ с томом, содержащим дамп.

Значения операндов CPYVOLID и PURGE те же, что и для операторов оператора DUMP.

Оператор IPLTXT предназначен для обозначения начала текста программы начальной загрузки и помещается перед текстом программы во входном потоке. Формат:

```
[имя] IPLTXT
```

Оператор PUTIPL предназначен для обеспечения внесения на дорожку 0 цилиндра 0 тома прямого доступа записей IPL и программы начальной загрузки. Формат:

```
[имя] PUTIPL FROMDD=имяdd
TODD=имяdd
[PURGE={YES}
NO}]
```

где FROMDD=имяdd — указывает имя оператора DD, который определяет входной набор данных;

TODD=имяdd — указывает имя оператора DD, который определяет выходной том (в операнде DISP указывается значение OLD);

PURGE — указывает, можно ли затирать метки пользователя ЭВМ: YES — программа начальной загрузки и вспомогательные записи IPL могут быть записаны на метки пользователя ЭВМ

и другие данные, которые следуют за меткой тома; NO — метки пользователя ЭВМ затираются нельзя.

Коды возврата программы:

- 00 — успешное выполнение;
- 04 — успешное выполнение с предупреждающим сообщением;
- 08 — запрошенная функция выполнена неудачно;
- 16 — ошибка во время вызова программы IEHDASDR или программа не смогла открыть входной набор данных либо набор данных для сообщений; выполнение пункта задания прекращается.

Программа IEINITT. Предназначена для инициализации томов магнитных лент. Для выполнения программы используются операторы управления заданиями JOB, EXEC, DD с именами SYSPRINT, SYSIN, оператор DD в следующем виде:

```
//имяdd DD DCB=(DEN=x),UNIT=(xxxx,n,DEFER),
```

который определяет НМЛ или группу НМЛ, используемых при инициализации. Управляющий оператор программы: INITT. Имя управляющего оператора программы должно совпадать с именем оператора DD, определяющего НМЛ.

Инициализация тома магнитной ленты задается оператором INITT. Формат:

```
[имя] INITT  SER=xxxxxx  
              OWNER='ccccccccc'  
              NUMBTAPE=n  
              [DISP={REWIND}  
                {UNLOAD}]]
```

где имя — указывает имя управляющего оператора, совпадающее с именем оператора DD, определяющего одно или несколько НМЛ;

SER=xxxxxx — определяет регистрационный номер тома магнитной ленты; при разметке нескольких томов регистрационный номер должен содержать только цифры, и заданный регистрационный номер тома увеличивается на единицу для каждого последующего размечаемого тома магнитной ленты;

OWNER='ccccccc' — определяет информацию пользователя ЭВМ (не более 10 символов);

NUMBTAPE=n — определяет число размечаемых томов (n=1—255), по умолчанию размечается один том;

DISP=REWIND — указывает, что после записи меток магнитную ленту можно перемотать, но не выполнять разгрузку;

DISP=UNLOAD — указывает, что после записи меток магнитную ленту нужно разгрузить.

Коды возврата программы:

- 00 — успешное завершение программы;
- 04 — успешное завершение программы, однако пользователь ЭВМ не определил набор данных для сообщений;
- 08 — во время обработки имелись ошибки ввода-вывода;

12 — во время обработки имелись ошибки ввода-вывода, и пользователь ЭВМ не определил набор данных для сообщений;

16 — прекращение выполнения программами из-за ошибок при чтении управляющего набора данных.

Программа IENIOSUP. Предназначена для обновления относительных адресов в таблицах управления передачей управления, содержащихся в модулях библиотеки супервизора (SYS1.SVCLIB) и должна использоваться всякий раз после перемещения библиотеки SYS1.SVCLIB или изменения в ее содержании.

Для выполнения программы используются следующие операторы языка управления заданиями: операторы JOB, EXEC, DD с именем SYSPRINT и DD с именем SYSUT1, который определяет входной набор данных SYS1.SVCLIB. В случае работы ОС ЕС с системой разделения времени необходимо указать в операторе EXEC параметр PARM=TSO. Управляющие операторы программы не предусмотрены.

Коды возврата программы:

00 — успешное завершение;

12 — возникла некорректируемая ошибка ввода-вывода и выполнение пункта задания прекращается.

Программа IENLIST. Предназначена для вывода на печать элементов каталога, элементов справочника одного или нескольких библиотечных наборов данных, элементов оглавления тома прямого доступа.

Для выполнения программы используются операторы языка управления заданиями JOB, EXEC, DD с именами SYSPRINT, SYSIN и DD в следующем виде:

```
//имяdd DD UNIT=xxxx,DISP=OLD,VOLUME=SER=xxxxxx
```

который определяет требуемый том прямого доступа. Управляющие операторы программы: LISTCTLG, LISTPDS, LISTVTOC.

Для задания распечатки всего каталога или определенной его части используется оператор LISTCTLG. Формат:

```
[имя] LISTCTLG [VOL=тип=регистрационный]  
[NODE=имя]
```

где VOL=тип=регистрационный — определяет тип НМД и регистрационный номер тома, на котором находится заданная часть каталога (по умолчанию предполагается, что каталог находится на резидентном томе операционной системы);

NODE=имя — указывает определенное имя (распечатываются все имена каталогизированных наборов данных, входящих в группу с указанным именем).

Для задания распечатки справочника одного или нескольких библиотечных наборов данных, находящихся на одном томе, используется оператор LISTPDS. Формат:

```
[имя] LISTPDS DSNAME=(имя[,имя]...)  
[VOL=тип=регистрационный]  
{DUMP  
FORMAT}
```

где DSNAME=(имя[,имя]...) — указывает имена библиотечных наборов данных, справочники которых должны быть распечатаны (допускается указывать до 10 имен; при указании одного имени скобки могут быть опущены);

VOL=тип=регистрационный — определяет тип НМД и регистрационный номер тома, на котором находятся библиотечные наборы данных (по умолчанию предполагается, что наборы данных находятся на резидентном томе операционной системы);

DUMP — указывает, что распечатка должна быть получена в неотредактированном шестнадцатеричном представлении;

FORMAT — указывает, что распечатка должна быть получена в отредактированном формате (отредактированный формат может быть задан для разделов библиотечного набора данных, созданных Редактором связи).

Для задания распечатки части или всего оглавления тома прямого доступа используется оператор LISTVTOC. Формат:

```
[имя] LISTVTOC { {DUMP } }  
                { {FORMAT } }  
                [DATE=dddyu]  
                [VOL=тип=регистрационный]  
                [DSNAME=(имя[,имя]...)]
```

где DUMP — указывает, что распечатка должна быть в неотредактированном шестнадцатеричном представлении;

FORMAT — указывает, что должна быть получена полностью отредактированная распечатка (если DUMP и FORMAT опущены, распечатка будет в частично отредактированном формате);

DATE=dddyu — указывает, что каждая метка набора данных помечается звездочкой (*), если истек срок хранения набора данных до указанной даты (ddd — день года, уу — последние две цифры года);

VOL=тип=регистрационный — определяет тип НМД и регистрационный номер тома, оглавление которого должно быть распечатано (по умолчанию предполагается резидентный том операционной системы);

DSNAME=(имя[,имя]...) — указывает имена наборов данных, информация о которых должна быть распечатана (допускает указывать до 10 имен); по умолчанию оглавление тома распечатывается полностью.

Коды возврата программы:

00 — успешное завершение;

08 — возникновение ошибки во время обработки управляющего оператора программы, запрос не был выполнен, обработка продолжается;

12 — возникла некорректируемая ошибка ввода-вывода; выполнение задания прекращается;

16 — некорректируется ошибка ввода-вывода во время чтения набора данных; выполнение задания прекращается.

Программа IENMOVE. Предназначена для копирования или пересылки наборов данных с последовательной, библиотечной и

прямой организацией. Операция пересылки допустима только для набора данных на томе прямого доступа и характерна тем, что после создания копии стирается исходный набор данных.

Программа также обеспечивает: получение копии многотомных наборов данных, каталога, группы каталогизированных наборов данных, тома прямого доступа; объединение библиотечных наборов данных; выборку разделов для копирования; исключение разделов из операции копирования; переименование разделов; замену содержания раздела; разгрузку наборов данных с томов прямого доступа.

Для выполнения программы IEHMOVE используются операторы языка управления заданиями JOB, EXEC, DD с именами SYSPRINT, SYSIN, SYSUT1 — определяет том прямого доступа, содержащий рабочий набор, и записывается в следующем виде:

```
//SYSUT1 DD UNIT=xxxx,VOLUME=SER=xxxxxx,DISP=OLD
```

а также следующие операторы DD:

```
//имяdd DD UNIT=xxxx,VOLUME=SER=xxxxxx,DISP=OLD
```

указывается для каждого постоянно установленного тома, относящегося к пункту задания, в том числе и для резидентного тома операционной системы;

```
//имяdd DD UNIT=xxxx,VOLUME=SER=xxxxxx,DISP=OLD
```

определяет НМД, доступный для установки тома, используемого в пункте задания. Когда число томов, которые должны быть обработаны, больше числа НМД, определенных операторами DD, необходимо указать отсроченную установку тома:

```
..., DISP=(..., KEEP), VOLUME=(PRIVATE,...), UNIT=(xxxx, DEFER).  
//имяdd DD DSNAME=xxxxxxxxx, UNIT=xxxx, VOLUME=SER=xxxxxx,  
// DISP=(..., KEEP), LABEL=(..., ...), DCB=(DEN=x)
```

определяет НМЛ для установки тома магнитной ленты, который используется при пересылке или копировании.

Используются следующие управляющие операторы программы: MOVE, COPY, INCLUDE, EXCLUDE, REPLACE, SELECT.

Пересылка набора данных задается оператором MOVE в следующем формате:

```
[имя] MOVE DSNAME=имя  
TO=тип=список  
[FROM=тип=список]  
[CVOL=тип=регистрационный]  
[UNCATLG]  
[RENAME=имя]  
[FROMDD=имяdd]  
[TODD=имяdd]
```

где DSNAME=имя — указывает имя набора данных, который должен быть переслан;

TO=тип=список — указывается том (или тома), на котором (на которых) должна быть создана копия набора данных;

FROM=тип=список — указывает том (или тома), на котором (на которых) находится некаталогизированный исходный

набор данных (если набор данных каталогизирован, FROM можно не указывать);

CVOL=тип=регистрационный — указывает тип НМД и регистрационный номер тома, на котором должен быть начат поиск каталога (если не указано ни CVOL, ни FROM, то считается, что набор данных каталогизирован на резидентном томе операционной системы);

UNCATLG — указывает, что запись в каталоге, относящаяся к исходному набору данных, должна быть удалена;

RENAME=имя — указывает, что набор данных должен быть переименован и задает его новое имя;

FROMDD=имяdd — указывает имя оператора DD, содержащего информацию о метке тома и порядковом номере набора данных (для набора данных, находящегося на магнитной ленте);

TODD=имяdd — указывает имя оператора DD, содержащего информацию о номере набора данных и метке тома магнитной ленты, на котором будет находиться пересланный набор данных (информация о RECFM, LRECL, BLKSIZE игнорируется).

Копирование набора данных задается оператором COPY в следующем формате:

```
[имя] COPY  DSNAME=имя
              TO=тип=список
              [FROM=тип=список]
              [CVOL=тип=регистрационный]
              [UNCATLG]
              [CATLG]
              [RENAME=имя]
              [FROMDD=имяdd]
              [TODD=имяdd]
```

где DSNAME=имя — указывает имя набора данных, который копируется;

CATLG — указывает, что скопированный набор данных должен быть каталогизирован на томе, на который он помещен, если это том прямого доступа (если каталог на этом томе отсутствует, то создается каталог); значения остальных операндов те же, что и для операндов в операторе MOVE DSNAME....

Пересылка групп наборов данных, которые каталогизированы на одном томе, задается оператором MOVE в следующем формате:

```
[имя] MOVE  DSGROUP[=имя]
              TO=тип=список
              [CVOL=тип=регистрационный]
              [PASSWORD]
              [UNCATLG]
              [TODD=имяdd]
```

где DSGROUP=имя — указывает имя индекса структуры для каталогизированных наборов данных, которые должны быть пересланы (если имя не указано, то пересылаются все наборы данных, каталогизированные на указанном томе);

TO=тип=список — указывает том или тома, на которые должна быть скопирована указанная группа наборов данных;

CVOL=тип=регистрационный — указывает тип НМД и регистрационный номер тома, на котором должен быть начат поиск каталога для наборов данных (по умолчанию считается, что группа наборов данных каталогизирована на резидентном томе операционной системы);

PASSWORD — указывает, что защищенные паролем наборы данных, содержащиеся в группе, должны быть пересланы (по умолчанию пересылаются только незащищенные наборы данных);

UNCATLG — указывает, что записи в каталоге, относящиеся к указанной группе наборов данных, должны быть удалены;

TODD=имяdd — (для группы, которая должна быть переслана на том магнитной ленты) указывает имя оператора DD, содержащего информацию о порядковом номере набора данных и метке тома, на который должны быть пересланы наборы данных (информация о RECFM, LRECL BLKSIZE игнорируется).

Копирование групп наборов данных, которые каталогизированы на одном томе, задается оператором COPY в следующем формате:

```
[имя] COPY DSGROUP [=имя]
      TO=тип=регистрационный
      [CVOL=тип=регистрационный]
      [PASSWORD]
      [UNCATLG]
      [CATLG]
      [TODD=имяdd]
```

где DSGROUP=имя — указывает имя индекса структуры для каталогизированных наборов данных, которые должны быть скопированы (если имя не указано, то копируются все наборы данных, каталогизированные на указанном томе);

CATLG — указывает, что скопированные наборы данных должны быть каталогизированы на носителе данных, на котором они находятся. Значения остальных операндов те же, что и для оператора MOVE DSGROUP....

Пересылка библиотечных наборов данных задается оператором MOVE в следующем формате:

```
[имя] MOVE PDS=имя
      TO=тип=регистрационный
      [FROM=тип=регистрационный]
      [CVOL=тип=регистрационный]
      [EXPAND=nn]
      [UNCATLG]
      [RENAME=имя]
      [FROMDD=имяdd]
      [TODD=имяdd]
```

где PDS=имя — указывает имя библиотечного набора данных, который должен быть переслан;

EXPAND=nn — указывает число 256-байтовых записей (до 99 записей), которые должны быть добавлены в оглавлении указанного библиотечного набора данных (не может быть указан, если внешняя память для набора данных на НМД была распре-

делена предварительно). Значения остальных операндов те же, что и для операндов оператора MOVE DSGROUP. В операторах DD, задаваемых операндами FROMDD и TODD, необходимо указывать следующие характеристики DCB разгруженного набора данных: RECFM=FB, LRECL=80, BLKSIZE=800.

Копирование библиотечных наборов данных задается оператором COPY в следующем формате:

```
[имя] COPY PDS=имя
          TO=тип=регистрационный
          [FROM=тип=регистрационный]
          [CVOL=тип=регистрационный]
          [EXPAND=nn]
          [UNCATLG]
          [CATLG]
          [RENAME=имя]
          [FROMDD=имяdd]
          [TODD=имяdd]
```

где PDS=имя — указывает имя библиотечного набора данных, который должен быть скопирован;

EXPAND=nn — указывает число 256-байтовых записей (до 99 записей), которые должны быть добавлены в оглавление указанного библиотечного набора данных (не может быть указан, если внешняя память на НМД была распределена предварительно);

CATLG — указывает, что скопированный библиотечный набор данных должен быть каталогизирован на томе прямого доступа, на котором он находится (если каталог отсутствует, то он создается). Значения остальных операндов те же, что и для операндов в операторе MOVE DSGROUP.

Пересылка элементов каталога без пересылки наборов данных, связанных с этими элементами, задается оператором MOVE в следующем формате:

```
[имя] MOVE CATALOG [=имя]
          TO=тип=регистрационный
          [CVOL=тип=регистрационный]
          [FROM=тип=регистрационный]
          [FROMDD=имя dd]
          [TODD=имя dd]
```

где CATALOG=имя — указывает имя элемента каталога, который должен быть переслан (если имя не указано, пересылаются все элементы каталога);

TO=тип=регистрационный — указывает тип НМД и регистрационный номер тома, на котором должен быть начат поиск каталога;

CVOL=тип=регистрационный — см. оператор MOVE DSGROUP;

FROM=тип=регистрационный — указывает тип внешнего запоминающего устройства и регистрационный номер тома, на котором находится разгруженный вариант каталога (если не указаны ни FROM, ни CVOL, то предполагается, что каталог расположен на резидентном томе операционной системы);

FROMDD=имяdd (для каталога, разгруженного на томах магнитной ленты) — указывает имя оператора DD, который должен задавать номер набора данных, метку тома магнитной ленты, характеристики разгруженного каталога: RECFM=FB, LRECL=80, BLKSIZE=800;

TODD=имяdd (для каталога, который должен быть разгружен на том магнитной ленты) — указывает имя оператора DD, в котором задается номер набора данных, метка тома магнитной ленты, характеристики разгруженного каталога: RECFM=FB, LRECL=80, BLKSIZE=800.

Копирование элементов каталога без копирования набора данных, связанных с этими элементами, задается оператором COPY в следующем формате:

```
[имя] COPY CATALOG [=имя]
      TO=тип=регистрационный
      [CVOL=тип=регистрационный]
      [FROM=тип=регистрационный]
      [FROMDD=имяdd]
      [TODD=имяdd]
```

где CATALOG=имя — указывает имя элемента каталога, который должен быть скопирован (если имя не указано, копируются все элементы каталога). Значения остальных операндов те же, что и для операндов оператора MOVE CATALOG...

Пересылка всех наборов данных, находящихся на томе прямого доступа, задается оператором MOVE в следующем формате:

```
[имя] MOVE VOLUME=тип=регистрационный
      TO=тип=список
      [PASSWORD]
      [TODD=имяdd]
```

где VOLUME=тип=регистрационный — указывает тип НМД и регистрационный номер исходного тома;

TO=тип=список — указывает том или тома, на которые должны быть пересланы наборы данных;

PASSWORD — указывает, что защищенные паролем наборы данных должны быть включены в операцию пересылки (по умолчанию пересылаются только незащищенные наборы данных);

TODD=имяdd (для томов магнитной ленты, на которую разгружается исходный том) — указывает имя оператора DD, содержащего информацию о DCB, номере набора данных и метке тома магнитной ленты.

Копирование всех наборов данных, находящихся на томе прямого доступа, задается оператором COPY в следующем формате:

```
[имя] COPY VOLUME=тип=регистрационный
      TO=тип=список
      [PASSWORD]
      [CATLG]
      [TODD=имяdd]
```

где VOLUME=тип=регистрационный — указывает тип НМД и регистрационный номер исходного тома;

CATLG — указывает, что все скопированные наборы данных должны быть каталогизированы в наборе данных SYSCATLG на том томе, на котором они находятся. Значения остальных операндов те же, что и для операндов оператора MOVE VOLUME....

Оператор INCLUDE дополняет возможности операторов MOVE и COPY и используется для задания включения раздела или набора данных, неявно определенных в этих операторах в форматах с операндами DSGROUP и PDS.

Оператор INCLUDE следует за оператором MOVE или COPY. Формат:

```
[имя] INCLUDE DSNAME=имя  
[MEMBER=имя-раздела]  
[FROM=тип=список]  
[CVOL=тип=регистрационный]
```

где DSNAME=имя — указывает имя набора данных (при пересылке или копировании группы наборов данных в группу включается названный набор данных; при пересылке или копировании библиотечного набора данных названный библиотечный набор данных полностью включается или раздел из этого набора данных);

MEMBER=имя-раздела (только для операторов MOVE или COPY в формате с операндом PDS) — указывает имя раздела библиотечного набора данных, названного в операнде DSNAME (этот раздел объединяется с библиотечным набором данных, который переслан или копирован; его характеристики должны совпадать с характеристиками этого набора данных); набор данных, содержащий этот раздел, не вычеркивается независимо от заданного оператора (MOVE или COPY);

FROM=тип=список — указывает том или тома, на которых находится набор данных, если набор данных не каталогизирован;

CVOL=тип=регистрационный — указывает тип НМД и регистрационный номер тома, на котором должен быть начат поиск каталога (если не указаны FROM и CVOL, то считается, что набор данных каталогизирован на резидентном томе операционной системы).

Оператор EXCLUDE дополняет возможности операторов MOVE и COPY и используется для задания исключения данных из операции пересылки или копирования, определенных в этих операторах в форматах с операндами DSGROUP, PDS и CATALOG. Формат:

```
[имя] EXCLUDE [DSGROUP=имя]  
[MEMBER=имя-раздела]
```

где DSGROUP=имя — указывает полное имя (при пересылке или копировании группы наборов данных все наборы данных, имена которых соответствуют указанному имени, должны быть исключены из обработки; при пересылке или копировании каталога все элементы в каталоге, имена которых соответствуют указанному имени, исключаются из обработки);

MEMBER=имя-раздела — при пересылке или копировании библиотечного набора данных указывает раздел, который должен быть исключен из обработки.

Оператор **SELECT** дополняет возможности операторов **MOVE** и **COPY** и используется для задания выборки разделов, которые должны быть пересланы или скопированы, и для выборочного переименования этих разделов. Формат:

```
[имя] SELECT MEMBER=(имя [имя]...)
MEMBER=((имя, новое-имя) [(имя, новое-имя)]...)
```

где **MEMBER**=(имя[имя]...) — указывает разделы, которые должны быть пересланы или скопированы из библиотечного набора данных, заданного в предшествующем операторе **MOVE** или **COPY** в формате с операндом **PDS**;

MEMBER=((имя, новое-имя) [(имя, новое-имя)]...) — указывает разделы, которые должны быть пересланы или скопированы, и их новые имена.

Оператор **REPLACE** дополняет возможности операторов **MOVE** и **COPY** и используется для задания замещения содержания указанного раздела библиотечного набора данных разделом из другого библиотечного набора данных. Формат:

```
[имя] REPLACE DSNAME=имя
MEMBER=имя-раздела
[FROM=тип=регистрационный]
[CVOL=тип=регистрационный]
```

где **DSNAME**=имя — указывает имя библиотечного набора данных, который содержит раздел с новым содержанием;

MEMBER=имя-раздела — указывает имя раздела. Значения остальных операндов те же, что и для операндов оператора **INCLUDE**.

Операторы **INCLUDE**, **EXCLUDE**, **SELECT**, **REPLACE** не задаются, когда набор данных разгружается или когда разгруженные данные пересылаются или копируются.

Коды возврата программы:

00 — удачное завершение задания;

04 — заданная функция программы выполнена не полностью, обработка продолжается;

08 — возникали исправимые ошибки, обработка продолжается;

12 — возникла неисправимая ошибка ввода-вывода, пункт задания завершается.

Программа IENPROGM. Предназначена для изменения системных управляющих данных и обеспечивает следующие возможности: вычеркивание набора данных или раздела набора данных, переименование набора данных или раздела набора данных, каталогизирование и исключение из каталога набора данных, создание в каталоге или исключение из каталога индекса или дополнительного имени индекса, установление и устранение логической связи между томами прямого доступа на уровне ката-

лога, построение и обслуживание индекса групп поколений данных, обработку паролей наборов данных.

Для выполнения или вызова программы необходимы операторы языка управления заданиями JOB, EXEC, DD с именами SYSPRINT, SYSIN, операторы DD в следующей записи:

```
//имяdd DD UNIT=xxxx,VOLUME=SER=xxxxxx,DISP=OLD
```

(для каждого постоянно установленного тома, относящегося к пункту задания);

```
//имяdd DD VOLUME=SER=xxxxxx,UNIT=xxxx,DISP=OLD или  
//имяdd DD VOLUME=(PRIVATE, SER=xxxxxx), UNIT=(xxxx, DEFER),  
// DISP=OLD
```

(для сменных томов, относящихся к пункту задания).

Управляющие операторы программы: SCRATCH, RENAME, CATLG, UNCATLG, BLDX, DLTG, BLDA, DLTA, CONNECT, RELEASE, BLDG, ADD, REPLACE, DELETEP, LIST.

Оператор SCRATCH предназначен для задания вычеркивания набора данных (сведения о каталогизированном наборе данных в каталоге не изменяются) или раздела библиотечного набора данных из тома прямого доступа. Формат:

```
[имя] SCRATCH {DSNAME=имя}  
                {VTOC  
                VOL=тип=список  
                [PURGE]  
                [MEMBER=имя]  
                [SYS]
```

где DSNAME=имя — указывает имя вычеркиваемого набора данных или набора данных, из которого должен быть вычеркнут раздел.

VTOC — указывает, что на данном томе должны быть вычеркнуты все наборы данных, за исключением защищенных паролями и тех, срок хранения которых не истек.

VOL=тип=список — указывает том (или тома) с набором или наборами данных, подлежащих вычеркиванию (если указан операнд VTOC, то в операнде VOL можно указать только один том);

PURGE — указывает, что каждый набор данных, определенный операндами DSNAME или VTOC, вычеркивается и в том случае, если срок хранения набора данных не истек;

MEMBER=имя — указывает имя раздела библиотечного набора данных или дополнительное имя раздела, которое должно быть удалено из справочника библиотечного набора данных;

SYS — применяется только с операндом VTOC и указывает запрос на вычеркивание только тех наборов данных, имена которых начинаются с AAAAAAAAA.AAAAAAAAA.AAAAAAAAA.AAAAAAAAA. или SYSnnnnn, T и F или V в позиции 19. Эти имена присваиваются наборам данных операционной системой.

Примечание. При использовании управляющего оператора SCRATCH не допускается одновременное выполнение программы IENPROGM и другой программы, использующей том или набор данных, указанных в управляющем операторе SCRATCH.

Оператор RENAME предназначен для задания изменения имени набора данных либо основного или дополнительного имени раздела библиотечного набора данных, расположенных на томе прямого доступа (сведения о каталогизированном наборе данных в каталоге не изменяются). Формат:

```
[имя] RENAME DSNAME=имя  
VOL=тип=список  
NEWNAME=имя  
[MEMBER=имя]
```

где DSNAME=имя — указывает имя набора данных, который должен быть переименован или содержит раздел библиотечного набора данных, подлежащий переименованию;

VOL=тип=список — указывает том (тома), который содержит набор данных или раздел библиотечного набора данных, подлежащий переименованию (при указании операнда MEMBER в операнде VOL указывается только один том);

NEWNAME=имя — указывает новое имя набора данных либо основное или дополнительное имя раздела библиотечного набора данных;

MEMBER=имя — указывает основное или дополнительное имя раздела библиотечного набора данных, который должен быть переименован.

Оператор CATLG предназначен для задания построения элемента каталога, соответствующего указанному набору данных (автоматически создаются необходимые дополнительные уровни индексов). Формат:

```
[имя] CATLG DSNAME=имя  
VOL=тип=список  
[CVOL=тип=регистрационный]
```

где DSNAME=имя — указывает имя набора данных, подлежащего каталогизированию (составное имя не должно превышать 44 символа, включая разделители);

VOL=тип=список — указывает том (тома), который содержит набор данных, подлежащий каталогизированию (при указании списка томов для последовательного или индексно-последовательного набора данных регистрационные номера томов должны записываться в том порядке, в котором они приводились в операторах DD во входном потоке при создании набора данных); указываются регистрационные номера томов для определенного типа устройства внешней памяти; для набора данных, расположенного на томе магнитной ленты, следует указывать регистрационный номер тома и порядковый номер набора данных:

VOL=тип=([регистрационный, порядковый номер], ...);

CVOL=тип=регистрационный — указывает тип НМД и регистрационный номер тома, на котором начинается поиск каталога (по умолчанию считается резидентный том операционной системы).

Оператор UNCATLG предназначен для задания удаления элемента, соответствующего указанному набору данных, из каталога. Формат:

[имя] UNCATLG DSNAME=имя
[CVOL=тип=регистрационный]

где DSNAME=имя — указывает имя набора данных, подлежащего исключению из каталога;

CVOL=тип=регистрационный — совпадает по назначению с операндом в операторе CATLG.

Оператор BLDX используется для создания нового индекса в каталоге (автоматически создаются необходимые индексы более высокого уровня). Формат:

[имя] BLDX INDEX=имя
[CVOL=тип=регистрационный]

где INDEX=имя — указывает составное имя создаваемого индекса;

CVOL=тип=регистрационный — совпадает по значению с операндом в операторе CATLG.

Оператор DLTX предназначен для задания удаления индекса из каталога (можно удалить только индекс, не имеющий элементов). Формат:

[имя] DLTX INDEX=имя
[CVOL=тип=регистрационный]

где INDEX=имя — указывает составное имя исключаемого индекса;

CVOL=тип=регистрационный — совпадает по значению с операндом в операторе CATLG.

Примечание. Так как не исключается индекс высшего уровня, то для исключения всей индексной структуры оператор DLTX нужно задать несколько раз. Например, для исключения индексной структуры A.B.C нужно задать:

DLTX INDEX=A.B.C
DLTX INDEX=A.B
DLTX INDEX=A

Оператор BLDA предназначен для задания присвоения дополнительного имени индекса высшего уровня в каталоге. Формат:

[имя] BLDA INDEX=имя
ALIAS=имя
[CVOL=тип=регистрационный]

где INDEX=имя — указывает несоставное имя индекса, которому нужно присвоить дополнительное имя (количество символов в имени не должно превышать 8);

ALIAS=имя — указывает несоставное имя, присваиваемое в качестве дополнительного имени (количество символов в имени не должно превышать 8);

CVOL=тип=регистрационный — совпадает по значению с операндом в операторе CATLG.

Оператор DLTА предназначен для задания исключения дополнительного имени, присвоенного индексу высшего уровня в каталоге. Формат:

[имя] DLTА ALIAS=имя
[CVOL=тип=регистрационный]

где ALIAS=имя — указывает несоставное дополнительное имя индекса, подлежащее исключению (количество символов в имени не должно превышать 8);

CVOL=тип=регистрационный — совпадает по значению с операндом в операторе CATLG.

Оператор CONNECT предназначен для задания установления логической связи носителей данных, содержащих части каталога. Формат:

[имя] CONNECT INDEX=имя
VOL=тип=регистрационный
[CVOL=тип=регистрационный]

где INDEX=имя — указывает несоставное имя индекса, которое должно быть включено в индекс высшего уровня первого тома (количество символов в имени не должно превышать 8);

VOL=тип=регистрационный — указывает тип НМД и регистрационный номер второго тома, информация о которых заносится в индекс высшего уровня первого тома;

CVOL=тип=регистрационный — указывает тип НМД и регистрационный номер первого тома (по умолчанию под первым томом подразумевается резидентный том операционной системы).

Оператор RELEASE предназначен для задания устранения логической связи носителей данных, содержащих части каталога. Формат:

[имя] RELEASE INDEX=имя
[CVOL=тип=регистрационный]

где INDEX=имя — указывает несоставное имя индекса, которое должно быть удалено из индекса высшего уровня первого тома (количество символов в имени не должно превышать 8);

CVOL=тип=регистрационный — совпадает по значению с операндом в операторе CONNECT.

Оператор BLDG предназначен для задания построения индекса группы поколений данных. Формат:

[имя] BLDG INDEX=имя
ENTRIES=п
[CVOL=тип=регистрационный]
[EMPTY]
[DELETE]

где INDEX=имя — указывает имя индекса группы поколений данных (имя может иметь до 35 символов);

ENTRIES=п — указывает количество элементов, содержащихся в индексе группы поколений данных (и не должно превышать 255);

CVOL=тип=регистрационный — совпадает по значению с операндом в операторе CATLG;

EMPTY — указывает, что при переполнении индекса группы поколений данных все элементы из него удаляются. Это приводит к исключению из каталога всей группы поколений данных;

DELETE — указывает, что группа поколений данных должна вычеркиваться из тома, если элементы этой группы поколений данных удаляются из индекса.

Оператор **ADD** предназначен для внесения пароля в набор данных **PASSWORD**. Формат:

```
[имя] ADD DSNNAME=имя
      [PASSWORD2=новый-пароль]
      [CPASSWORD=основной-пароль]
      [TYPE=код]
      [VOL=тип=список]
      [DATA=данные-пользователя-ЭВМ]
```

где **DSNNAME=имя** — указывает имя набора данных, которому присваивается пароль;

PASSWORD2=новый-пароль — указывает пароль, который должен быть занесен в набор данных **PASSWORD** (пароль может содержать до 8 алфавитно-цифровых символов);

CPASSWORD=основной-пароль — указывает основной пароль для набора данных;

TYPE=код — указывает вид защиты набора данных и состояние защиты, устанавливаемые в **DSCB**, если основной пароль присваивается набору данных, находящемуся на НМД; параметр «код» может принимать следующие значения:

1 — указывает, что пароль разрешает чтение и запись набора данных; если набору данных присваивается основной пароль, в **DSCB** устанавливается защита по чтению и записи;

2 — указывает, что пароль разрешает только чтение набора данных; если набору данных присваивается основной пароль, в **DSCB** устанавливается защита по чтению и записи;

3 — указывает, что пароль разрешает чтение и запись набора данных; если набору данных присваивается основной пароль, в **DSCB** устанавливается защита по записи;

VOL=тип=список — указывает тип устройства внешней памяти, на котором расположен том (тома) с набором данных, защищаемым паролем;

DATA='данные-пользователя-ЭВМ' — указывает данные пользователя ЭВМ, которые должны быть занесены в соответствующую запись набора данных **PASSWORD** (данные заключаются в апострофы и не должны превышать 77 символов).

Оператор **REPLACE** предназначен для указания замены следующих записей набора данных **PASSWORD**: пароля, вида защиты, данных пользователя ЭВМ. Формат:

```
[имя] REPLACE DSNNAME=имя
      [PASSWORD1=текущий-пароль]
      [PASSWORD2=новый-пароль]
      [CPASSWORD=основной-пароль]
      [TYPE=код]
      [VOL=тип=список]
      [DATA='данные-пользователя-ЭВМ']
```

где DSNAME=имя — указывает имя набора данных, которому присваивается пароль;

PASSWORD1=текущий пароль — указывает текущий пароль, который должен быть заменен;

PASSWORD2=новый пароль — указывает новый пароль, который приписывается набору данных;

CPASSWORD=основной пароль — указывает основной пароль для набора данных (операнд указывается, если заменяется дополнительный пароль);

TYPE=код — соответствует операнду TYPE оператора ADD;

VOL=тип=список — соответствует операнду VOL оператора ADD и не указывается, если заменяется дополнительный пароль или если состояние защиты в DSCB не надо менять;

DATA='данные-пользователя-ЭВМ' — соответствует операнду DATA оператора ADD.

Оператор DELETEP предназначен для задания вычеркивания записи пароля из набора данных PASSWORD. Формат:

```
[имя] DELETEP DSNAME=имя
                        [PASSWORD1=текущий-пароль]
                        [CPASSWORD=основной-пароль]
                        [VOL=тип=список]
```

где DSNAME=имя — указывает имя набора данных, пароль которого должен быть вычеркнут;

PASSWORD1=текущий пароль — указывает пароль, который должен быть вычеркнут;

CPASSWORD=основной пароль — указывает основной пароль для набора данных, дополнительный пароль которого должен быть вычеркнут;

VOL=тип=список — соответствует операнду VOL оператора ADD (не указывается при вычеркивании дополнительного пароля).

Оператор LIST предназначен для задания распечатки записей набора данных PASSWORD. Формат:

```
[имя] LIST DSNAME=имя
          PASSWORD1=текущий-пароль
```

где DSNAME=имя — указывает имя набора данных, защищаемого паролем, выводимым на печать;

PASSWORD1=текущий-пароль — указывает пароль из записи набора данных PASSWORD, которая должна быть выведена на печать.

Коды возврата программы:

00 — успешное завершение;

04 — синтаксическая ошибка в поле имени управляющего оператора программы или в поле PARM оператора EXEC; обработка продолжается;

- 08 — требование оператора ЭВМ игнорируется из-за неправильного управляющего оператора программы или неправильного запроса; заданная функция не выполняется;
- 12 — ошибка ввода-вывода при чтении из наборов данных, определяемых операторами DD с именами SYSPRINT, SYSIN, оглавления тома прямого доступа (VTOC) или при записи в один из этих наборов данных;
- 16 — неисправимая ошибка ввода-вывода; выполнение пункта задания прекращается.

Системные коды аварийного завершения

Коды аварийного завершения вырабатываются для указания причины, по которой произошло аварийное завершение. В данной главе рассматриваются системные коды аварийного завершения задания или пункта задания. Такие коды устанавливаются в макрокоманде ABEND и выводятся на устройство системного вывода (в набор данных SYSPRINT). Системные коды завершения выводятся в виде трехзначного шестнадцатеричного числа, которому предшествует буква S.

Кроме системных кодов завершения, могут быть пользовательские коды завершения, которые вырабатываются в обрабатываемой программе также с помощью макрокоманды ABEND. Они выводятся в виде четырехзначного десятичного числа, которому предшествует буква U. Пользовательские коды завершения планируются программистом. По этой причине они здесь не рассматриваются.

Коды аварийного завершения задания или пункта задания могут быть проверены через операционную систему на соответствие некоторым условиям с помощью параметра COND в операторах JOB или EXEC. Кроме того, их можно интерпретировать в обрабатываемой программе на языке Ассемблера путем использования макрокоманды STAE или операнда ETXR макрокоманды ATTACH. Как системный, так и пользовательский коды аварийного завершения помещаются в блок управления задачей пункта задания (TCB) в поле TCBCMP.

Ниже дается краткое описание каждого системного кода аварийного завершения. Описания расположены в порядке возрастания кода. В начале каждого описания указывается шестнадцатеричный код аварийного завершения, а затем — причины возникновения аварийного завершения. Далее для каждой причины могут быть указаны действия программиста (сокращение ДП) и действия оператора операционной системы ОС ЕС (сокращение ДО), соответствующие коду аварийного завершения. Стандартные ответные действия оператора операционной системы ОС ЕС в описании обозначаются цифрами в скобках. Описание этих стандартных действий оператора приведено в приложении 2.

001. При использовании BSAM, BISAM или BDAM — ошибка при выполнении макрокоманды CHECK. DCB не содержит адрес программы SYNAD.

При использовании QSAM — ошибка в макрокоманде GET или PUT. Поле DCBEROPT в DCB содержит признак аварийного завершения (ABE).

ДО. (1, 2, 3, 5, 9, 18).

002. При использовании BSAM или QSAM — ошибка при обработке последовательного набора данных, или при создании набора данных с прямой организацией, или при использовании QISAM в режиме загрузки при открытии индексно-последовательного набора данных:

длина передаваемой записи превышает максимальную длину 32768 байт;

длина записи превышает размер дорожки магнитного диска при планировании цепочками;

длина записи превышает установленный размер блока при формате записи, определяющем переполнение дорожки;

блок не может быть помещен в один экстенд на магнитном диске при формате записи, определяющем переполнение дорожки;

для области переполнения цилиндра указано слишком много дорожек;

слово описания записи (RDW) переменной длины содержит неверную информацию о ее длине, которая должна быть от 4 до 32768 байт;

слово описания блока (BDW) меньше 9 байт, или слово описания записи (RDW) меньше 4 байт;

при длине записи, превышающей длину дорожки, не указано переполнение дорожки;

общая длина объединенной записи, формируемой в области записей, превышает значение LRECL в DCB.

ДО. (1, 3, 5, 9, 18).

008. При создании набора данных с прямой организацией программа SYNAD пыталась вернуть управление для выполнения макрокоманды CHECK, но обнаружила, что область сохранения, адрес которой находится в регистре 13, испорчена.

ДО. (1, 3, 6, 9, 18).

00B. Ошибка операционной системы при выполнении программы инициализации планировщика (IEEVIPL) или ABEND (дамп записан в SYS1.DUMP).

ДО. (2, 10, 18, 22).

00D. Ошибка при выполнении программы инициализации планировщика (IEEVIPL). Либо ошибка операционной системы, либо ABEND.

ДО. (18, получить диагностический дамп, 22).

013. При выполнении макрокоманды OPEN были обнаружены несовместимые или непредусмотренные параметры в DCB, либо

не найден раздел с именем, указанным в операторе DD, либо нет подпараметра размещения справочника в операторе DD. Выдается сообщение IEC141I.

ДП. Выполнить действия, указанные в сообщении.

ДО. (1, 3, 6, 9, 18).

020. При использовании BDAM произошла ошибка при выполнении макрокоманды OPEN. Поле DCBMACRF в DCB не содержит А,К или I либо была сделана попытка открыть набор данных прямой организации, который содержит нулевые экстенды.

ДО. (1, 3, 6, 9, 18).

025. При использовании BDAM обнаружено ошибочное поле DCBSQND в DCB. Адрес в этом поле выходит за пределы области основной памяти, отведенной задаче (поле содержит адрес последнего блока ввода-вывода (IOB) в очереди). Аварийное завершение производится без выдачи дампа.

ДО. (1, 3, 9, 13, 18).

026. При использовании BDAM — ошибка при выполнении операции ввода-вывода, для которой затребовано монопольное управление блоком. Поле DCBXARG в DCB не указывает адреса модуля монопольного чтения или модуль монопольного чтения не отмечает состояния монопольного управления блоком набора данных.

ДО. (1, 3, 5, 9, 18).

028. Ошибка операционной системы, обнаруженная супервизором страниц.

ДО. (1, 2, 4, 5, 10, 18).

030. При использовании BISAM или QISAM — ошибка при выполнении макрокоманды OPEN. Поле DCBMACRF в DCB, соответствующее операнду MACRF макрокоманды DCB, определяет неверный режим работы.

ДО. (1, 3, 6, 9, 18).

031. При использовании QISAM при попытке передать управление программе SYNAD обнаружено, что поле DCBSYNAD содержит неверный адрес. Поля DCBEXCD1 и DCBEXCD2 содержат указания на возможные причины ошибки.

ДО. (1, 3, 5, 9, 18).

032. При использовании BISAM и QISAM — ошибка при выполнении макрокоманды OPEN. Поле DCBMACRF в DCB, соответствующее операнду MACRF макрокоманды DCB, содержит неверную информацию.

ДО. (1, 3, 6, 9, 18).

033. При выполнении макрокоманды OPEN для индексно-последовательного набора данных произошла ошибка ввода-вывода при чтении индекса наивысшего уровня или при поиске маркера конца файла в области переполнения цилиндра или независимой области переполнения. Поля DCBLIOV и DCBLPDA в DCB содержат адреса последних записей в области переполнения цилиндра и независимой области соответственно. Адрес поля

DCBMSHI в DCB содержит адрес, выходящий за пределы основной памяти, выделенной задаче, либо определяет ключ защиты памяти, отличающийся от указанного в TCB.

ДО. (1, 3, 6, 9, 18).

034. При использовании BISAM — ошибка при выполнении макрокоманды OPEN. Поля DCBSMSI и DCBMSHI в DCB определяют область основной памяти, недостаточную для размещения индекса наивысшего уровня для набора данных. Размер этой области помещается в поле DCBNCRHI при создании набора данных с помощью QISAM в режиме загрузки.

ДО. (1, 3, 6, 9, 18).

035. При использовании BISAM произошла ошибка при выполнении макрокоманды OPEN. При добавлении новых записей к набору данных поля DCBMSWA и DCBSMSW в DCB определяют область основной памяти, недостаточную для размещения одной дорожки.

ДО. (1, 3, 6, 9, 18).

036. При использовании BISAM и QISAM — ошибка при выполнении макрокоманды OPEN. Не выделено место на томе для основной области набора данных:

при использовании QISAM в режиме загрузки при создании набора данных параметр SPACE оператора DD не содержит указания количества;

обрабатывающая программа ошибочно модифицировала DSCB набора данных.

ДП. Удалить и создать вновь набор данных, используя QISAM в режиме загрузки. Для этого следует вновь запустить задание с указанием в параметре SPACE необходимого количества либо устранить ошибочную модификацию DSCB.

ДО. (1, 3, 6, 9, 18).

037. При использовании BISAM и QISAM — ошибка при выполнении макрокоманды OPEN. При построении буферов по макрокоманде BUILD либо не указан операнд BUFNO в DCB, либо операнд BUFL содержал слишком малую величину. При построении буферов по макрокоманде GETPOOL указана в операнде слишком малая длина буфера.

ДО. (1, 3, 6, 9, 18).

038. При использовании QISAM в режиме загрузки — ошибка при выполнении макрокоманды OPEN. Область индекса на томе либо слишком мала, либо превосходит размер тома.

ДП. Если создавался новый набор данных, выполнить задание заново, указав большую величину области индекса в параметре SPACE оператора DD. Если обновлялся существующий набор данных, переписать его на другое место в последовательный набор данных и создать новый набор данных с использованием QISAM в режиме загрузки, указав большую величину области индекса в параметре SPACE оператора DD.

ДО. (1, 3, 6, 9, 18).

039. При использовании QISAM для чтения набора данных обнаружен его конец, но поле DCBEODAD не определяет адрес программы обработки.

ДО. (1, 3, 6, 9, 18).

03A. При использовании BISAM и QISAM — ошибка при выполнении макрокоманды CLOSE. При записи считанного и скорректированного блока DSCB формата 2 возникла ошибка ввода-вывода.

ДП. Повторить задание заново.

ДО. (1, 3, 6, 9, 15, 18).

03B. Ошибка при выполнении макрокоманды OPEN для индексно-последовательного набора данных:

в DSCB формата 2 отмечено, что набор данных не созданы или в DSCB отмечено, что после создания DCB не был закрыт;

неверно задано поле DCBRKP в DCB; значение DCBRKP в сумме со значением DCBKEYLE не должно превосходить значение DCBLRECL. Если используются записи переменной длины, значение DCBRKP не должно быть меньше 4;

значение поля DCBKEYLE в DCB равно нулю;

не была выдана макрокоманда OPEN, подготавливающая к выполнению макрокоманды PUT;

ошибка в определении LRECL или BLKSIZE, или размер блока (DCBBLKSI) меньше длины логической записи (DCBLRECL), или длина логической записи равна нулю, или размер блока не кратен длине логической записи.

ДО. (1, 3, 6, 9, 18).

03D. При использовании BISAM, QISAM или BDAM — ошибка при выполнении макрокоманды OPEN. В макрокоманде DCB не указан операнд DSORG=IS или DSORG=ISU для индексно-последовательных наборов данных либо регистрационные номера томов подпараметра SER параметра VOLUME оператора DD не упорядочены или указаны не все. Иногда выдается сообщение IEC156I.

ДП. Исправить ошибки. Выполнить действия, указанные в сообщении.

ДО. (1, 3, 6, 9, 18).

03E. Ошибка при выполнении макрокоманды OPEN для индексно-последовательного набора данных в режиме возобновления загрузки. Нет достаточного количества памяти для добавления записей.

ДП. Запросить для набора данных большую область основной памяти либо добавить записи, используя BISAM.

ДО. (1, 3, 6, 9, 18).

040. Ошибка при выполнении макрокоманды OPEN для набора данных группы каналов связи ОТМД. Код ошибки — в регистре 0, если указан адрес в поле DCBEXLST, либо тип ошибки указывается сообщением IED008I. Пользовательская программа вы-

хода ABEND получает управление, если она указана в операнде EXLST макрокоманды DCB. Если происходит возврат управления из этой программы или если ее нет, происходит аварийное завершение.

ДП. Выполнить действия, соответствующие коду или типу ошибки.

ДО. (18).

041. Ошибка при выполнении макрокоманды OPEN для набора данных очередей сообщений ОТМД. Код ошибки — в регистре 0, если указан адрес в поле DCBEXLST, либо тип ошибки указан в сообщении IED0081. Далее то же, что для кода 040.

042. Ошибка при выполнении программы управления сообщениями со средством неавтономного тестирования ОТМД.

ДП. Повторить задание без средства неавтономного тестирования.

ДО. (1, 5, 7, 18).

043. Ошибка при выполнении макрокоманды OPEN для набора данных прикладной программы ОТМД. Содержимое регистра 0 указывает на причину ошибки. Далее то же, что для кода 040.

044. Ошибка при выполнении задачи COMWRITE ОТМД. Задача завершается. Содержимое регистра 3 указывает на причину ошибки.

ДП. Выполнить действия, соответствующие коду ошибки.

ДО. (18).

045. Ошибка при выполнении программы управления сообщениями ОТМД. Программа завершается. Содержимое регистра 15 указывает на причину ошибки.

ДП. Выполнить действия, соответствующие коду ошибки.

ДО. (18).

046. Программа управления сообщением ОТМД завершилась, но набор данных прикладной программы еще открыт. Последняя завершается с кодом 046.

ДП. Обеспечить необходимые условия для завершения программы управления сообщением.

ДО. (18).

056. При использовании графического метода доступа произошла ошибка при выполнении программы обработки сигналов внимания. В макрокомандах ANALYZ или GSERV указан адрес DCB, содержащий ссылку на DEB, который указывает на неверный UCB.

ДО. (1, 3, 5, 9, 18).

057. При использовании графического метода доступа произошла ошибка при выполнении программы обработки сигналов внимания. В макрокомандах ANALYZ или GSERV указан адрес DCB, содержащий ссылку на DEB, который указывает на UCB, относящийся к устройству, отличному от графического.

ДО. (1, 3, 5, 9, 18).

061. При выполнении макрокоманды CLOSE для графического DCB. Макрокоманда DAR выдана для блока GACB в задаче, отличной от той, в которой он был определен макрокомандой SPAR. Задача, выдавшая SPAR, аварийно завершается.

ДО. (1, 3, 5, 9, 18).

062. При выполнении программы пакета графических подпрограмм код возврата больше или равен значению нуль-переменной, указанной при вызове подпрограммы INGSP.

ДО. (1, 3, 5, 9, 18).

063. Оператор графического дисплея нажал клавишу отмены и использовал световое перо для завершения задачи.

083. При использовании программного обеспечения многомашинных комплексов завершается задача, которая выдала запрос на передачу данных через интерфейс прямого управления, но не дождалась ее окончания.

ДО. (3, 4, 13).

085. При использовании программного обеспечения многомашинных комплексов завершается задача, которая не освободила зарезервированные накопители на магнитных лентах как общее поле внешней памяти.

ДО. (3, 4, 13).

090. При выполнении макрокоманды OPEN для БТМД в UCS указано устройство, отличное от устройств телеобработки.

ДО. (1, 3, 5, 9, 18).

091. При выполнении макрокоманды OPEN для БТМД обнаружен непредусмотренный или неверный тип МПД, указанный в UCS, связанным с открываемым DCB.

ДО. (1, 3, 5, 9, 18).

092. При выполнении макрокоманды OPEN для БТМД обнаружено, что в UCS, связанным с открываемым DCB, указан неверный или непредусмотренный тип адаптера.

ДО. (1, 3, 5, 9, 18).

093. При выполнении макрокоманды OPEN для БТМД в UCS, связанным с открываемым DCB, указан неверный или непредусмотренный тип АП.

ДО. (1, 3, 5, 9, 18).

094. При выполнении макрокоманды OPEN для БТМД в UCS, связанным с открываемым DCB, указана неверная или непредусмотренная дополнительная характеристика.

ДО. (1, 3, 5, 9, 18).

095. При выполнении макрокоманды OPEN для БТМД обнаружилось, что типы каналов связи в группе не одинаковы, либо АП, подключенные к этим каналам, имеют неодинаковые характеристики.

ДО. (1, 3, 5, 9, 18).

096. При выполнении макрокоманды OPEN для БТМД обнаружилось, что DCBBFTEK определяет динамическую буферизацию, однако поля DCBBUFNO и DCBBUFL не заданы.

ДО. (1, 3, 5, 9, 18).

097. При выполнении макрокоманды OPEN для БТМД обнаружено, что справочник устройств ввода-вывода переполнен.

ДО. (1, 3, 5, 9, 18).

098. При выполнении макрокоманды OPEN для БТМД обнаружено, что указаны неправильные характеристики в макрокоманде DCB для синхронных устройств телеобработки.

ДО. (1, 3, 5, 9, 18).

0B0. Произошла постоянная ошибка ввода-вывода при чтении или записи информации планировщиком заданий в набор данных SYS1.SYSJOBQE, возможно, из-за испорченной дорожки. Задача завершается, дамп не выдается.

ДО. (2, 18).

0Cх. Произошло программное прерывание в программе, отличающейся от SVC-программы первого типа или обработчика прерываний ввода-вывода, и не указана программа обработки прерываний этого типа; х — код программного прерывания.

ДО. (5, 10, 18).

0D0. Произошла ошибка в результате прерывания из-за недоступности сегмента.

ДО. (2, 3, 18). Произвести повторный пуск операционной системы.

0D1. Произошла ошибка в результате прерывания из-за отсутствия страницы.

ДО. (2, 3, 18). Произвести повторный пуск операционной системы.

0F1. Произошло программное прерывание в обработчике прерываний ввода-вывода. Возможно, перекрыта программа метода доступа либо DCB или DEB были скорректированы после выдачи EXCP.

ДО. (5, 10, 18).

0F2. Произошло программное прерывание при выполнении команды в SVC-программе первого типа. Возможно, этой программе переданы неверные параметры.

ДО. (1, 5, 10, 18).

0F3. Произошло прерывание от схем контроля машины. Программа SER1 или ПОСК завершила аварийно пункт задания и продолжила обработку следующего пункта. На консоль выдается сообщение ECB920I. Работоспособность операционной системы сохраняется. Если выполнялась программа ПОСК, производится выдача дампа.

ДО. (2, 12, 19).

0F4. Обнаружен особый случай использования страницы в программе с закрытыми масками. Задача аварийно завершается. Возможна ошибка в SVC-программе или программе-аппендиксе.
ДО. (4, 5, 10, 18).

0F5. Произошло программное прерывание при загрузке SVC-программы 3-го или 4-го типа в транзитную область. Возможно, оказались неверными параметры в рабочей области, используемой при загрузке SVC-программ.

ДО. (5, 10, 18).

0F7. При обработке программного прерывания произошло новое программное прерывание.

ДП. Выполнить пункт задания вновь.

ДО. (1, 5, 10, 18).

101. При выполнении макрокоманды WAIT событий указано больше, чем блоков ЕСВ.

ДО. (1, 3, 6, 9, 18).

102. При выполнении макрокоманды POST обнаружен неверный адрес блока ЕСВ.

ДО. (1, 3, 5, 9, 18).

103. При нормальном завершении задачи по макрокоманде RETURN или передачей управления по регистру 14 обнаружилось, что ЕСВ уже отмечен или что в ЕСВ указан неверный адрес RB. Причиной может быть макрокоманда POST, ошибочно использующая ЕСВ, предназначенный для завершения задачи.

ДО. (1, 5, 10, 18).

104. При выполнении макрокоманды GETMAIN обнаружен недостаток памяти в области системных очередей.

ДП. При частом появлении данного кода необходимо увеличить размер области системных очередей.

ДО. (2, 18).

106. При выполнении одной из макрокоманд LINK, LOAD, XCTL или ATTACH обнаружилась ошибка при загрузке модуля в основную память. Возможно, неверные данные вразброс, либо в загрузочном модуле обнаружена запись недопустимого типа, либо в загрузочном модуле обнаружен недопустимый адрес, либо возникла постоянная ошибка ввода-вывода.

ДО. (1, 5, 10, 18, 19).

10A. При выполнении макрокоманды GETMAIN в регистровой форме обнаружен недостаток памяти в области системных очередей.

ДП. При частом появлении данного кода необходимо увеличить размер области системных очередей.

ДО. (2, 18).

10E. При выполнении макрокоманды SPIE неверно задан адрес PISA.

ДО. (1, 3, 5, 9, 10, 13, 18).

113. При выполнении макрокоманды OPEN или макрокоманды

OPEN с операндом TYPE=J произошла ошибка ввода-вывода при чтении или записи блока JFCB либо при чтении расширения этого блока; не предусмотрен код выхода. Выдается сообщение IEC142I.

ДП. Выполнить действия, указанные в сообщении.

ДО. (1, 3, 5, 9, 18).

117. При выполнении макрокоманды CLOSE с операндом TYPE=T для BSAM произошла ошибка ввода-вывода при установке тома магнитной ленты в нужную позицию либо при записи ленточной марки. Выдается сообщение IEC218I.

ДП. Выполнить действия, указанные в сообщении.

ДО. (1, 3, 5, 9, 18).

122. Оператор отменил задание с выдачей дампа.

ДО. (1, 2, 5 или 6, 10, 18).

128. При выполнении макрокоманды EXTRACT неверно указана область ответа (выходит за пределы раздела или ее адрес не кратен 4).

ДО. (1, 5, 10, 18).

12A. При выполнении макрокоманды ATTACH производится попытка передать подпул, совместно используемый с другими задачами, в монопольное использование подзадачей.

ДО. (1, 5, 10, 18).

12C. При выполнении макрокоманды SNAP неверно указан адрес TCB. Возможно, что по этому адресу нет TCB, либо он не кратен 4, либо подзадача уже завершилась, либо TCB не принадлежит подзадаче задачи, выдавшей макрокоманду.

ДО. (1, 5, 10, 18).

12D. При выполнении оверлейной программы оверлейный супервизор обнаружил, что 3-е и 4-е слова таблицы сегментов неверные.

ДО. (1, 5, 10, 18).

130. При выполнении макрокоманды DEQ в форме безусловного запроса (не указано RET=HAVE) производится попытка освободить ресурс, не запрошенный ранее макрокомандой ENQ.

ДО. (1, 5, 10, 18).

137. При обработке конца тома магнитной ленты произошла ошибка ввода-вывода при работе с метками. Выдается сообщение IEC022I.

ДП. Выполнить действия, указанные в сообщении.

ДО. (1, 3, 5, 9, 18).

138. При выполнении макрокоманды ENQ в форме безусловного запроса (не указан операнд RET) выдается повторный запрос на один и тот же ресурс.

ДО. (1, 5, 10, 18).

13E. Задача использует макрокоманду DETACH с операндом STAE=NO для подзадачи до ее завершения. Аварийно завершается подзадача с кодом 13E. Задача, выдавшая DETACH, не завершается.

ДО. (1, 10, 18).

13F. Ошибка при выполнении повторного пуска задания с контрольной точки. Выдается сообщение IHJ007I. Дамп не выдается. ДП. Выполнить действия, указанные в сообщении.

140. Произошла ошибка ввода-вывода при считывании блока JFCB с помощью макрокоманды RDJFCB. Выдается сообщение IEC154I.

ДП. Выполнить действия, указанные в сообщении.

ДО. (1, 3, 5, 9, 18).

14F. Произошла попытка выполнить макрокоманду STATUS для функций, отличающихся от STOP или START и при ненулевом ключе защиты памяти.

ДО. (1, 3, 5, 9, 18).

155. Программа SVC 85 использована пользовательской программой, что недопустимо.

ДО. (5, 10, 18).

16D. При выполнении одного из расширенных обработчиков SVC-прерываний (ESR, SVC 109, SVC 116, SVC 117) неверный код ESR был помещен в регистр 15. Возможно, что вызываемая функция не была включена в операционную систему либо код ESR передан программам, не имеющим соответствующей функции.

ДО. (1, 5, 10, 18).

16E. Не может быть выполнена функция DEBCHK для DEB блока DCB, переданного программой. Причина ошибки определяется содержимым регистра 15.

1B0. Системная программа преобразования относительных адресов в абсолютные встретила неверный TTR для набора данных SYS1.SYSJOBQE при работе планировщика заданий. Задача завершается без обращения к SYS1.SYSJOBQE.

ДП. Повторить выполнение задания. При удобном случае заново загрузить операционную систему с изменением параметров указанного набора данных. Убедиться, что размер блока для него не превышает пределов для того типа магнитных дисков, на которых располагается набор данных.

ДО. (2, 10, 18).

200. При запросе операции ввода-вывода ключ защиты блоков ECB, DCB и IOB отличается от заданного в DEB. Возможно преждевременное освобождение памяти, занятой блоками, или их переполнение, неверное использование.

ДО. (1, 3, 5, 9, 13, 18).

201. При выполнении макрокоманды WAIT неверно указан адрес ECB.

ДО. (1, 3, 6, 9, 13, 18).

202. При выполнении макрокоманды POST обнаружен неверный адрес блока RB в блоке ECB. Адрес RB помещается в блок ECP

при выполнении макрокоманды WAIT и не должен изменяться до выполнения макрокоманды POST.

ДО. (1, 3, 5, 9, 13, 18).

206. При выполнении одной из макрокоманд LINK, LOAD, XCTL или DELETE указан неверный адрес списка параметров, неверный адрес имени или элемента справочника либо неверно указана длина для элемента справочника.

ДО. (1, 4, 5, 10, 18).

207. Макрокоманда XCTL выдана программой асинхронного выхода, что недопустимо. Такая программа может быть завершена только по макрокоманде RETURN.

ДО. (1, 5, 10, 18).

20A. При попытке получения освобождения или перемещения раздела памяти для нового пункта задания обнаружено, что память еще отведена предыдущему пункту задания.

ДП. Повторить выполнение задания.

ДО. (1, 5, 10, 18).

20E. При выполнении макрокоманды SPIE указан неверный адрес PIE.

ДО. (1, 3, 5, 9, 10).

213. При выполнении макрокоманды OPEN произошла ошибка ввода-вывода, при чтении или записи DSCB или DSCB не найден. Выдается сообщение IEC143I.

ДП. Выполнить действия, указанные в сообщении.

ДО. (1, 3, 5, 9, 15, 18).

214. При выполнении макрокоманды CLOSE для набора данных на магнитной ленте произошла ошибка ввода-вывода при установке или диспозиции тома. Выдается сообщение IEC143I.

ДП. Выполнить действия, указанные в сообщении.

ДО. (1, 3, 5, 9, 18).

217. При выполнении макрокоманды CLOSE с операндом TYPE=T для BSAM произошла ошибка ввода-вывода при считывании блока JFCB. Выдается сообщение IEC219I.

ДП. Выполнить действия, указанные в сообщении.

ДО. (1, 3, 5, 9, 18).

222. Оператор отменил задание без выдачи дампа.

ДО. (1, 2, 5 или 6, 10, 18).

228. При выполнении макрокоманды EXTRACT обнаружено, что список параметров не выравнен по границе полного слова либо не располагается в разделе памяти пункта задания.

ДО. (1, 5, 10, 18).

22A. При выполнении макрокоманды ATTACH указывается номер подпула, больший 127, что допустимо только для системных задач.

ДО. (1, 5, 10, 18).

22C. При выполнении макрокоманды SNAP указан адрес TCB, который не кратен 8, либо превышает максимально допустимый

адрес основной памяти, либо расположен не в области памяти раздела пункта задания.

ДО. (1, 5, 10, 18).

22D. При выполнении оверлейной программы обнаружено, что адрес в таблице сегментов или таблице входов выходит за границы памяти раздела пункта задания.

ДО. (1, 5, 10, 18).

230. При выполнении макрокоманды DEQ неверно указана длина имени ресурса.

ДО. (1, 5, 10, 18).

237. При обработке конца тома обнаружена ошибка в процессе обработки меток. Выдается сообщение IEC023I.

ДП. Выполнить действия, указанные в сообщении.

ДО. (1, 3, 5, 9, 13, 18).

238. При выполнении макрокоманды ENQ неверно указана длина имени ресурса.

ДО. (1, 5, 10, 18).

23E. При выполнении макрокоманды DETACH:

адрес TCB подзадачи, переданный в регистре 1, превосходит максимальный адрес памяти и не находится на границе полного слова;

ключ защиты памяти, адрес которой находится в регистре 1, не совпадает с ключом защиты для задачи, выдавшей DETACH;

параметр, переданный в регистре 1, не является адресом TCB подзадачи.

Задача, выдавшая макрокоманду DETACH, аварийно завершается.

ДО. (1, 5, 10, 18).

240. При выполнении макрокоманды RDJFCB нет указателей на JFCB в списке выходов DCB. Выдается сообщение IEC155I.

ДП. Выполнить действия, указанные в сообщении.

ДО. (1, 5, 10, 18).

2F3. При обработке задания произошел сбой операционной системы; был выполнен повторный пуск операционной системы. Во время сбоя задание находилось в системной очереди заданий.

ДП. Если результаты выполнения задания не являются удовлетворительными, выполнить задание или пункт задания вновь.

2FE. При попытке выполнить развертку пункта задания возникла постоянная ошибка ввода-вывода. Пункт задания завершается аварийно.

ДО. (1, 5, 10, 18).

2FF. При запросе программы-аппендикса аварийного завершения (свертывание-развертывание) пункт задания был завершен аварийно.

ДО. (1, 5, 18).

300. При запросе операции ввода-вывода:

ключ защиты памяти DEB оказался ненулевым (MVT) или не совпадал с ключом защиты в TCB;

число экстенгов в DEB меньше числа экстенгов в поле MIOB.

ДП. Обеспечить, чтобы DCB содержал адрес DEB.

ДО. (1, 3, 5, 9, 13, 18).

301. При выполнении макрокоманды WAIT в указанном ECB уже установлен бит ожидания (две макрокоманды WAIT для одного и того же события либо неверная модификация ECB обрабатываемой программой).

ДО. (1, 3, 5, 9, 18).

305. При выполнении макрокоманды FREEMAIN не найден указанный подпул, освобождаемая область не принадлежит указанному подпулу или не описана в DQE.

ДО. (1, 5, 10, 18).

308. При выполнении макрокоманды LOAD загружаемая программа указана с помощью точки входа, определенной по макрокоманде IDENTIFY.

ДО. (1, 5, 10, 18).

30A. При выполнении макрокоманды FREEMAIN регистрового типа обнаружена ошибка:

указан параметр SP=(0), но длина в регистре 0 ненулевая;

указанный подпул не найден;

освобождаемая область памяти не принадлежит указанному подпулу;

отсутствует параметр SP, но освобождаемая область памяти не принадлежит нулевому подпулу;

освобождаемая область памяти не описывается в DQE.

ДО. (1, 5, 10, 18).

30D. При выполнении макрокоманды ABEND обнаружена ошибка. Задача была поставлена в очередь к ресурсу, который стал постоянно непригодным.

ДО. (1, 5, 10, 13, 18).

313. При выполнении макрокоманды OPEN для набора данных, расположенного на томе прямого доступа, произошла ошибка ввода-вывода при чтении DSCB формата 3. Выдается сообщение IEC144I.

ДП. Выполнить действия, указанные в сообщении.

ДО. (1, 3, 5, 9, 15, 18).

314. При выполнении макрокоманды CLOSE для набора данных, расположенного на томе прямого доступа, произошла ошибка ввода-вывода при чтении DSCB. Выдается сообщение IEC211I.

ДП. Выполнить действия, указанные в сообщении.

ДО. (1, 3, 5, 9, 15, 18).

317. При выполнении макрокоманды CLOSE с TYPE=T для BSAM для набора данных, расположенного на томе прямого доступа, произошла ошибка ввода-вывода при чтении DSCB. Выдается сообщение IEC220I.

ДП. Выполнить действия, указанные в сообщении.

ДО. (1, 3, 5, 9, 15, 18).

322. Выполнение пункта задания, каталогизированной процедуры или пункта каталогизированной процедуры превысило предельное время, установленное параметром TIME оператора EXEC или в каталогизированной процедуре системного ввода. Возможно наличие ошибок типа бесконечного цикла.

ДО. (1, 5, 17, 18).

328. При выполнении макрокоманды EXTRACT указанный адрес TCB не принадлежит подзадаче задачи, выдавшей макрокоманду.

ДО. (1, 5, 10, 18).

32A. При выполнении макрокоманды ATTACH производится попытка отдать область задания, содержащую активные программы.

ДО. (5, 21, 18).

32D. При выполнении оверлейной программы обнаружена запись неверной длины либо возникла постоянная ошибка ввода-вывода при загрузке сегмента из библиотеки.

ДО. (1, 5, 18).

330. При выполнении макрокоманды DEQ указан признак обязательного завершения задачи (RMC) не в состоянии супервизора.

ДО. (1, 5, 10, 18).

337. При обработке конца набора данных нет адреса в поле DCBEODAD. Выдается сообщение IEC024I.

ДП. Выполнить действия, указанные в сообщении.

ДО. (1, 3, 6, 9, 14, 18).

338. При выполнении макрокоманды ENQ указан признак обязательного завершения задачи (SMC) не в состоянии супервизора.

ДО. (1, 5, 10, 18).

33E. Выполняется макрокоманда DETACH с параметром STAE=YES для подзадачи, которая не завершилась. Подзадача завершается аварийно. Задача, выдавшая макрокоманду DETACH, не завершается аварийно.

ДО. (1, 5, 10, 18).

3FE. При попытке нормального завершения задания запросы ввода-вывода телеобработки были активными или находились в очереди. Эта ситуация обычно возникает, когда завершающаяся задача выдает запросы ввода-вывода на набор данных, открытый другой задачей.

ДО. (5, 9, 10, 18).

400. При запросе операции ввода-вывода поле DEBDCBAD в DEB и поле IOBDCBPT в JOB содержит разные адреса.

ДО. (1, 3, 5, 9, 13, 18).

406. При выполнении макрокоманды LINK, ATTACH или XCTL запрашиваемая программа отмечена редактором связей как

«только загружаемая» (OL) или запрашиваемая программа указана с помощью точки входа, определенной по макрокоманде IDENTIFY.

ДО. (1, 5, 10, 18).

40А. При выполнении макрокоманды FREEMAIN регистрового типа обрабатывающей программой с ключом защиты, отличным от нуля, указано освобождение нулевого подпула.

ДО. (1, 5, 10, 18).

413. При выполнении макрокоманды OPEN для набора данных на томе магнитной ленты или прямого доступа указано INPUT, INOUT или RDBACK, но не указан регистрационный номер SER в операторе DD, или произошла ошибка ввода-вывода при считывании метки тома, или том не мог быть установлен на устройство, или томов указано меньше, чем выделено периферийных устройств. Выдается сообщение IEC145I.

ДП. Выполнить действия, указанные в сообщении.

ДО. (1, 3, 5, 9, 18).

414. При выполнении макрокоманды CLOSE для набора данных на томе прямого доступа произошла ошибка ввода-вывода при записи обновленного DSCB. Выдается сообщение IEC212I.

ДП. Выполнить действия, указанные в сообщении.

ДО. (1, 3, 5, 9, 15, 18).

417. При выполнении макрокоманды CLOSE с операндом TYPE=T для набора данных на томе прямого доступа с использованием BSAM произошла ошибка ввода-вывода при считывании обновленного DSCB. Выдается сообщение IEC221I.

ДП. Выполнить действия, указанные в сообщении.

ДО. (1, 3, 5, 9, 15, 18).

422. Задание требует для инициирования слишком много памяти в наборе данных SYS1.SYSJOBQE. Размер доступной области памяти указан либо при генерации, либо в параметре t в ответе на сообщение IEF423A.

ДП. Поделить задание на два или более либо в ответе на сообщение IEF423A указать большее значение параметра.

ДО. (11, 18).

425. При выполнении макрокоманды SEGWT в оверлейной программе была указана загрузка исключаящего сегмента.

ДО. (1, 5, 18).

42А. При выполнении макрокоманды ATTACH неверно указан адрес в операнде ECB; он не кратен 4, либо превышает максимальный адрес памяти, либо ключ защиты памяти в TCB для этой задачи не совпадает с ключом защиты памяти ECB.

ДО. (1, 5, 10, 18).

430. При выполнении макрокоманды DEQ обнаружен неверный список параметров.

ДО. (1, 5, 10, 18).

437. При обработке конца тома не совпадают ключи защиты памяти в полях TCBPKF в TCB и DEBDEBIO в DEB.

ДО. (1, 3, 5, 9, 18).

438. При выполнении макрокоманды ENQ обнаружен неверный список параметров.

ДО. (1, 5, 10, 18).

4FE. При попытке нормального завершения задачи существовал по крайней мере один неудовлетворенный запрос ввода-вывода. Эта ситуация обычно возникает, когда завершающаяся задача выдает запросы ввода-вывода на набор данных, открытый другой задачей.

ДО. (5, 9, 10, 18).

500. При запросе операции ввода-вывода DEB не содержит правильного адреса UCB или поле UCBID в UCB, указанное полем DEBUCBAD в DEB, не содержит FF.

ДО. (1, 3, 5, 9, 13, 18).

504. При выполнении макрокоманды GETMAIN с операндами LA или A списки длин и адресов перекрывают друг друга.

ДО. (1, 5, 10, 18).

505. При выполнении макрокоманды FREEMAIN с операндами LA или A списки длин и адресов перекрывают друг друга.

ДО. (1, 5, 10, 18).

506. При выполнении макрокоманды LINK, XCTL, ATTACH или LOAD в оверлейной программе недостаточно памяти для вызываемой программы и оверлейного супервизора.

ДО. (1, 5, 10, 18).

513. При выполнении макрокоманды OPEN для набора данных на магнитной ленте открывается второй DCB для одного и того же тома. Выдается сообщение IEC146I.

ДП. Выполнить действия, указанные в сообщении.

ДО. (1, 3, 5, 9, 18).

514. При выполнении макрокоманды CLOSE произошла ошибка ввода-вывода при чтении JFCB. Выдается сообщение IEC231I.

ДП. Выполнить действия, указанные в сообщении.

ДО. (1, 3, 5, 9, 18).

522. Для всех задач пункта задания превышено максимальное время непрерывного состояния ожидания, которое определяется параметром JWT с СМП или равно 30 мин без СМП. Блок ECB не был отмечен макрокомандой POST.

ДП. Исправить ошибки. Если ошибок нет и время непрерывного состояния ожидания должно быть больше максимального, указать параметр TIME=1440 в операторе EXEC.

ДО. (1, 18).

52A. При выполнении макрокоманды ATTACH не хватает памяти в области очередей раздела (LSQA) для обработки параметров STAI, который был определен для порождающей задачи. Подзадача при этом не образуется, а задача, выдавшая макрокоманды ATTACH, завершается аварийно.

ДО. (1, 2, 5, 18).

530. При выполнении макрокоманды DEQ задача, выдавшая макрокоманду, не имела управление указанным ресурсом.

ДП. Исправить ошибки. По возможности необходимо избегать выдачи макрокоманды DEQ в программах выхода.

ДО. (1, 5, 10, 18).

537. При обработке конца тома магнитной ленты для сцепленных наборов данных указано устройство, используемое для другого набора данных. Выдается сообщение IEC016I.

ДП. Выполнить действия, указанные в сообщении.

538. При выполнении макрокоманды ENQ задача затребовала монопольный доступ к постоянно недоступному ресурсу.

ДО. (1, 5, 10, 13, 18).

600. Макрокоманда EXCPVR неправильно используется:

ключ защиты не равен нулю;

запрос был выдан не в режиме супервизора.

ДО. (1, 2, 10, 18).

604. При выполнении макрокоманды GETMAIN:

неверный адрес или длина указаны в FQE;

адрес в А или LA выходит за границы памяти или не кратен 4;

в регистре 1 неверный адрес списка параметров;

адрес или длина в GQE неверные.

ДО. (1, 5, 10, 18).

605. При выполнении макрокоманды FREEMAIN:

в FQE неверные адрес или длина;

адрес в А или LA выходит за границы памяти или не кратен 4;

в регистре 1 неверный адрес списка параметров;

адрес или длина в GQE неверные.

ДО. (1, 5, 10, 18).

606. При выполнении макрокоманды LINK, XCTL, ATTACH или LOAD недостаточно основной памяти для загрузки вызываемой программы.

ДО. (1, 5, 10, 18).

60A. При выполнении макрокоманды FREEMAIN или GETMAIN регистрового типа:

в FQE неверные адрес или длина;

адрес списка параметров неверный;

адрес или длина в GQE неверные.

ДО. (1, 5, 10, 18).

613. При выполнении макрокоманды OPEN для набора данных на томе магнитной ленты произошла ошибка ввода-вывода при обработке метки или установке тома в нужное положение. Выдается сообщение IEC147I.

ДП. Выполнить действия, указанные в сообщении.

ДО. (1, 3, 5, 9, 18).

614. При выполнении макрокоманды CLOSE для набора данных на томе прямого доступа произошла ошибка ввода-вывода при записи марки файла. Выдается сообщение IEC214I.

ДП. Выполнить действия, указанные в сообщении.

ДО. (1, 3, 5, 9, 18).

622. При инициализации выполнения задачи с абонентского пункта в режиме разделения времени она была завершена:

при выполнении макрокоманды GETMAIN был обнаружен недостаток памяти в разделе;

обнаружена процедура из нескольких пунктов;

при выполнении макрокоманды OBTAIN обнаружено отсутствие искомого набора данных;

задача посредника завершилась аварийно;

ошибка ввода-вывода при чтении набора данных SYS1.SYSJOBQE;

оператор выдал команду STOP для системы разделения времени;

пользователь абонентского пункта выдал сигнал ATTN после того, как процесс распределения был завершен;

пользователь абонентского пункта представил задачу для выполнения после отсоединения абонентского пункта от системы.

ДО. Обратиться к системному программисту.

62A. При выполнении макрокоманды ATTACH в системе MFT с подзадачами превышено максимальное число задач в системе.

ДО. (1, 5, 10, 18).

637. Произошла ошибка ввода-вывода при обработке конца тома магнитной ленты или конца тома во время сцепления, при записи ленточной марки, установке тома магнитной ленты в нужное положение, считывании метки, уточнении состояния для кольца защиты тома или в DCB не отмечено сцепление с несхожими характеристиками. Выдается сообщение IEC026I.

ДП. Выполнить действия, указанные в сообщении.

ДО. (1, 3, 5).

704. Выполнение макрокоманды GETMAIN спискового типа (с операндом LU или LC) в MFT, где по макрокоманде можно запрашивать только одну область памяти.

ДО. (1, 5, 10, 18).

705. Выполнение макрокоманды FREEMAIN спискового типа (с операндом L) в MFT, где по макрокоманде можно освобождать только одну область памяти.

ДО. (1, 5, 10, 18).

706. При выполнении макрокоманды LINK, XCTL, ATTACH или LOAD вызываемый модуль помечен редактором связей как невыполняемый.

ДО. (1, 5, 10, 18).

713. При выполнении макрокоманды OPEN для набора данных на магнитной ленте или томе прямого доступа при истекшем сроке хранения открытие производится для вывода, и параметр DISP оператора DD определяет значение MOD. Выдается сообщение IEC148I.

ДП. Выполнить действия, указанные в сообщении.

ДО. (1, 2, 3, 5, 9, 18).

714. При выполнении макрокоманды CLOSE для набора данных на магнитной ленте произошла ошибка ввода-вывода при обработке метки. Выдается сообщение IEC215I.

ДП. Выполнить действия, указанные в сообщении.

ДО. (1, 3, 5, 9, 18).

717. При выполнении макрокоманды CLOSE с TYPE=T для BSAM для набора данных на магнитной ленте произошла ошибка ввода-вывода при обработке метки. Выдается сообщение IEC222I.

ДП. Выполнить действия, указанные в сообщении.

ДО. (1, 3, 5, 9, 18).

722. Превышен предел выводимой информации, указанный параметром OUTLIM в операторе DD для SYSOUT.

ДО. (1, 5, 18).

72A. При выполнении макрокоманды ATTACH неверно указан адресный параметр.

ДО. (1, 5, 9, 10, 13, 18).

737. При обработке конца тома или распределения памяти прямого доступа произошла ошибка ввода-вывода или не найден DSCB для многотомного или сцепленного набора данных. Выдается сообщение IEC027I.

ДП. Выполнить действия, указанные в сообщении.

ДО. (1, 3, 5, 9, 15, 18).

800. Супервизор ввода-вывода обнаружил ошибку при попытке фиксировать страницу для запроса на ввод-вывод или при выполнении запроса на ввод-вывод обнаружено, что не все страницы, необходимые для ввода-вывода, зафиксированы.

ДО. (1, 2, 10, 18).

804. При выполнении макрокоманды GETMAIN с параметром EU или VU или при работе транслятора было запрошено больше основной памяти, чем доступно, или в MFT выдан запрос на нулевое количество памяти.

ДО. (1, 5, 10, 18).

806. При выполнении макрокоманды LINK, XCTL, ATTACH или LOAD с операндом EP или EPLOC обнаружена ошибка при выполнении программы BLDL:

запрашиваемая в операнде программа не найдена в библиотеке;

при просмотре справочника библиотеки возникла постоянная ошибка ввода-вывода.

ДО. (1, 5, 10, 16, 18).

80A. При выполнении макрокоманды GETMAIN регистрового типа либо с параметром EU или VU запрос превышает доступное количество памяти, либо в MFT запрос на нулевое количество памяти.

ДО. (1, 5, 10, 18).

813. При выполнении макрокоманды OPEN для набора данных на магнитной ленте обнаружена ошибка при обработке метки. Выдается сообщение IEC149I.

ДП. Выполнить действия, указанные в сообщении.

ДО. (1, 3, 5, 9, 18).

822. Запрос обрабатываемой программы, выполняемой в области $V=R$, не может быть удовлетворен из-за недостатка реальной памяти.

ДО. (1, 4, 5, 18).

837. При обработке конца тома для последовательного набора данных произошла ошибка ввода-вывода: не найден DSCB для многотомного или сцепленного набора данных. Выдается сообщение IEC028I.

ДП. Выполнить действия, указанные в сообщении.

ДО. (1, 3, 5, 9, 18).

900. Неверно составлена программа канала для режима SVS. Для запроса виртуальной памяти супервизор ввода-вывода не может найти логическую точку в программе канала, куда он может поместить команду TIC без изменения логики программы.

ДО. (1, 2, 10, 18).

905. При выполнении макрокоманды FREEMAIN:

адрес освобождаемой области не кратен 8;

в MFT адрес освобождаемой области не находится в области памяти пункта задания;

в MFT с подзадачами размер освобождаемой области больше размера, определяемого GQE.

ДО. (1, 5, 10, 18).

906. При выполнении макрокоманды LINK число запросов на выполнение реентерабельной или повторно используемой программы превысило 255.

ДО. (1, 5, 10, 18).

90A. При выполнении макрокоманды GETMAIN или FREEMAIN регистрового типа:

адрес освобождаемой области не кратен 8;

в MFT адрес освобождаемой области не находится в области памяти пункта задания;

в MFT с подзадачами размер освобождаемой области больше размера, определяемого GQE.

ДО. (1, 5, 10, 18).

913. При выполнении макрокоманды OPEN для набора данных, защищенного паролем, оператор дважды ввел неверный пароль (второй раз — в ответ на сообщение IEC310A). Выдается сообщение IEC150I.

ДП. Выполнить действия, указанные в сообщении.

ДО. (1, 3, 5, 9, 18).

A00. При трансляции CCW после выхода из программы-аппендикса PCI супервизором ввода-вывода обнаружено, что буфер

или область программы канала ввода-вывода не зафиксированы в реальной памяти или обнаружен неправильный элемент в списке параметров расширения.

ДО. (1, 2, 10, 18).

A03. При попытке нормального завершения по макрокоманде RETURN или передачей управления по регистру 14 у задачи обнаружены незавершенные подзадачи.

ДО. (1, 5, 10, 18).

A05. При выполнении макрокоманды FREE MAIN освобождаемая область пересекается с существующей свободной областью основной памяти.

ДО. (1, 5, 10, 18).

A06. При выполнении макрокоманды LINK, XCTL, ATTACH или LOAD обнаружен второй запрос на один и тот же повторно используемый модуль.

ДО. (1, 5, 10, 18).

A0A. При выполнении макрокоманды FREE MAIN регистрового типа обнаружено, что освобождаемая область пересекается со свободным участком.

ДО. (1, 5, 10, 18).

A13. При выполнении макрокоманды OPEN для набора данных на магнитной ленте на томе не обнаружено набора данных с номером файла, указанным в параметре LABEL оператора DD. Выдается сообщение IEC151I.

ДП. Выполнить действия, указанные в сообщении.

ДО. (1, 3, 5, 9, 18).

A14. При выполнении макрокоманды CLOSE для набора данных на томе прямого доступа произошла ошибка ввода-вывода при освобождении неиспользуемой памяти на томе прямого доступа. Выдается сообщение IEC216I.

ДП. Выполнить действия, указанные в сообщении.

ДО. (1, 3, 5, 9, 18).

A23. Задача находится в очереди ожидания ответа оператора (элемент ORE), когда произошло аварийное завершение системной задачи связи с оператором. Требуется перезагрузка операционной системы.

ДО. (2, 10, 18).

A37. При обработке конца тома выдана макрокоманда EOV, но DSB не был открыт. Выдается сообщение IEC015I.

ДП. Выполнить действия, указанные в сообщении.

ДО. (1, 4, 5, 9, 10, 18).

B00. Степень загрузки операционной системы такова, что потребность в фиксировании страниц превышает границу, установленную супервизором страниц.

ДО. (1, 2, 10, 18).

B04. При выполнении макрокоманды GET MAIN был указан номер подпула, превышающий 127.

ДО. (1, 5, 10, 18).

B05. При выполнении макрокоманды FREEMAIN был указан номер подпула, превышающий 127.

ДО. (1, 5, 10, 18).

B06. В результате ошибки ввода-вывода в системной задаче, вызванной задачей пользователя, системная задача восстановлена, задача пользователя аварийно завершена.

ДО. (5, 10, 18).

B0A. При выполнении макрокоманды GETMAIN или FREEMAIN регистрового типа был указан номер подпула, превышающий 127.

ДО. (1, 5, 10, 18).

B13. При выполнении макрокоманды OPEN для набора данных на печатающем устройстве с UCS произошла ошибка. Выдается сообщение IEC152I.

ДП. Выполнить действия, указанные в сообщении.

ДО. (1, 3, 5, 9, 18).

B14. При выполнении макрокоманды CLOSE для библиотечного набора данных программа STOW не может занести, модифицировать или удалить данные из справочника библиотеки, так как имя уже находится в нем, либо справочник уже заполнен, либо возникла ошибка ввода-вывода при просмотре справочника. Выдается сообщение IEC217I.

ДП. Выполнить действия, указанные в сообщении.

ДО. (1, 3, 5, 9, 18).

B23. После выдачи макрокоманды WTOR, ответ на которую еще не был получен, произошло аварийное завершение системной задачи связи. Требуется перезагрузка операционной системы.

ДО. (2, 10, 18).

B37. При выполнении обработки конца тома том должен быть снят с устройства, но система не может этого сделать. Выдается сообщение IEC030I.

ДО. (1, 3, 5, 9, 18).

C00. Программа-аппендикс фиксирования страниц указала более семи элементов в своем списке фиксирования страниц.

ДО. (1, 2, 10, 18).

C03. При попытке нормального завершения задачи по макрокоманде RETURN или передачей управления по регистру 14 невозможно закрыть один из открытых наборов данных, так как DCB был испорчен (поле TCBDEB в TCB указывает на DEB, связанный с этим DCB):

программа, содержащая DCB, выдала макрокоманду XCTL;

программа освободила память, связанную с DCB;

DCB был неверно модифицирован.

ДО. (1, 5, 10, 18).

C06. Задача загрузки транзитной области, вызванная задачей пользователя, находилась под угрозой аварийного завершения.

Система завершила аварийно задачу пользователя. Задание необходимо выполнить вновь.

ДО. (10, 18).

C13. При выполнении макрокоманды OPEN, либо макрокоманды OPEN с TYPE=J, либо макрокоманды OPEN для графического набора данных произошла ошибка ввода-вывода при чтении JFCB или DSCB для сцепленного набора данных. Либо JFCB или DSCB не найдены для сцепленного набора данных, либо графический DCB уже открыт другой задачей. Выдается сообщение IEC153I или IEC157I.

ДП. Выполнить действия, указанные в сообщении.

ДО. (1, 4, 5, 10, 18).

C2D. При выполнении оверлейной программы обнаружена неверная запись вразброс при попытке загрузить сегмент.

ДО. (1, 5).

D00. При выполнении запроса на ввод-вывод было произведено обращение к странице, ссылка к которой не предусмотрена этим запросом.

ДО. (1, 2, 10, 18).

D03. При попытке нормального завершения задачи по макрокоманде RETURN или передачей управления по регистру 14 выяснилось, что ресурс, запрошенный ранее макрокомандой ENQ, не был освобожден к моменту завершения.

ДО. (1, 5, 10, 18).

D05. При выполнении макрокоманды FREEMAIN по запросу какой-либо системной программы производится попытка освободить память в области системных очередей, не принадлежащую задаче.

ДО. (1, 5, 10, 18).

D0A. При выполнении макрокоманды FREEMAIN регистрового типа по запросу какой-либо системной программы производится попытка освободить память в области системных очередей, не принадлежащую задаче.

ДО. (1, 5, 10, 18).

D0D. Возникла недопустимая рекурсия программы ABEND при аварийном завершении подзадачи. Пункт задания завершен аварийно.

ДО. (18).

D13. При выполнении макрокоманды OPEN для графического устройства указан DCB для устройства, не являющегося графическим. Выдается сообщение IEC158I.

ДП. Выполнить действия, указанные в сообщении.

ДО. (1, 3, 4, 5, 9, 10, 18).

D14. При выполнении макрокоманды CLOSE для графического устройства закрываемый DCB был открыт другой задачей.

ДО. (1, 3, 5, 9, 18).

D23. При выполнении макрокоманды WTO или WTOR:

- список параметров не начинается с границы полуслова;
указана длина текста меньше или равная нулю;
нет свободных буферов.

ДО. (1, 5, 10, 18).

D2D. При загрузке сегмента оверлейной программы обнаружена запись недопустимого типа.

ДО. (1, 5, 18).

D37. При запросе операции вывода для набора данных на томе прямого доступа первичное количество памяти исчерпано, а вторичное не указано в параметре SPACE оператора DD.

E03. При попытке нормального завершения задачи по макрокоманде RETURN или передачей управления по регистру 14 выяснилось, что задача выполнялась в состоянии обязательного завершения действия и не сбросила это состояние перед завершением. Задача завершается аварийно.

ДО. (1, 4, 5, 10, 13, 18).

E04. При выполнении макрокоманды GETMAIN в состоянии супервизора в области системных очередей было запрошено памяти больше, чем доступно.

ДО. (2, 18).

E0D. В режимах MVT и SVS при аварийном завершении одной из подзадач для программы ABEND не оказалось памяти в подпуле 252. Поэтому память была выделена из области пользователя, в результате чего весь пункт задания был завершен аварийно.

ДО. (18).

E13. При выполнении макрокоманды OPEN для графического устройства значение в поле DCBGNCP не находится в диапазоне 1—99. Выдается сообщение IEC159I.

ДП. Выполнить действия, указанные в сообщении.

ДО. (1, 3, 4, 5, 9, 10, 18).

E23. При выполнении макрокоманды WTOR были неверно указаны адреса блоков ECB или RB либо области основной памяти для ответа.

ДО. (1, 5, 10, 18).

E2D. При загрузке сегмента оверлейной программы обнаружен недопустимый адрес.

ДО. (1, 5, 18).

E37. При запросе операции вывода для набора данных на томе прямого доступа или магнитной ленте требуется дополнительная память. При этом исчерпаны все тома в параметре SER или REF или счетчика томов оператора DD, либо память недоступна в библиотечном наборе данных, либо требуется 17-й экстенст. Выдается сообщение IEC032I.

ДП. Выполнить действия, указанные в сообщении.

ДО. (1, 3, 6, 9, 18).

Fnn. Команда SVC содержит неверный операнд.

ДО. (5, 10, 18).

Коды состояния ожидания

Коды состояния ожидания находятся в слове состояния программы. Этот код определяет причины перехода в состояние ожидания. Он предназначен для оператора ЭВМ, который считывает его визуально через пульт ЭВМ. Код состояния ожидания занимает последние 12 бит слова состояния программы (PSW).

Если причиной перехода в состояние ожидания является постоянная ошибка ввода-вывода, то в регистре 10 содержится адрес периферийного устройства, обращение к которому вызвало ошибку.

Ниже дается краткое описание каждого кода состояния ожидания. Описания расположены в порядке возрастания кода. В начале каждого описания указываются шестнадцатеричный код состояния ожидания и программа, вызвавшая это состояние: программа начальной загрузки (IPL), программа инициализации ядра (NIP), супервизор, супервизор ввода-вывода, программа обработки прерываний от схем контроля (ПОСК), программы регистрации состояния операционной системы (SER0 и SER1). Затем указываются причины возникновения состояния ожидания. Далее указываются рекомендуемые действия оператора ЭВМ (сокращенно ДО). Стандартные действия оператора, обозначенные номерами, описаны в приложении 2.

Возможно возникновение состояния ожидания, не имеющее кода. Причины такого состояния и действия оператора рассматриваются в руководстве оператора ОС ЕС «Коды завершения и состояния ожидания».

001. IPL, NIP. При выполнении команды TIO, выданной для определения текущего состояния устройства, с которого выполняется загрузка операционной системы, установлен код условия **ВЫКЛЮЧЕНО**. При выполнении NIP четвертый байт PSW содержит FF

ДО. Привести устройство в готовность, (22).

002. IPL, NIP. Не была инициирована операция ввода-вывода, CSW сохранено и канал не занят. При выполнении NIP четвертый байт PSW содержит FF.

ДО. (22).

003. IPL, NIP. Не была инициирована операция ввода-вывода, CSW не сохранено и канал не занят, т. е. не был найден UCB

для устройства, с которого происходит загрузка, либо в MVT и SVS обнаружен недостаток памяти, либо SIO закончилась с ненулевым признаком. При выполнении NIP четвертый байт PSW содержит FF.

ДО. (22).

004. IPL, NIP. Не была инициирована операция ввода-вывода, CSW не сохранено и канал не занят сразу же после выполнения команды TIO. При выполнении NIP четвертый байт PSW содержит FF.

ДО. (22).

005. IPL, NIP. Прерывание ввода-вывода из-за ошибки устройства. При выполнении IPL четвертый байт PSW содержит 00. При выполнении NIP четвертый байт PSW содержит FF.

ДО. (22). При повторении ситуации (8, 11, 18).

006. IPL, NIP. Произошла ошибка в интерфейсе ввода-вывода, либо в канале (в управлении или в данных), либо в программе, либо в цепочке канала. При выполнении NIP четвертый байт PSW содержит FF.

ДО. (22). При повторении ситуации (8, 11, 18).

007. NIP. Консоль оказалась недоступной.

ДО. Обеспечить доступность консоли, (22). При повторении ситуации (8, 11, 18).

008. NIP. Ошибка периферийного устройства, либо том прямого доступа с неправильным форматом установлен на системное устройство, либо метка тома располагается на альтернативной дорожке (дефектная нулевая дорожка).

ДО. Обеспечить, чтобы не было томов с неправильным форматом на системном устройстве, (22). При повторении ситуации (8, 11, 18).

009. NIP. Ошибка ввода-вывода из-за того, что в байте уточненного состояния установлен бит «ЗАЩИТА НАКОПИТЕЛЯ».

ДО. (22). При повторении ситуации (8, 11, 18).

00A. NIP. Отсутствует в каталоге SYS1.LINKLIB.

ДО. (18).

00F. IPL. Том магнитных дисков, с которого производится загрузка, не содержит записи IPL.

ДО. Убедиться, что установлен нужный том, (22). При повторении ситуации (8, 18).

010. NIP. Ошибка периферийного устройства, означающая обнаружение конца цилиндра до того, как требуемая запись была найдена.

ДО. (22). При повторении ситуации (8, 11, 18).

011. NIP. Ошибка периферийного устройства по контролю состояния дорожки.

ДО. (22). При повторении ситуации (8, 11, 18).

016. NIP. Ошибка в области постоянно распределенной памяти.

ДО. (19).

017. IPL, NIP. Ошибка периферийного устройства во время уточнения состояния. В случае NIP четвертый байт PSW содержит FF.

ДО. (22). При повторении ситуации (8, 11, 18).

018. IPL. Размер ядра слишком велик для данной вычислительной установки; не хватает памяти для записей RLD.

ДО. (18).

019. IPL. Из-за ошибки стала непригодной память, содержащая IPL.

ДО. (22). При повторении ситуации (19).

021. NIP. Прерывание ввода-вывода на графическом или телекоммуникационном устройстве, используемом в качестве консоли. Регистр 1 содержит адрес IOB для EXCP.

028. Выдается сообщение IEA047A.

ДО. Выполнить действия, указанные в сообщении.

036. NIP в SVS. Обращение к супервизору страниц до завершения его инициализации.

ДО. (22). При повторении ситуации (18).

038. NIP в SVS. Не хватает основной памяти для инициализации SYS1.PAGE.

ДО. (22). При повторении ситуации (18).

03A. NIP в SVS. Требуется новая страница виртуальной памяти, но в наборе данных SYS1.PAGE нет места.

ДО. Переопределить набор данных SYS1.PAGE, обеспечив ему больший размер (22).

03F. NIP в SVS. Сформирован неправильный список параметров для программы IEAVNPA4 при инициализации SYS1.PAGE.

ДО. (22). При повторении ситуации (18).

0E2. Супервизор. Машинная ошибка или ошибка в канале при отсутствии программ SER0 или SER1.

ДО. Выполнить программу SEREP, (22, 2, 20, 19).

101. Супервизор SVS. При выполнении макрокоманды GETMAIN обнаружено отсутствие памяти в разделе задачи за пределами области V=R.

ДО. (22). При повторении ситуации (18).

102. Супервизор SVS. При выполнении макрокоманды GETMAIN обнаружено отсутствие памяти в разделе задачи в области V=R.

ДО. (22). При повторении ситуации (18).

A01. ПОСК. Во время работы ПОСК произошло вторичное прерывание от схем контроля машины, либо первичное прерывание от схем контроля машины с признаком «Повреждение системы», либо при прерывании от схем контроля аппаратные средства не смогли записать состояние управляющих регистров или PSW. Выдается звуковой сигнал. Код соответствует сообщению ECB910W.

ДО. Если сообщение не выдается или в нем указано RUN SEREP, (2, 20, 19, 22), иначе — (2, 12, 18, 22).

A02. ПОСК. При обработке прерываний от схем контроля машины произошло еще два таких прерывания.

ДО. (2, 4, 19).

A03. ПОСК. Программное прерывание. Выдается звуковой сигнал. Код соответствует сообщению ECB910W.

ДО. То же, что для кода A01.

A04. ПОСК. Ошибка ввода-вывода. Выдается звуковой сигнал. Код соответствует сообщению ECB930W.

ДО. То же, что для кода A01.

A05. ПОСК. Ошибка основной памяти, используемой супервизором. Выдается звуковой сигнал. Код соответствует сообщению ECB900W.

ДО. То же, что для кода A01.

A09. ПОСК. Ошибка канала, которую не может исправить программа обработки ошибок канала. Запись состояния в SYS1.LOGREC не производится. Выдается звуковой сигнал. Код соответствует сообщению ECB930W.

ДО. (2, 20, 19).

A0A. ПОСК. Ошибка канала, которую не может исправить программа обработки ошибок канала. Запись состояния в SYS1.LOGREC произведена. Выдается звуковой сигнал. Код соответствует сообщению ECB930W.

ДО. (2, 12, 18, 22).

A11. ПОСК. Обнаружено, что код прерываний от схем контроля ошибочен. Выдается звуковой сигнал. Код соответствует сообщению ECB910W.

ДО. То же, что для кода A01.

A16. ПОСК. Обнаружена неисправность часов, таймера центрального процессора или компаратора. Выдается звуковой сигнал. Код соответствует сообщению ECB950W.

ДО. То же, что для кода A01.

A17. ПОСК. Обнаружена неисправность таймера. Выдается звуковой сигнал. Код соответствует сообщению ECB950W.

ДО. То же, что для кода A01.

D01. Супервизор. Повторное обращение к программе ABTERM при выполнении аварийного завершения либо обращение к программе ABTERM из-за того, что программа, выдавшая макрокоманду WAIT, аварийно завершилась.

ДО. (22, 8, 18).

E02. Супервизор ввода-вывода. Постоянная ошибка ввода-вывода при использовании графического дисплея в качестве первичной консоли, когда не была доступна дополнительная консоль. Байты состояния и уточненного состояния в регистре 15.

ДО. Выполнить (22) с помощью дополнительной консоли. Если она не исправна, (19).

E04. Супервизор. При выполнении макрокоманды GETMAIN обнаружено отсутствие памяти в области системных очередей. Задача не могла быть завершена аварийно.

ДО. (8, 18). Если код появляется часто, расширить область системных очередей.

F05. SER0 или SER1. Произведена запись состояния в SYS1.LOGREC.

ДО. (22, 2, 24, 19).

F06. SER0 или SER1. При выполнении записи о состоянии в SYS1.LOGREC произошла вторая машинная ошибка. Запись была произведена.

ДО. Выполнить программу EREP, (22, 2, 21, 12, 19).

F07. SER0 или SER1. Ошибки происходят так часто, что запись о состоянии в SYS1.LOGREC не может быть выполнена.

ДО. Выполнить программу SEREP, (22, 2, 21, 19).

F08. SER0. Не выполнена запись о состоянии в SYS1.LOGREC из-за постоянных ошибок ввода-вывода или ошибок канала, связанных с системным резидентным устройством.

ДО. Выполнить программу SEREP, (22, 2, 21, 19).

F09. SER0 и SER1. Не выполнена запись о состоянии в SYS1.LOGREC из-за того, что он заполнен, либо из-за ошибки в начальной записи.

ДО. Выполнить программу SEREP, (22), выполнить программу EREP для распечатки SYS1.LOGREC (21, 19).

F0A. SER0. Не выполнена загрузка нерезидентного модуля из SYS1.LINKLIB.

ДО. Выполнить программу SEREP, (22, 2, 21, 19).

F0D. SER0 или SER1. При выполнении записи о состоянии SYS1.LOGREC возникла постоянная ошибка ввода-вывода или центрального процессора. Запись не была произведена.

ДО. Выполнить программу SEREP, (22, 2, 21, 19).

Приложения

Приложение 1.

Перечень эксплуатационной документации на операционную систему ОС ЕС изданий 4.1 и 6.1

Издание 4.1		
Наименование документа	Обозначение документа	Примечание
<i>Общесистемные эксплуатационные документы</i>		
1. Ведомость эксплуатационных документов	XXX.804.004.ЭД	
2. Введение в операционную систему ОС ЕС. Общее описание	XXX.804.004 Д	В издании 6.1 документы XXX.804.004 Д и XXX.804.004 Д1 объединены в документ XXX.804.006 Д1
3. Концепции и возможности. Общее описание	XXX.804.004 Д1	
4. Состояние системы. Руководство системного программиста	XXX.804.004 Д23	
5. Генерация. Руководство системного программиста	XXX.804.004.Д33	
6. Макрокоманды генерации. Руководство системного программиста.	XXX.804.004 Д34	
7. Оценки памяти. Руководство системного программиста	XXX.804.004 Д24	
8. Автоматическая инициализация наборов данных генерируемой ОС. Руководство системного программиста	XXX.804.004 Д80	В издании 6.1 документ включен в документ XXX.804.006 Д33
9. Язык управления заданиями. Описание языка	XXX.804.001-01 Д2	
10. Использование символов кириллицы. Руководство программиста	XXX.804.002 Д65	
11. Супервизор. Руководство программиста	XXX.804.002 Д3	
12. Управление данными. Руководство программиста	XXX.804.002 Д4	
13. Макрокоманды супервизора и управления данными. Руководство программиста	XXX.804.002 Д5	В издании 6.1 документ разделен на документы XXX.804.006 Д5 XXX.804.006 Д102
14. Дополнительные возможности. Руководство системного программиста	XXX.804.002 Д35	

Наименование документа	Обозначение документа	Примечание
15. Отладка. Руководство системного программиста	XXX.804.002 Д36	
16. Однопрограммный режим. Руководство программиста	XXX.804.002 Д8	В издании 6.1 документ аннулирован
17. Мультипрограммный режим для фиксированного числа задач. Руководство программиста	XXX.804.002 Д9	
18. Мультипрограммный режим с переменным числом задач. Руководство программиста	XXX.804.002 Д67	
19. Процедуры обслуживания. Руководство оператора	XXX.804.002 Д25	
20. Однопрограммный режим. Руководство оператора	XXX.804.001-01 Д26	В издании 6.1 документ аннулирован
21. Мультипрограммный режим для фиксированного числа задач. Руководство оператора	XXX.804.001-01 Д27	В издании 6.1 документы XXX.804.001-01 Д27
22. Мультипрограммный режим для переменного числа задач. Руководство оператора	XXX.804.002 Д63	XXX.804.001-01 Д63 переработаны и включены в документ XXX.804.006 Д101
23. Дисплей-консоль. Руководство оператора	XXX.804.004 Д83	
24. Управление консолями. Руководство оператора	XXX.804.004 Д84	
25. Коды завершения и состояния ожидания. Руководство оператора	XXX.804.002 Д28	
26. Сообщения супервизора и управления данными. Руководство оператора	XXX.804.002 Д29	
27. Сообщения главного планировщика. Руководство оператора	XXX.804.001-01 Д30	
28. Сообщения планировщика заданий. Руководство оператора	XXX.804.001-01 Д31	
29. Управляющие блоки системы. Введение. Руководство системного программиста	XXX.804.001-01 Д37	В издании 6.1 документ включен в документ XXX.804.006 Д51
30. Управляющие блоки системы. Управление задачами, заданиями и данными. Руководство системного программиста	XXX.804.004 Д51	
31. Управляющие блоки системы. Справочник библиотеки и разметка томов. Руководство системного программиста	XXX.804.001-01 Д52	
32. Метки на магнитной ленте. Руководство программиста	XXX.804.001-01 Д7	

Наименование документа	Обозначение документа	Примечание
33. Загрузка и проверка системы. Руководство системного программиста	XXX.804.001-01 Д22	Программы IBCDASDI и IBCDMPRS описаны в этом документе
34. Утилиты. Введение. Руководство программиста	XXX.804.002 Д12	В издании 6.1 документы
35. Утилиты. Обслуживание системных управляющих данных и разметка магнитных лент. Руководство программиста	XXX.804.002 Д13	XXX.804.002 Д12 XXX.804.002 Д13 XXX.804.002 Д14 XXX.804.002 Д15
36. Утилиты. Копирование наборов данных. Руководство программиста	XXX.804.002 Д14	XXX.804.001-01 Д54 XXX.804.001-01 Д55
37. Утилиты. Преобразование, вывод на печать и перфорация наборов данных. Руководство программиста	XXX.804.002 Д15	переработаны и объединены в документы XXX.804.005 Д94 XXX.804.005 Д95 XXX.804.005 Д98
38. Утилиты. Операции с наборами данных. Руководство программиста	XXX.804.001-01 Д54	
39. Утилиты. Программа разметки томов прямого доступа. Руководство программиста	XXX.804.001-01 Д55	
40. Программы обслуживания системного программиста. Руководство системного программиста	XXX.804.004 Д68	

Программное обеспечение телеобработки данных

41. Базисный телекоммуникационный метод доступа. Руководство программиста	XXX.804.003 Д58
42. Общий телекоммуникационный метод доступа. Общее описание	XXX.804.004 Д59
43. Общий телекоммуникационный метод доступа. Руководство программиста	XXX.804.004 Д60
44. Общий телекоммуникационный метод доступа. Руководство программиста	XXX.804.004 Д61
45. Общий телекоммуникационный метод доступа. Руководство программиста	XXX.804.004 Д62
46. Сообщения общего телекоммуникационного метода доступа. Руководство оператора	XXX.804.004 Д91

Программное обеспечение диалогового удаленного ввода заданий

47. Диалоговый удаленный ввод заданий. Общее описание	XXX.804.004 Д69
-------------------------------------------------------	-----------------

Наименование документа	Обозначение документа	Примечание
48. Диалоговый удаленный ввод заданий. Руководство программиста	XXX.804.004 Д70	
49. Диалоговый удаленный ввод заданий. Руководство оператора	XXX.804.004 Д92	
<i>Программное обеспечение машинной графики</i>		
50. Средства программирования для комплекса алфавитно-цифровых дисплеев ЕС-7906. Руководство программиста	XXX.804.003 Д56	
51. Средства программирования графического дисплея ЕС-7064. Руководство программиста	XXX.804.003 Д57	
52. Пакет графических подпрограмм для языков Фортран, Кобол и ПЛ/1. Руководство программиста	XXX.804.003 Д64	
53. Фортран IV. Базисные программы для графопостроителей. Руководство программиста	XXX.804.004 Д87	
54. Фортран IV. Функциональные программы для графопостроителей. Руководство программиста	XXX.804.004 Д88	
<i>Программные средства восстановления</i>		
55. Средства восстановления системы. Руководство системного программиста	XXX.804.004 Д21	
56. Контрольная точка. Руководство программиста	XXX.804.001-01 Д6	
57. Независимая программа редактирования и печати информации об ошибках SEREP. Руководство оператора	XXX.804.004 Д73	
58. Программа редактирования и печати записей об ошибках EREP. Руководство оператора	XXX.804.004 Д74	В издании 6.1 документы XXX.804.004 Д74 XXX.804.002 Д71
59. Программа инициализации набора данных SYS1.LOGREC IFCDIP00. Руководство оператора	XXX.804.002 Д71	переработаны и объединены в документ XXX.804.006 Д74
60. Сообщения средств восстановления и контрольной точки. Руководство оператора	XXX.804.004 Д32	

Наименование документа	Обозначение документа	Примечание
61. Средства восстановления системы ввода-вывода. Руководство системного программиста	XXX.804.004 Д66	
62. Записи программной регистрации ошибок устройств ввода-вывода. Руководство по техническому обслуживанию	XXX.804.004 Д78	
63. Программа выполнения неавтономных тестов. Руководство системного программиста	XXX.804.002 Д50	
64. Записи программной регистрации ошибок ЭВМ ЕС-1020. Руководство по техническому обслуживанию	XXX.804.004 Д75	В издании 6.1 документ аннулирован
65. Записи программной регистрации ошибок ЭВМ ЕС-1022. Руководство по техническому обслуживанию	XXX.804.004 Д85	
66. Записи программной регистрации ошибок ЭВМ ЕС-1030. Руководство по техническому обслуживанию	XXX.804.004 Д76	
67. Записи программной регистрации ошибок ЭВМ ЕС-1050. Руководство по техническому обслуживанию	XXX.804.004 Д77	
<i>Программные средства учета и измерений</i>		
68. Системная мониторинговая программа. Руководство системного программиста	XXX.804.002 Д72	
<i>Программные средства сортировки и объединения данных</i>		
69. Сортировка - объединение. Руководство программиста	XXX.804.001-01 Д38	
<i>Система программирования Ассемблер</i>		
70. Ассемблер. Описание языка	XXX.804.004 Д16	
71. Ассемблер. Руководство программиста	XXX.804.001-01 Д17	
<i>Алгол-60</i>		
72. Алгол-60. Описание языка	XXX.804.001-01 Д41	
73. Алгол-60. Руководство программиста	XXX.804.001-01 Д42	
<i>Кобол</i>		
74. Кобол. Описание языка	XXX.804.002 Д48	
75. Кобол. Руководство программиста	XXX.804.002 Д49	
<i>ПЛ/1</i>		
76. ПЛ/1. Описание языка	XXX.804.002 Д45	Часть 1

Наименование документа	Обозначение документа	Примечание
77. ПЛ/1. Описание языка	XXX.804.002 Д53	Часть 2
78. ПЛ/1. Руководство программиста	XXX.804.002 Д46	
79. ПЛ/1. Библиотека подпрограмм. Руководство программиста	XXX.804.002 Д47	
80. ПЛ/1. Справочник. Руководство программиста	XXX.804.002 Д81	
81. ПЛ/1. Сообщения компилятора. Руководство программиста	XXX.804.002 Д82	
<i>РПГ</i>		
82. РПГ. Описание языка	XXX.804.001-01 Д39	
83. РПГ. Руководство программиста	XXX.804.001-01 Д40	
<i>Фортран</i>		
84. Фортран IV. Описание языка	XXX.804.001-01 Д18	
85. Фортран IV. Руководство программиста	XXX.804.002 Д19	
86. Фортран IV. Библиотека подпрограмм. Руководство программиста	XXX.804.001-01 Д20	
<i>Редактор связи. Загрузчик. Тестран</i>		
87. Редактор связей. Руководство программиста	XXX.804.002 Д10	
88. Загрузчик. Руководство программиста	XXX.804.001-01 Д43	
89. Тестран. Руководство программиста	XXX.804.001-01 Д11	В издании 6.1 аннулирован

Издание 6.1

Общесистемные эксплуатационные документы

1. Ведомость эксплуатационных документов XXX.804.006 ЭД
2. Введение в операционную систему ОС ЕС. Общее описание XXX.804.006 Д1
3. Состояние операционной системы ОС ЕС. Руководство системного программиста XXX.804.006 Д23
4. Генерация. Руководство системного программиста XXX.804.006 Д33
5. Макрокоманды генерации. Руководство системного программиста XXX.804.006 Д34
6. Оценки памяти. Руководство системного программиста XXX.804.006 Д24
7. Язык управления заданиями. Описание языка XXX.804.006 Д2
8. Использование символов кириллицы. Руководство программиста XXX.804.002 Д65
9. Супервизор. Руководство программиста XXX.804.006 Д3

Наименование документа	Обозначение документа	Примечание
10. Управление данными. Руководство программиста	XXX.804.005 Д4	
11. Макрокоманды супервизора. Руководство программиста	XXX.804.006 Д5	
12. Макрокоманды управления данными. Руководство программиста	XXX.804.006 Д102	
13. Дополнительные возможности супервизора и управления заданиями. Руководство системного программиста	XXX.804.006 Д35	
14. Дополнительные возможности управления данными. Руководство системного программиста	XXX.804.006 Д103	
15. Отладка. Руководство системного программиста	XXX.804.006 Д36	
16. Монитор динамической отладки. Руководство программиста	XXX.804.006 Д106	
17. Мультипрограммный режим для фиксированного числа задач. Руководство программиста	XXX.804.006 Д9	
18. Мультипрограммный режим для переменного числа задач. Руководство программиста	XXX.804.006 Д67	
19. Режим мультипрограммирования с переменным числом задач, совместно использующих виртуальную память. Руководство программиста	XXX.804.006 Д100	
20. Процедуры обслуживания. Руководство оператора	XXX.804.006 Д25	
21. Управление работой операционной системы ОС ЕС. Руководство оператора	XXX.804.006 Д101	
22. Дисплей-консоль. Руководство оператора	XXX.804.006 Д83	
23. Управление консолями. Руководство оператора	XXX.804.006 Д84	
24. Коды завершения и состояния ожидания. Руководство оператора	XXX.804.006 Д28	
25. Сообщения супервизора и управления данными. Руководство оператора	XXX.804.006 Д29	
26. Сообщения главного планировщика. Руководство оператора	XXX.804.006 Д30	
27. Сообщения планировщика заданий. Руководство оператора	XXX.804.006 Д31	

Наименование документа	Обозначение документа	Примечание
28. Управляющие блоки. Управление задачами, заданиями и данными. Руководство системного программиста	XXX.804.006 Д51	
29. Управляющие блоки. Справочник библиотеки и разметка томов. Руководство системного программиста	XXX.804.006 Д52	
30. Метки томов магнитных лент. Руководство программиста	XXX.804.006 Д7	
31. Загрузка и проверка операционной системы ОС ЕС. Руководство системного программиста	XXX.804.006 Д22	
32. Автономные и системные программы обслуживания. Руководство программиста	XXX.804.005 Д94	
33. Программы обслуживания наборов данных. Руководство программиста	XXX.804.005 Д95	
34. Сообщения программ обслуживания. Руководство программиста	XXX.804.005 Д98	
35. Программы обслуживания системного программиста. Руководство системного программиста	XXX.804.006 Д68	
<i>Программное обеспечение телеобработки данных</i>		
36. Базисный телекоммуникационный метод доступа. Руководство программиста	XXX.804.006 Д58	
37. Базисный телекоммуникационный метод доступа. Комплексы ЕС-7920. Руководство программиста	XXX.180.000 Д1	
38. Общий телекоммуникационный метод доступа. Общее описание	XXX.804.004 Д59	
39. Общий телекоммуникационный метод доступа. Руководство программиста	XXX.804.004 Д60	
40. Общий телекоммуникационный метод доступа. Руководство программиста	XXX.804.004 Д61	
41. Общий телекоммуникационный метод доступа. Руководство программиста	XXX.804.004 Д62	
42. Сообщения общего телекоммуникационного метода доступа. Руководство программиста	XXX.804.004 Д91	
43. Общий телекоммуникационный метод доступа. Отладка. Руководство программиста	XXX.804.005 Д97	

Наименование документа	Обозначение документа	Примечание
44. Общий телекоммуникационный метод доступа. Комплексы ЕС-7920. Руководство программиста	XXX.180.003 Д1	
45. Дополнительные средства телеобработки данных. Руководство программиста	XXX.804.006 Д96	
<i>Программное обеспечение диалогового удаленного ввода заданий</i>		
46. Система ввода заданий с абонентских пунктов в режиме диалога. Руководство программиста	XXX.804.005 Д69	
47. Диалоговый удаленный ввод заданий. Руководство программиста	XXX.804.004 Д70	
48. Диалоговый удаленный ввод заданий. Руководство оператора	XXX.804.006 Д92	
<i>Программное обеспечение машинной графики</i>		
49. Средства программирования для алфавитно-цифровых дисплеев. Руководство программиста	XXX.804.006 Д56	
50. Средства программирования для графических дисплеев. Руководство программиста	XXX.804.006 Д57	
51. Пакет графических подпрограмм для языков Фортран, Кобол и ПЛ/1. Руководство программиста	XXX.804.006 Д64	
52. Фортран IV. Базисные подпрограммы для графопостроителей. Руководство программиста	XXX.804.006 Д87	
53. Фортран IV. Функциональные подпрограммы для графопостроителей. Руководство программиста	XXX.804.006 Д88	
54. Сообщения средств графического программирования. Руководство программиста	XXX.804.005 Д93	
<i>Программные средства восстановления</i>		
55. Средства восстановления в операционной системе ОС ЕС. Руководство системного программиста	XXX.804.006 Д21	
56. Контрольная точка-повторный пуск. Руководство программиста	XXX.804.006 Д6	

Наименование документа	Обозначение документа	Примечание
57. Независимая программа редактирования и печати информации об ошибках SEREP. Руководство оператора	XXX.804.006 Д73	
58. Программы IFCDIP00 и EREP. Руководство оператора	XXX.804.006 Д74	
59. Сообщения средств восстановления и контрольной точки. Руководство оператора	XXX.804.006 Д32	
60. Средства восстановления системы ввода-вывода. Руководство системного программиста	XXX.804.006 Д66	
61. Записи программ регистрации ошибок периферийных устройств. Руководство по техническому обслуживанию	XXX.804.006 Д78	
62. Программа обработки прерываний от схем контроля. Руководство по техническому обслуживанию	XXX.804.006 Д107	
63. Программа неавтономного тестирования. Руководство оператора	XXX.804.006 Д50	
64. Записи программной регистрации машинных и канальных ошибок цифровой ЭВМ ЕС-1022. Руководство по техническому обслуживанию	XXX.804.004 Д85	
65. Записи программной регистрации машинных и канальных ошибок цифровой ЭВМ ЕС-1030. Руководство по техническому обслуживанию	XXX.804.004 Д76	
66. ЭВМ цифровая ЕС-1033. Записи программной регистрации машинных и канальных ошибок. Руководство по техническому обслуживанию	XXX.700.000 Д40	
67. Средства восстановления для ЭВМ ЕС-1035. Руководство системного программиста	XXX.180.001 Д1	
68. Записи программной регистрации машинных и канальных ошибок ЭВМ ЕС-1045. Руководство по техническому обслуживанию	XXX.990.179 Д18	

Наименование документа	Обозначение документа	Примечание
69. Записи программной регистрации машинных и канальных ошибок цифровой ЭВМ ЕС-1050. Руководство по техническому обслуживанию	XXX.804.004 Д77	
70. Записи программ регистрации машинных и канальных ошибок ЭВМ ЕС-1052. Руководство по техническому обслуживанию	XXX.700.000 Д40	
71. Записи программной регистрации машинных и канальных ошибок цифровой ЭВМ ЕС-1060. Руководство по техническому обслуживанию	XXX.804.006 Д108	
<i>Программное обеспечение многомашинных комплексов</i>		
72. Управление средствами комплексирования. Руководство системного программиста	XXX.804.006 Д90	
<i>Программные средства учета и измерений</i>		
73. Системная мониторная программа. Руководство системного программиста	XXX.804.005 Д72	
74. Универсальное средство трассировки. Руководство системного программиста	XXX.804.006 Д79	
<i>Средства программирования для оптическогочитающего устройства</i>		
75. Средства программирования для оптическогочитающего устройства. Руководство программиста	XXX.804.006 Д109	
<i>Программные средства сортировки и объединения данных</i>		
76. Сортировка - объединение. Руководство программиста	XXX.804.006 Д38	
<i>Система программирования Ассемблер</i>		
77. Ассемблер. Основные возможности. Описание языка	XXX.804.003 Д15	
78. Ассемблер. Макросредства. Описание языка	XXX.804.003 Д16	
79. Ассемблер. Руководство программиста	XXX.804.003 Д17	

Наименование документа	Обозначение документа	Примечание
<i>Алгол-60</i>		
80. Алгол-60. Описание языка	XXX.804.001-01 Д41	
81. Алгол-60. Руководство программиста	XXX.804.001-01 Д42	
<i>Кобол</i>		
82. Кобол. Описание языка	XXX.804.002 Д48	
83. Кобол. Руководство программиста	XXX.804.002 Д49	
<i>ПЛ/1</i>		
84. ПЛ/1. Описание языка	XXX.804.002 Д45	Часть 1 Часть 2
85. ПЛ/1. Описание языка	XXX.804.002 Д53	
86. ПЛ/1. Руководство программиста	XXX.804.002 Д46	
87. ПЛ/1. Библиотека подпрограмм. Руководство программиста	XXX 804.002 Д47	
88. ПЛ/1. Справочник. Руководство программиста	XXX.804.002 Д81	
89. ПЛ/1. Сообщения компилятора. Руководство программиста	XXX.804. 002 Д82	
<i>РПГ</i>		
90. РПГ. Описание языка	XXX.804.001-01 Д39	
91. РПГ. Руководство программиста	XXX.804.001-01 Д40	
<i>Фортран</i>		
92. Фортран IV. Описание языка	XXX.804.001-01 Д18	
93. Фортран IV. Руководство программиста	XXX.804.002 Д19	
94. Фортран IV. Библиотека подпрограмм. Руководство программиста	XXX.804.001-01 Д20	
<i>Редактор связей. Загрузчик</i>		
95. Редактор связей. Руководство программиста	XXX.804.003 Д10	
96. Загрузчик. Руководство программиста	XXX.804.003 Д43	

Приложение 2.

Список стандартных действий оператора

1. Если в задании не был запрошен максимальный уровень сообщений системы, ввести задание повторно, указав в операторе JOB параметр MSGLEVEL = (1,1)
2. Сохранить консольный протокол. В системах с мультиконсольным обеспечением получить и сохранить распечатку сборного протокола.
3. Сохранить все перфокарты из входного потока, связанные с заданием.

4. Сохранить все распечатки результатов выполнения задания из выходного потока.

5. Если во время аварийного завершения пункта задания не был получен полный дамп основной памяти, выполнить задание повторно, добавив в соответствующий пункт задания оператор SYSABEND DD.

6. Если во время аварийного завершения пункта задания не был получен дамп используемого им раздела основной памяти, выполнить задание повторно, добавив в соответствующий пункт задания оператор SYSUDUMP DD.

7. Подготовить полученную при генерации программы управления сообщениями (MCP) распечатку трансляции.

8. С помощью программы IMDSADMP записать на том магнитной ленты дамп основной памяти и страничных наборов данных. После перезагрузки для получения распечатки этого дампа выполнить программу IMDPRDMP с функцией GO. Сохранить том магнитной ленты с дампом и распечатку.

9. Сохранить распечатку результатов выполнения программы.

10. Сохранить дампы.

11. Подготовить распечатку второй стадии генерации используемой операционной системы.

12. Распечатать содержимое журнала регистрации записей об ошибках (набора данных SYS1. LOGREC) с помощью программы EREP. Сохранить полученную распечатку.

13. Сохранить распечатку трансляции выполняемой в задании программы.

14. Проверить работоспособность указанного в сообщении периферийного устройства с помощью программы неавтономного тестирования периферийных устройств OLTER. Сохранить полученные распечатки.

15. Распечатать оглавление указанного тома с помощью программы IEHLISIT, указав параметр DUMP.

16. Распечатать справочник указанного библиотечного набора данных с помощью программы IEHLIST.

17. Распечатать соответствующую процедуру с помощью программы IEVRTPCN.

18. Обратиться к системному программисту.

19. Обратиться к инженерам по обслуживанию ЭВМ.

20. Выполнить независимую программу SEREP и сохранить распечатку результатов ее выполнения.

21. Выполнить указанную в описании сообщения программу и сохранить распечатку ее результатов.

22. Загрузить операционную систему заново.

Литература

1. Единая система ЭВМ/Под общ. ред. А. М. Ларьонов. М., Статистика, 1974.

2. Система математического обеспечения ЕС ЭВМ/Под общ. ред. А. М. Ларьонов. М., Статистика, 1974.

3. Брич З. С., Воюш В. И., Дегтярева Г. С., Ковалевич Э. В. Программирование на языке Ассемблера ЕС ЭВМ. М., Статистика, 1975.

4. Наумов В. В., Пеледов Г. В., Тимофеев Ю. А., Чекалов А. Г. Супервизор ОС ЕС ЭВМ. М., Статистика, 1975.

5. Пеледов Г. В., Райков Л. Д. Введение в ОС ЕС ЭВМ. М., Статистика, 1977.

6. Митрофанов В. В., Одинцов Б. В. Программы обслуживания ОС ЕС ЭВМ. М., Статистика, 1977.

7. Лебедев В. Н., Соколов А. П. Введение в систему программирования ОС ЕС. М., Статистика, 1978.

8. Рейтборг И. М. Пособие для оператора ЕС ЭВМ. М., Статистика, 1979.

9. Ерофеев В. И., Меркушов Ю. П., Першиков В. И., Соколов А. П. Средства отладки программ в ОС ЕС ЭВМ. М., Статистика, 1979.

Условные обозначения

1. Слова или наборы слов, записанные строчными буквами с допустимым употреблением цифр, дефисов и пробелов, представляют собой синтаксические переменные, которые необходимо заменять на соответствующие значения. Способ замены задается в описании формата. Все остальные знаки, присутствующие в описании формата, в том числе и прописные буквы, должны употребляться так, как они записаны.

2. Фигурные скобки { }, содержащие несколько значений, обозначают, что нужно сделать выбор одного из них.

3. Квадратные скобки [], содержащие одно или несколько значений, обозначают, что нужно либо опустить значение, либо сделать выбор одного из них.

4. Многоточие «...» обозначает, что предшествующее значение может быть употреблено один раз или повторено несколько раз (т. е. обозначает список значений). При этом список значений обычно заключается в круглые скобки, и элементы списка отделяются друг от друга запятой. Если список состоит из одного элемента, то обычно допускается опустить заключающие его круглые скобки.

5. Значениями операндов или параметров макрокоманд могут быть абсолютные выражения, десятичные цифры, регистры 2-12, в отдельных случаях регистры 0 и 1, выражения, допустимые в адресных константах типа А или командах типа RX. Если в качестве значения операнда или параметров указывается регистр, то его следует заключить в круглые скобки. Более точно допустимые значения операндов или параметров указаны в эксплуатационной документации по макрокомандам операционной системы ОС ЕС.

Оглавление

Предисловие	3
Глава 1. Ассемблер	5
1.1. Основные элементы языка Ассемблера	5
1.1.1. Операторы	5
1.1.2. Термы	6
1.1.3. Выражения	8
1.2. Машинные команды	9
1.3. Команды Ассемблера	17
1.3.1. Общие сведения о константах	23
1.4. Макросредства Ассемблера	23
1.4.1. Макроопределение	27
1.4.2. Макрокоманда	27
1.4.3. Условное ассемблирование	31
Глава 2. Команды оператора ОС ЕС ЭВМ	32
2.1. Общие сведения	32
2.2. Информационные команды	33
2.2.1. Команда CONTROL	33
2.2.2. Команда DISPLAY	37
2.2.3. Команда LOG	38
2.2.4. Команда MONITOR	39
2.2.5. Команда MSGRT	39
2.2.6. Команда REPLY	40
2.2.7. Команда STOPMN	40
2.3. Команды управления операционной системой	41
2.3.1. Команда CANCEL	41
2.3.2. Команда DEFINE	42
2.3.3. Команда HALT	42
2.3.4. Команда HOLD	42
2.3.5. Команда MODIFY	43
2.3.6. Команда RELEASE	43
2.3.7. Команда RESET	43
2.3.8. Команда SET	44
2.3.9. Команда START	44
2.3.10. Команда STOP	45
2.3.11. Команда WRITELOG	45
2.3.12. Команда SWITCH	45
2.4. Команды управления периферийными устройствами	45
2.4.1. Команда MOUNT	46
2.4.2. Команда UNLOAD	46
2.4.3. Команда VARY	47
2.4.4. Команды SWAP	47
2.5. Команды управления консолями	47
2.5.1. Команда VARY CONSOLE	48
2.5.2. Команды основной консоли	49
2.6. Управление программой системного ввода (RDR)	52
2.6.1. Команда START (RDR)	52
2.6.2. Команда STOP (RDR)	53
2.6.3. Команда CANCEL (RDR)	53
2.7. Управление инициатором выполнения заданий (INIT)	53
2.7.1. Команда START (INIT)	53
2.7.2. Команда MODIFY (INIT)	54

2.7.3. Команда STOP (INIT)	54
2.8. Управление программой системного вывода (WTR)	54
2.8.1. Команда START (WTR)	54
2.8.2. Команда MODIFY (WTR)	55
2.8.3. Команда CANCEL (WTR)	53
2.8.4. Команда STOP (WTR)	56
2.9. Управление программой прямого системного вывода (DSO)	56
2.9.1. Команда START (DSO)	56
2.9.2. Команда MODIFY (DSO)	57
2.9.3. Команда STOP (DSO)	57
2.10. Управление работой универсального средства трассировки	57
2.10.1. Команда START (GTF)	58
2.10.2. Команда STOP (GTF)	58
2.11. Управление работой системы ДУВЗ (CRJE)	59
2.11.1. Команда START (CRJE)	59
2.11.2. Команда MODIFY (CRJE)	59
2.11.3. Команда CENOUT	60
2.11.4. Команда SHOW	60
2.11.5. Команда USERID	61
2.11.6. Команда BRDCST	61
2.11.7. Команда MSG	62
2.11.8. Команда STOP (CRJE)	62
Глава 3. Процедура начальной загрузки	63
3.1. Общие сведения	63
3.2. Использование пульта ЭВМ	64
3.3. Определение параметров ОС ЕС	64
3.4. Управление консолями	71
3.4.1. Уровни консольного обеспечения ОС ЕС	71
3.4.2. Приемы работы с пишущей машинкой ЕС-7070	72
3.4.3. Приемы работы с пишущей машинкой ЕС-7077	72
3.4.4. Приемы работы с составной консолью	72
3.4.5. Ввод команд оператора ОС ЕС ЭВМ во входном потоке	73
3.4.6. Дисплей в качестве консолей	73
3.4.7. Приемы работы с дисплеями ЕС-7064 и ЕС-7065	74
3.4.8. Приемы работы с дисплеем ЕС-7066	75
3.4.9. Приемы работы с дисплеем ЕС-7970-01	76
Глава 4. Язык управления заданиями	77
4.1. Оператор задания (оператор JOB)	77
4.1.1. Параметр учетной информации	77
4.1.2. Параметр идентификации программиста	78
4.1.3. Параметр CLASS	78
4.1.4. Параметр PRTY	78
4.1.5. Параметр MSGLEVEL	78
4.1.6. Параметр MSGCLASS	79
4.1.7. Параметр TYPRUN	79
4.1.8. Параметр ADDRSPC	79
4.1.9. Параметр REGION	80
4.1.10. Параметр ROLL	80
4.1.11. Параметр TIME	80
4.1.12. Параметр COND	81
4.1.13. Параметр RD	81
4.1.14. Параметр RESTART	82
4.1.15. Параметр NOTIFY	82
4.2. Оператор пункта задания (оператор EXEC)	82
4.2.1. Параметр PGM	83
4.2.2. Параметр PROC	83
4.2.3. Параметр PARM	84
4.2.4. Параметр ACCT	84
4.2.5. Параметр DPRTY	84
4.2.6. Параметры ADDRSPC, REGION, TIME и RD	85

4.2.7. Параметр ROLL	85
4.2.8. Параметр COND	85
4.3. Оператор определения данных (оператор DD)	86
4.3.1. Параметр *	87
4.3.2. Параметр DATA	87
4.3.3. Параметр DUMMY	87
4.3.4. Параметр DYNAM	88
4.3.5. Параметр DDNAME	88
4.3.6. Параметр SYSOUT	88
4.3.7. Параметр DSNAME	88
4.3.8. Параметр UNIT	89
4.3.9. Параметр VOLUME	90
4.3.10. Параметр DCB	91
4.3.11. Параметр LABEL	91
4.3.12. Параметр DISP	92
4.3.13. Параметр SPACE	93
4.3.14. Параметр SPLIT	95
4.3.15. Параметр SUBALLOC	95
4.3.16. Параметр SEP	96
4.3.17. Параметр AFF	96
4.3.18. Параметр OUTLIM	96
4.3.19. Параметр DLM	96
4.3.20. Параметр QNAME	97
4.3.21. Параметр TERM	97
4.3.22. Параметр COPIES	97
4.3.23. Параметр FCB	97
4.3.24. Параметр UCS	98
4.4. Оператор команды	98
4.5. Оператор комментариев	98
4.6. Ограничительный оператор	98
4.7. Пустой оператор	99
4.8. Оператор процедуры (PROC)	99
4.9. Оператор конца процедуры (PEND)	99
4.10. Процедуры	100

Глава 5. Макрокоманды супервизора 103

5.1. Управление задачами	103
5.1.1. Макрокоманда ATTACH	103
5.1.2. Макрокоманда DETACH	105
5.1.3. Макрокоманда CHAP	106
5.1.4. Макрокоманда EXTRACT	106
5.2. Синхронизация событий	107
5.2.1. Макрокоманда WAIT	107
5.2.2. Макрокоманда WAITR	108
5.2.3. Макрокоманда POST	108
5.3. Управление связями	108
5.3.1. Макрокоманда LINK	108
5.3.2. Макрокоманда XCTL	109
5.3.3. Макрокоманда LOAD	109
5.3.4. Макрокоманда DELETE	110
5.3.5. Макрокоманда IDENTIFY	110
5.3.6. Макрокоманда SAVE	110
5.3.7. Макрокоманда RETURN	111
5.3.8. Макрокоманда CALL	111
5.4. Управление основной памятью	112
5.4.1. Макрокоманда GETMAIN	112
5.4.2. Макрокоманда FREEMAIN	114
5.5. Служба времени	115
5.5.1. Макрокоманда TIME	115
5.5.2. Макрокоманда STIMER	116
5.5.3. Макрокоманда TTIMER	117

5.6.	Управление последовательно используемыми ресурсами	117
5.6.1.	Макрокоманда ENQ	117
5.6.2.	Макрокоманда DEQ	118
5.7.	Обработка программных прерываний	119
5.7.1.	Макрокоманда SPIE	119
5.8.	Аварийное завершение задач	119
5.8.1.	Макрокоманда ABEND	119
5.8.2.	Макрокоманда STAE	120
5.9.	Выдача дампа	121
5.9.1.	Макрокоманда SNAP	121
5.10.	Вывод сообщений	122
5.10.1.	Макрокоманда WTL	122
5.10.2.	Макрокоманда WTO	123
5.10.3.	Макрокоманда WTOR	123
5.10.4.	Макрокоманда DOM	124
5.11.	Макрокоманды оверлейной структуры	125
5.11.1.	Макрокоманда SEGLD	125
5.11.2.	Макрокоманда SEGWT	125
5.12.	Макрокоманды управления страницами виртуальной памяти	125
5.12.1.	Макрокоманда PGRlse	125
5.12.2.	Макрокоманда STATUS	126
5.13.	Контрольная точка	126
5.13.1.	Макрокоманда CHKPT	126
Глава 6.	Управление данными ОС ЕС	127
6.1.	Общие сведения	127
6.1.1.	Форматы данных	127
6.1.2.	Управляющие символы логических записей	128
6.1.3.	Формат и емкость дорожек устройств прямого доступа	129
6.2.	Организация данных в ОС ЕС	131
6.2.1.	Последовательный набор данных	131
6.2.2.	Прямой набор данных	132
6.2.3.	Библиотечный набор данных	132
6.2.4.	Индексно-последовательный набор данных	132
6.2.5.	Оглавление тома прямого доступа	134
6.3.	Метки	135
6.3.1.	Метки томов	135
6.3.2.	Метки набора данных на магнитной ленте	135
6.3.3.	Форматы блоков оглавления тома прямого доступа	136
Глава 7.	Макрокоманды управления данными	141
7.1.	Общие сведения	141
7.2.	Макрокоманда DCB	141
7.2.1.	Операнды макрокоманды DCB для BISAM	141
7.2.2.	Операнды макрокоманды DCB для BDAM	143
7.2.3.	Операнды макрокоманды DCB для BPAM	144
7.2.4.	Операнды макрокоманды DCB для BSAM	146
7.2.5.	Операнды макрокоманды DCB для QISAM	149
7.2.6.	Операнды макрокоманды DCB для QSAM	151
7.3.	Макрокоманды организации обработки данных	152
7.3.1.	Макрокоманда CHECK	152
7.3.2.	Макрокоманда CLOSE	153
7.3.3.	Макрокоманда DCBD	154
7.3.4.	Макрокоманда FEOV	155
7.3.5.	Макрокоманда OPEN	155
7.4.	Макрокоманды управления буферами	157
7.4.1.	Макрокоманда BUILD	158
7.4.2.	Макрокоманда BUILDRCDD	158
7.4.3.	Макрокоманда GETPOOL	159
7.4.4.	Макрокоманда GETBUF	160

7.4.5. Макрокоманда	FREEBUF	160
7.4.6. Макрокоманда	FREEPOOL	160
7.4.7. Макрокоманда	FREEDBUF	160
7.4.8. Макрокоманда	RELSE	161
7.4.9. Макрокоманда	TRUNC	161
7.5. Макрокоманды управления периферийными устройствами		161
7.5.1. Макрокоманда	BSP	161
7.5.2. Макрокоманда	CNTRL	162
7.5.3. Макрокоманда	NOTE	163
7.5.4. Макрокоманда	POINT	163
7.5.5. Макрокоманда	PRTOV	164
7.5.6. Макрокоманда	SETPRT	164
7.6. Макрокоманды управления операциями ввода-вывода		
	BDAM	166
7.6.1. Макрокоманда	READ(BDAM)	166
7.6.2. Макрокоманда	RELEX	168
7.6.3. Макрокоманда	WRITE(BDAM)	168
7.7. Макрокоманды управления операциями ввода-вывода		
	BISAM	171
7.7.1. Макрокоманда	READ(BISAM)	171
7.7.2. Макрокоманда	WRITE(BISAM)	172
7.8. Макрокоманды управления операциями ввода-вывода		
	BPAM	173
7.8.1. Макрокоманда	BLDL	173
7.8.2. Макрокоманда	FIND	174
7.8.3. Макрокоманда	STOW	174
7.9. Макрокоманды управления операциями ввода-вывода		
	BSAM	176
7.9.1. Макрокоманда	READ(BSAM)	176
7.9.2. Макрокоманда	WRITE(BSAM)	176
7.10. Макрокоманды управления операциями ввода-вывода		
	QISAM	177
7.10.1. Макрокоманда	ESETL	177
7.10.2. Макрокоманда	GET(QISAM)	177
7.10.3. Макрокоманда	PUT(QISAM)	178
7.10.4. Макрокоманда	PUTX(QISAM)	178
7.10.5. Макрокоманда	SETL	178
7.11. Макрокоманды управления операциями ввода-вывода		
	QSAM	179
7.11.1. Макрокоманда	GET(QSAM)	179
7.11.2. Макрокоманда	PUT(QSAM)	179
7.11.3. Макрокоманда	PUTX(QSAM)	180
7.12. Макрокоманды обработки ошибок ввода-вывода		180
7.12.1. Макрокоманда	SYNADAF	180
7.12.2. Макрокоманда	SYNADDRLS	182
7.12.3. Содержимое регистров при входе в программу	SYNAD	182
Глава 8. Редактор связей		184
8.1. Функции Редактора связей		184
8.2. Оператор EXEC для Редактора связей		185
8.3. Имя Редактора связей		185
8.4. Характеристики загрузочных модулей и режимы работы Редактора связей		185
8.4.1. Характеристики загрузочных модулей		186
8.4.2. Режим работы Редактора связей		188
8.5. Размер раздела основной памяти		190
8.6. Коды возврата Редактора связей		190
8.7. Операторы DD для Редактора связей		191
8.8. Каталогизированные процедуры для Редактора связей		192
8.9. Динамический вызов Редактора связей		192

8.10.	Управляющие операторы Редактора связей	193
8.10.1.	Включение дополнительного ввода Редактора связей	193
8.10.2.	Дополнительные автоматически вызываемые библиотеки	193
8.10.3.	Определение имени, дополнительного имени и точки входа загрузочного модуля	194
8.10.4.	Редактирование модуля	195
8.10.5.	Оверлейные программы	196
Глава 9.	Загрузчик	197
9.1.	Функции Загрузчика	197
9.2.	Оператор EXEC для Загрузчика	198
9.3.	Операторы DD для Загрузчика	199
9.4.	Динамический вызов Загрузчика	200
9.5.	Коды возврата Загрузчика	201
Глава 10.	Программа сортировка-объединение	202
10.1.	Функции программы	202
10.1.1.	Общие сведения	202
10.1.2.	Требования к конфигурации ЭВМ	204
10.2.	Управляющие операторы программы	204
10.3.	Выходы на программы пользователя ЭВМ	208
10.4.	Вызов программы СОРТИРОВКА-ОБЪЕДИНЕНИЕ	208
10.5.	Операторы языка управления заданиями для программы СОРТИРОВКА-ОБЪЕДИНЕНИЕ	209
10.6.	Динамический вызов программы СОРТИРОВКА-ОБЪЕДИНЕНИЕ	210
10.7.	Коды возврата	211
Глава 11.	Программное обеспечение устройств машинной графики	212
11.1.	Графический метод доступа	212
11.1.1.	Макрокоманда DCB	212
11.1.2.	Макрокоманда OPEN	213
11.1.3.	Макрокоманда GREAD	213
11.1.4.	Макрокоманда GWRITE	215
11.1.5.	Макрокоманда GCNTRL	216
11.1.6.	Макрокоманда GREADR	218
11.1.7.	Макрокоманда CLOSE	218
11.1.8.	Базисный метод обработки сигналов внимания	213
11.1.9.	Специальный метод обработки сигналов внимания	222
11.1.10.	Управление буфером для графических дисплеев	223
11.2.	Пакет графических подпрограмм для языков Фортран, Кобол и ПЛ/1	225
11.2.1.	Общие сведения	225
11.2.2.	Общие аргументы подпрограмм ПГП	226
11.2.3.	Подпрограммы инициализации и завершения	223
11.2.4.	Подпрограммы настройки	223
11.2.5.	Подпрограммы порождения образов	229
11.2.6.	Подпрограммы группировки элементов	231
11.2.7.	Буферные подпрограммы	231
11.2.8.	Подпрограммы управления образами	232
11.2.9.	Управление воспроизведением образов	232
11.2.10.	Подпрограмма ввода с клавиатуры и анализа буферных данных	232
11.2.11.	Подпрограммы обработки запросов	233
11.2.12.	Подпрограммы для светового пера	236
11.2.13.	Подпрограммы для планшета (только для ЕС-7905)	237
11.2.14.	Отладочные средства (только для языка Фортран)	238
11.2.15.	Вызов подпрограмм ПГП из программ, написанных на языке Ассемблера	239

11.2.16.	Растровое описание графических символов	239
11.2.17.	Преобразование координат—подпрограмма CNVRT	241
11.2.18.	Прямое порождение графических приказов — под- программа ORGEN	241
11.2.19.	Преобразование входных-выходных числовых дан- ных — подпрограмма BCNV (для программ на язы- ках Ассемблер и Фортран)	242
11.3.	Базисное программное обеспечение графопостроителей	243
11.3.1.	Общие сведения	243
11.3.2.	Подпрограмма PLOTS	243
11.3.3.	Подпрограмма SYMBOL	246
11.3.4.	Подпрограмма NUMBER	248
11.3.5.	Подпрограмма SCALE — масштабирование мас- сивов данных	248
11.3.6.	Подпрограмма AXIS — вычерчивание осей	248
11.3.7.	Подпрограмма LINE — вычерчивание графиков	249
11.4.	Функциональные подпрограммы для графопостроителей	249
11.4.1.	Подпрограммы общего назначения	250
11.4.2.	Научно-технические подпрограммы	251
11.4.3.	Экономические подпрограммы	254
11.4.4.	Чертежные подпрограммы	255
11.5.	Средства описания изображений для графических дисплеев	257
11.5.1.	Формат графического приказа	257
11.5.2.	Макрокоманды инициализации	257
11.5.3.	Вспомогательные макрокоманды	259
11.5.4.	Макрокоманды образования приказов	259
11.5.5.	Макрокоманды образования данных	266
11.5.6.	Макрокоманды, блоки и программы управления областью вывода графических данных	267
11.5.7.	Проблемно-ориентированные программы	269

Глава 12. Диалоговый удаленный ввод заданий (ДУВЗ) 275

12.1.	Общие сведения	275
12.2.	Генерация ДУВЗ	275
12.2.1.	Требования к операционной системе	275
12.2.2.	Макрокоманды генерации ДУВЗ	276
12.2.3.	Программные выходы	280
12.2.4.	Генерация модулей ДУВЗ	282
12.2.5.	Размещение и инициализация наборов данных системы ДУВЗ	283
12.2.6.	Процедура запуска ДУВЗ	285
12.2.7.	Процедура системного ввода	286
12.3.	Команды оператора АП	287
12.3.1.	CANCEL — отметить задание	287
12.3.2.	CONTINUE — продолжить вывод	287
12.3.3.	DELETE — удалить набор данных	287
12.3.4.	EDIT — редактировать набор данных	288
12.3.5.	EXEC — выполнить группу команд	294
12.3.6.	LISTBC — выдать копии циркулярных сообщений	295
12.3.7.	LISTDS — выдать информацию о наборе данных ДУВЗ	295
12.3.8.	LISTLIB — выдать оглавление библиотек	295
12.3.9.	LOGOFF — закончить сеанс	295
12.3.10.	LOGON — начать сеанс	296
12.3.11.	OUTPUT — выдать результаты задания	296
12.3.12.	SEND — послать сообщение	296
12.3.13.	STATUS — запросить информацию о задании	296
12.3.14.	SUBMIT — передать в пакетную обработку	297
12.3.15.	TABSET — установить табуляцию	297
12.3.16.	VARY — изменить характеристики ввода-вывода (издание 6.1)	297

Глава 13. Базисный телекоммуникационный метод доступа	298
13.1. Определение системы телеобработки	298
13.1.1. DCB — определить группу каналов связи	298
13.1.2. DFTRMLST — построить абонентский список	300
13.1.3. CHGNTRY — изменить элемент абонентского списка	304
13.2. Управление буферами	305
13.2.1. REQBUF — запросить буфер	305
13.2.2. RELBUF — освободить буфер	305
13.3. Трансляция сообщений	305
13.3.1. ASMTRTAB — построить таблицу трансляции	308
13.3.2. TRNSLATE — транслировать сообщение	309
13.4. Активизация и деактивизация системы телеобработки	309
13.4.1. OPEN и CLOSE — открыть и закрыть группу каналов связи	309
13.4.2. LOPEN — повторно открыть канал связи	309
13.5. Прием и передача сообщений	310
13.5.1. READ и WRITE — читать и писать блок данных	310
13.5.2. RESETPL — сброс канала связи	311
13.5.3. WAIT — ждать окончания операции ввода-вывода	311
13.5.4. TWAIT — ожидать окончания одной из операций ввода-вывода	314
13.6. Регистрация ошибок	315
13.6.1. LERB — построить блок регистрации ошибок в канале связи	315
13.6.2. LERPRT — распечатать содержимое счетчиков на консоли оператора	315
13.7. Оперативная проверка устройств	316
13.7.1. Формат тестового сообщения для стартстоппного режима	316
13.7.2. Форматы запроса тестов для синхронных каналов связи (кроме EC-7920)	317
13.7.3. Формат тестового сообщения для локальных дисплеев EC-7920	318
13.7.4. Формат тестового сообщения для удаленных дисплеев EC-7920	319
13.7.5. Макрокоманда ONLTST	319
13.8. Особенности программирования для различных АП	320
13.8.1. АП-1 (EC-8501)	320
13.8.2. АП-2 (EC-8502) и АП-3 (EC-8503) (стартстоппные)	322
13.8.3. АП-2 (EC-8502) синхронный	322
13.8.4. АП-4 (EC-8504)	325
13.8.5. АП-5 (EC-8505)	326
13.8.6. АП-61, АП-62, АП-63, АП-64 (EC-8561, EC-8562, EC-8563, EC-8564)	326
13.8.7. АП-70 (EC-8570)	329
13.8.8. Телеграфный аппарат (ТА)	331
13.8.9. Комплексы EC-7920	332
Глава 14. Общий телекоммуникационный метод доступа	338
14.1. Программа управления сообщениями	338
14.1.1. Активизация и деактивизация ПУС	338
14.1.2. Определение набора данных	343
14.1.3. Управление каналами связи	346
14.1.4. Обработчик сообщений	351
14.1.5. Программы пользователя в ПУС	366
14.2. Прикладные программы ОТМД	366
14.2.1. Макрокоманды DCB в прикладной программе	366
14.2.2. Активизация и деактивизация сопряжения с прикладной программой	338
14.2.3. Формат резервных полей в рабочей области	368
14.2.4. Макрокоманды доступа к данным	369
14.2.5. Выход на подпрограмму SYNAD	370

14.2.6. Макрокоманды управления системой телеобработки	371
14.2.7. Возврат сообщений из очереди	374
14.2.8. Координация контрольных точек макрокомандой CKREQ	374
14.3. Оперативное управление	374
14.3.1. Общие сведения	374
14.3.2. Команды оператора	375
14.4. Средства отладки	379
14.4.1. Программа распечатки трассировки (IEDQXB)	379
14.4.2. Программа распечатки набора данных очередей сообщений (IEDQXC)	379
14.5. Запись об ошибке	380
Глава 15. Программы обслуживания ОС ЕС	382
15.1. Автономные программы обслуживания	382
15.1.1. Общие сведения	382
15.1.2. Возможности и функциональные операторы автономных программ обслуживания	383
15.2. Программы обслуживания наборов данных	386
15.2.1. Общие сведения	386
15.2.2. Возможности и управляющие операторы программ обслуживания наборов данных	387
15.3. Системные программы обслуживания	405
15.3.1. Общие сведения	405
15.3.2. Возможности и управляющие операторы системных программ обслуживания	406
Глава 16. Системные коды аварийного завершения	428
Глава 17. Коды состояния ожидания	453
Приложения	458
Литература	470
Условные обозначения	471

Валерий Петрович Данилочкин,
Валерий Викторович Митрофанов,
Борис Васильевич Одинцов,
Геннадий Васильевич Пеледов

Операционная система ОС ЕС

Научный редактор *Л. Д. Райков*
Рецензент *А. В. Камордин*
Зав. редакцией *И. Г. Дмитриева*
Редакторы *Э. С. Асланова, Л. И. Ганина, Л. Д. Григорьева*
Мл. редакторы *М. В. Ульянова, И. Н. Кубасова, Г. В. Розанова*
Техн. редактор *Р. Н. Феоктистова*
Корректоры *Т. М. Васильева, Г. В. Хлопцева, А. Т. Сидорова*
Худож. редактор *Э. А. Смирнов*
Переплет художника *А. К. Малкина*

ИБ № 851

Сдано в набор 05.05.80. Подписано в печать 10.10.80. А12648. Формат 60×90¹/₁₆.
Бум. тип. № 2 Гарнитура «Литературная». Печать высокая. П. л. 30. Усл. п. л. 30.
Уч.-изд. л. 32,34. Тираж 20 000 экз. Заказ 3794. Цена 1 р. 80 к.

Издательство «Статистика», Москва, ул. Кирова, 39.

Областная типография управления издательств, полиграфии и книжной торговли Ивановского облисполкома, 153628, г. Иваново, ул. Типографская, 6.

