ОПЕРАЦИОННАЯ СИСТЕМА ДОС ЕС

*

Справочник

ОПЕРАЦИОННАЯ СИСТЕМА ДОС ЕС

Справочник

МОСКВА «СТАТИСТИКА» 1978

6Ф7.3
О-60

Авторы

Ю. Ю. Битель, В. И. Воюш, Р. В. Горбунова, Е. И. Дащинская, А. Г. Лях, З. А. Марук, М. Г. Скоромник

Операционная система ДОС ЕС. Справочник. М., О-60 «Статистика», 1978.

271 с. с ил.

На обороте тит. л. авт.: Ю. Ю. Битель, В. И. Воюш, Р. В. Горбунова [и др.].

В справочнике обобщен материал по операционной системе ДОС ЕС, широко применяемой на ЭВМ ЕС-1020, ЕС-1022, ЕС-1030. В нем приводятся краткие сведения о языке Ассемблера, форматах директив, операторов и макрокоманд системных программ, об использовании таких системных функций, как создание и обслуживание библиотек, редактирование и трансляция программ и др.

Справочник предназначен для системных и прикладных программистов ДОС ЕС, но он может быть полезен также программистам других операционных систем. Для эффективного использования книги читатель должен быть знаком с операционной системой ДОС ЕС.

О $\frac{30502\text{-}016}{008(01)\text{-}78}$ 94-77 6Ф7.3

© Издательство «Статистика», 1977

ПРЕДИСЛОВИЕ

Эффективность использования многообразных возможностей операционной системы ДОС ЕС, которая рассчитана на различные конфигурации оборудования и области применения ЭВМ Единой системы, в значительной мере зависит от имеющихся в распоряжении специалистов учебных и справочных пособий.

Настоящий справочник, предназначенный для различных категорий пользователей ДОС ЕС: программистов, системных программистов, операторов и обслуживающего персонала, — содержит информацию, необходимую указанным категориям специалистов для эффективного использования версии 2.0 операционной системы ДОС ЕС. Он охватывает те компоненты, которые составляют основу операционной системы и используются практически на любом вычислительном центре.

Основной объем материала справочника адресован программисту. Знание функциональных возможностей и способов использования управляющей программы ДОС ЕС (гл. 2 и 9) является обязательным условием эффективности любого конкретного применения вычислительной системы с ДОС ЕС. Вопросы организации данных на носителях и методов доступа к ним (гл. 3), а также сведения о средствах ДОС ЕС для перезаписи и сортировки информации (гл. 7) представляют интерес при решении проблем, связанных с обработкой большого объема данных. Ввиду тесной связи Ассемблера с другими компонентами системы информация для программирования на Ассемблере представлена в справочнике наиболее полно: язык Ассемблера (гл. 1), макрокоманды связи с Супервизором (гл. 2), макрокоманды логической системы управления вводом-выводом (гл. 3), макрокоманды перезаписи (гл. 7). Программисты, работающие с языками высокого уровня, могут дополнительно обращаться к краткому справочнику по языкам программирования ДОС ЕС, вышедшему в издательстве «Статистика». При использовании любого языка программирования неотъемлемую часть процесса подготовки проблемы к решению на ЭВМ составляют процедуры обслуживания библиотек (гл. 4), трансляции, редактирования и отладки программ (гл. 5 и 8) и обслуживания носителей данных (гл. 6).

Главы 8 и 9 представляют интерес главным образом для системных программистов, круг обязанностей которых включает работы по созданию вариантов операционной системы, по оценке эффективности и на-

дежности вычислительной системы, по учету использования времени ЭВМ и поиску причин сложных программных ошибок. Информация главы 8 может быть использована также обслуживающим персоналом ЭВМ для анализа данных о работе оборудования, локализации и определения причин устойчивых ошибок, оценки качества носителей.

Сведения о процедурах инициализации работы системы, запуска заданий и управления их выполнением (гл. 2), обслуживания библиотек и носителей данных (гл. 4 и 6), информация о способах использования программ, облегчающих поиски ошибок (гл. 8), обеспечивающих перезапись информации (гл. 7), необходимы операторам системы для обеспечения квалифицированного обслуживания ЭВМ.

Приведенное выше целевое разбиение материала справочника ни в коей мере не является абсолютным, так как и в других главах пользователь любой категории может найти полезные для себя либо просто интересные сведения. Более детальная информация (особенности эксплуатации, примеры использования и пр.) по включенным в справочник компонентам может быть найдена в документации на операционную систему ДОС ЕС и в вышедших в печати изданиях по этой системе [1,2].

Для поиска нужных сведений в справочнике следует пользоваться оглавлением и предметным указателем. Предварительно необходимо ознакомиться с принятыми соглашениями и условными обозначениями, приведенными в конце книги.

ГЛАВА 1

АССЕМБЛЕР

1.1. ОСНОВНЫЕ ЭЛЕМЕНТЫ ЯЗЫКА АССЕМБЛЕРА

1.1.1. АЛФАВИТ

Алфавит языка Ассемблера составляют следующие символы:
буквы от A до Z;

специальные знаки #, @, ¤ , которые Ассемблер разрешает использовать в тех же элементах языка, где и буквы;

цифры от 0 до 9;

специальные знаки +—= *( ) /&. , ′ и пробел.

Примечание. В комментариях, а также между парными апострофами в операндах могут использоваться все символы ДКОИ.

1.1.2. ОПЕРАТОРЫ

Операторы предназначены для записи алгоритмов задач. Они включают машинные команды, команды Ассемблера и операторы макросредств.

Оператор языка Ассемблера состоит из четырех полей: названия, операции, операндов и идентификации. Оператор записывается на специальном бланке кодирования в стандартном или произвольном формате записи, допускается продолжение оператора в следующих строках.

Поле названия состоит не более чем из восьми колонок. В нем может записываться символическое имя, идентифицирующее оператор. Символическим именем может быть простое символическое имя, параметр, имя перехода и параметры, соединенные с другими символами (в этом случае длина поля может превышать восемь колонок).

Поле операции состоит не более чем из восьми колонок. В нем записывается мнемонический код операции, который определяет функцию оператора.

Поле операндов предназначено для записи операндов оператора и комментариев. Операнды отделяются друг от друга запятой. Комментарии записываются по желанию программиста.

Поле идентификации состоит не более чем из восьми колонок. Оно предназначено для идентификации программы и указания номера оператора.

Стандартный формат записи операторов означает, что каждое поле оператора записывается в определенной графе бланка. Запись поля названия должна начинаться с самой левой позиции. Все поля оператора должны отделяться друг от друга по крайней мере одним пробелом.

Произвольный формат записи операторов означает, что стандартные значения колонок начала и конца оператора и колонки продолжения могут быть изменены с помощью оператора ICTL.

Продолжение оператора. Оператор может занимать более одной строки. Указателем того, что оператор продолжается на следующей строке, является любой символ, кроме пробела, в колонке указателя продолжения (колонка 72 в стандартном формате). Продолжение оператора записывается, начиная с колонки продолжения (колонка 16 в стандартном формате). Перфокарты, содержащие продолжение оператора, являются перфокартами продолжения.

1.1.3. ТЕРМЫ

Операнды операторов языка Ассемблера строятся из выражений, а выражения — из термов. Термом является самоопределенная величина (самоопределенный терм), простое символическое имя, значение счетчика адреса, характеристика длины, литерал. Каждый терм характеризуется значением, характеристикой длины и переместимостью.

Значение терма — это некоторая величина, которая определяется или транслятором, или самим термом непосредственно. Значение терма не должно превышать 16 777 215 (т. е. $2^{24} - 1$).

Характеристика длины терма, определяющего область памяти, равна длине этой области. Характеристика длины простого символического имени, определяемого оператором EQU, равна характеристике длины первого терма выражения из поля операндов этого оператора EQU. Для остальных термов характеристика длины равна 1.

Переместимость терма. Если значение терма при перемещении программы изменяется, терм является переместимым. Терм называется абсолютным, если его значение при перемещении программы не изменяется.

Самоопределенный терм — это число в десятичном (десятичный терм), шестнадцатеричном (шестнадцатеричный терм), двоичном (двоичный терм) и знаковом (знаковый терм) виде.

Десятичный терм записывается как последовательность десятичных цифр (не больше восьми) без знака. Шестнадцатеричный терм записывается как буква X, за которой следует последовательность шестнадцатеричных цифр, заключенная в апострофы (количество цифр не должно быть больше шести). Двоичный терм записывается как буква B, за которой следует последовательность двоичных цифр (не больше 24), заключенная в апострофы. Знаковый терм записывается как буква C, за которой следует последовательность любых символов ДКОИ,

заключенная в апострофы (количество символов не должно быть больше трех). Если в качестве символа этого терма необходимо использовать апостроф или знак &, то он должен записываться как два апострофа или два знака & соответственно. Самоопределенные термы являются абсолютными.

Простое символическое имя записывается в виде последовательности букв или цифр. Первым символом должна быть буква, количество символов не должно превышать восьми.

Имя определяется тогда, когда оно появляется в поле названия какого-либо оператора. Значением имени, называющего область памяти (т. е. команду, константу, программную секцию), является адрес крайнего левого байта этой области памяти. Значением имени, определяемого оператором EQU, является значение выражения, записанного в поле операндов этого оператора. Имена, использующиеся в некоторых операторах, должны быть предварительно определены. Это значит, что такое имя должно появиться в поле названия предшествующего оператора. Имена, называющие области памяти, являются перемещаемыми, остальные имена — абсолютными.

Счетчик адреса представляется знаком *. Этот знак, записанный в поле операндов, указывает, что значение счетчика адреса данного оператора используется как терм. Характеристикой длины счетчика адреса является характеристика длины того оператора, в котором счетчик адреса используется. Этот терм перемещаемый.

Характеристика длины записывается как буква L, апостроф и элемент, характеристика которого используется (этим элементом может быть простое символическое имя или счетчик адреса). Значением данного терма является характеристика длины элемента. Этот терм абсолютный.

Литерал представляется знаком равенства, за которым следует константа, записанная по правилам записи операнда команды DC (см. 1.4). Литерал можно использовать только в машинных командах. Значением литерала является адрес области, в которую транслятор помещает константу, заданную этим литералом. Литерал — перемещаемый терм.

1.1.4. ВЫРАЖЕНИЯ

Выражение представляет собой отдельный терм или арифметическую комбинацию термов. В качестве действий в выражениях используются: сложение (+), вычитание (—), умножение (×) и деление (/). Могут употребляться круглые скобки, уровень которых не должен быть больше пяти. Количество термов в выражении не должно превышать 16. Значение выражения должно находиться в пределах от -2^{24} до $2^{24}-1$. Характеристика длины выражения равна характеристике длины первого терма. Если первым термом выражения является самоопределенный терм или счетчик адреса в операторе EQU, то характеристика длины выражения принимается равной 1. Выражение является абсолютным, если его значение не изменяется при перемещении программы, в противном случае выражение называется перемещаемым.

1.2. МАШИННЫЕ КОМАНДЫ

В ЕС ЭВМ команды могут иметь один из пяти форматов: RR, RX, RS, SI, SS. Список машинных команд и форматы их записи на языке Ассемблера приведены в табл. 1.1 — 1.9. В таблицах используются следующие обозначения: R1, R2, R3 — регистры, X2 — регистр индекса, B1,B2 — регистры базы, I 2 — непосредственный операнд, D1, D2 — смещения, S1, S2 — неявные адреса, L, L1, L2 — длины операндов,[1] над мнемоническим кодом указывает, что команда устанавливает признак результата. Запись операндов команд формата RX приведена с указанием регистра индекса X2, а команд формата SS — с указанием длин операндов L, L1, L2. Если указывать эти элементы нет необходимости, то их можно опускать. При записи явного адреса, представленного в виде регистра базы и смещения, опускается только сам элемент X2 (или L, L1, L2), например L R1,D2(,B2) или CP D1(,B1), D2(,B2), а при записи неявного адреса — не только элемент X2 (или L, L1, L2), но и скобки, в которые он заключен, например L R1, S2 или CP S1,S2.

Таблица 1.1

Команды арифметики с фиксированной точкой

Команда	Мнемонический код	Формат	Машинный код	Адреса: явные или неявные
Сложение	A[1]	RX	5A	R1,D2(X2,B2) или R1,S2(X2)
Сложение полуслова	AH[1]	RX	4A	R1,D2(X2,B2) или R1,S2(X2)
Сложение кодов	AL[1]	RX	5E	R1,D2(X2,B2) или R1,S2(X2)
Сложение кодов	ALR[1]	RR	1E	R1,R2
Сложение	AR[1]	RR	1A	R1,R2
Деление	D	RX	5D	R1,D2(X2,B2) или R1,S2(X2)
Деление	DR	RR	1D	R1,R2
Умножение	M	RX	5C	R1,D2(X2,B2) или R1,S2(X2)
Умножение полуслова	MH	RX	4C	R1,D2(X2,B2) или R1 S2(X2)
Умножение	MR	RR	1C	R1,R2
Вычитание	S[1]	RX	5B	R1,D2(X2,B2) или R1,S2(X2)
Вычитание полуслова	SH[1]	RX	4B	R1,D2(X2,B2) или R1,S2(X2)
Вычитание кодов	SL[1]	RX	5F	R1,D2(X2,B2) или R1,S2(X2)
Вычитание кодов	SLR[1]	RR	1F	R1,R2
Вычитание	SR[1]	RR	1B	R1,R2

Таблица 1.2

Команды сдвига

Команда	Мнемонический код	Формат	Машинный код	Адреса: явные или неявные
Сдвиг влево	SLA[1]	RS	8B	R1,D2(B2) или R1,S2
Сдвиг влево двойной	SLDA[1]	RS	8F	R1,D2(B2) или R1,S2
Сдвиг влево двойной кодов	SLDL	RS	8D	R1,D2(B2) или R1,S2
Сдвиг влево кодов	SLL	RS	89	R1,D2(B2) или R1,S2
Сдвиг вправо	SRA[1]	RS	8A	R1,D2(B2) или R1,S2
Сдвиг вправо двойной	SRDA[1]	RS	8E	R1,D2(B2) или R1 S2
Сдвиг вправо двойной кодов	SRDL	RS	8C	R1,D2(B2) или R1 S2
Сдвиг вправо кодов	SRL	RS	88	R1,D2(B2) или R1,S2

Таблица 1.3

Команды пересылки

Команда	Мнемонический код	Формат	Машинный код	Адреса: явные или неявные
Прочитать символ	IC	RX	43	R1,D2(X2,B2) или R1,S2(X2)
Загрузка	L	RX	58	R1,D2(X2,B2) или R1,S2(X2)
Загрузка адреса	LA	RX	41	R1,D2(X2,B2) или R1,S2(X2)
Загрузка полуслова	LH	RX	48	R1,D2(X2,B2) или R1,S2(X2)
Загрузка групповая	LM	RS	98	R1,R3,D2(B2) или R1,R3,S2
Загрузка	LR	RR	18	R1,R2
Загрузка дополнения	LCR[1]	RR	13	R1,R2
Загрузка отрицательная	LNR[1]	RR	11	R1,R2
Загрузка и проверка	LTR[1]	RR	12	R1,R2
Загрузка положительная	LPR[1]	RR	10	R1,R2
Загрузка PSW	LPSW	SI	82	D1(B1) или S1
Пересылка	MVC	SS	D2	D1(L,B1),D2(B2) или S1(L),S2
Пересылка символа	MVI	SI	92	D1(B1),I2 или S1,I2
Пересылка цифр	MVN	SS	D1	D1(L,B1),D2(B2) или S1(L),S2
Пересылка со сдвигом	MVO	SS	F1	D1(L1,B1),D2(L2,B2) или S1(L1),S2(L2)
Пересылка зон	MVZ	SS	D3	D1(L,B1),D2(B2) или S1(L),S2
Запись в память	ST	RX	50	R1,D2(X2,B2) или R1,S2(X2)
Запись в память символа	STC	RX	42	R1,D2(X2,B2) или R1,S2(X2)
Запись в память полуслова	STH	RX	40	R1,D2(X2,B2) или R1,S2(X2)
Запись в память групповая	STM	RS	90	R1,R3,D2(B2) или R1,R3,S2

Таблица 1.4

Логические команды

Команда	Мнемонический код	Формат	Машинный код	Адреса: явные или неявные
Преобразование в двоичную	CVB	RX	4F	R1,D2(X2,B2) или R1,S2(X2)
Преобразование в десятичную	CVD	RX	4E	R1,D2(X2,B2) или R1,S2(X2)
И	N[1]	RX	54	R1,D2(X2,B2) или R1,S2(X2)
И	NC[1]	SS	D4	D1(L,B1),D2(B2) или S1(L),S2
И	NI[1]	SI	94	D1(B1),I2 или S1,I2
И	NR[1]	RR[1]	14	R1,R2
ИЛИ	O[1]	RX	56	R1,D2(X2,B2) или R1,S2(X2)
ИЛИ	OC[1]	SS	D6	D1(L,B1), D2(B2) или S1(L),S2
ИЛИ	OI[1]	SI	96	D1(B1),I2 или S1,I2
ИЛИ	OR[1]	RR	16	R1,R2
Упаковать	PACK	SS	F2	D1(L1,B1),D2(L2,B2) или S1(L1),S2(L2)
Перекодировать	TR	SS	DC	D1(L,B1),D2(B2) или S1(L),S2
Перекодировать и проверить	TRT[1]	SS	DD	D1(L,B1),D2(B2) или S1(L),S2
Распаковать	UNPK	SS	F3	D1(L1,B1),D2(L2,B2) или S1(L1),S2(L2)
Исключающее ИЛИ	X[1]	RX	57	R1,D2(X2,B2) или R1,S2(X2)
Исключающее ИЛИ	XC[1]	SS	D7	D1(L,B1),D2(B2) или S1(L),S2
Исключающее ИЛИ	XI[1]	SI	97	D1(B1),I2 или S1,I2
Исключающее ИЛИ	XR[1]	RR	17	R1,R2

Таблица 1.5

Команды проверки условий и передачи управления

Команда	Мнемонический код	Формат	Машинный код	Адреса: явные или неявные
Переход с возвратом	BAL	RX	45	R1,D2(X2,B2) или R1,S2(X2)
Переход с возвратом	BALR	RR	05	R1,R2
Условный переход	BC	RX	47	R1,D2(X2,B2) или R1,S2(X2)
Условный переход	BCR	RR	07	R1,R2
Переход по счетчику	BCT	RX	46	R1,D2(X2,B2) или R1,S2(X2)
Переход по счетчику	BCTR	RR	06	R1,R2
Переход по индексу «больше»	BXH	RS	86	R1,R3,D2(B2) или R1,R3,S2
Переход по индексу «меньше или равно»	BXLE	RS	87	R1,R3,D2(B2) или R1,R3,S2
Сравнение	C[1]	RX	59	R1,D2(X2,B2) или R1,S2(X2)
Сравнение полуслова	CH[1]	RX	49	R1,D2(X2,B2) или R1,S2(X2)
Сравнение кодов	CL[1]	RX	55	R1,D2(X2,B2) или R1,S2(X2)
Сравнение кодов	CLC[1]	SS	D5	D1(L,B1),D2(B2) или S1(L),S2
Сравнение кодов	CLI[1]	SI	95	D1(B1),I2 или S1,I2
Сравнение	CR[1]	RR	19	R1,R2
Выполнить	EX	RX	44	R1,D2(X2,B2) или R1,S2(X2)
Проверить по маске	TM[1]	SI	91	D1(B1),I2 или S1,I2
Проверить и установить	TS[1]	SI	93	D1(B1) или S1

Таблица 1.6

Привилегированные команды

Команда	Мнемонический код	Формат	Машинный код	Адреса: явные или неявные
Остановить ввод-вывод	HIO[1]	SI	9E	D1(B1) или S1
Прочитать ключ памяти	ISK	RR	09	R1,R2
Прямое чтение	RDD	SI	85	D1(B1),I2 или S1,I2
Начать ввод-вывод	SIO[1]	SI	9C	D1(B1) или S1
Установить маску программы	SPM	RR	04	R1
Установить ключ памяти	SSK	RR	08	R1,R2
Установить маску системы	SSM	SI	80	D1(B1) или S1
Обращение к Супервизору	SVC	RR	OA	I2
Опросить канал	TCH[1]	SI	9F	D1(B1) или S1
Опросить ввод-вывод	TIO[1]	SI	9D	D1(B1) или S1
Прямая запись	WRD	SI	84	D1(B1), I2 или S1,I2

Таблица 1.7

Команды десятичной арифметики

Команда	Мнемонический код	Формат	Машинный код	Адреса: явные или неявные
Сложение десятичное	AP[1]	SS	FA	D1(L1,B1),D2(L2,B2) или S1(L1),S2(L2)
Сравнение десятичное	CP[1]	SS	F9	D1(L1,B1),D2(L2,B2) или S1(L1),S2(L2)

Продолжение табл. 1.7

Команда	Мнемонический код	Формат	Машинный код	Адреса: явные или неявные
Деление десятичное	DP	SS	FD	D1(L1,B1),D2(L2,B2) или S1(L1),S2(L2)
Отредактировать	ED[1]	SS	DE	D1(L,B1),D2(B2) или S1(L),S2
Отредактировать и отметить	EDMK[1]	SS	DF	D1(L,B1),D2(B2) или S1(L),S2
Умножение десятичное	MP	SS	FC	D1(L1,B1),D2(L2,B2) или S1(L1),S2(L2)
Вычитание десятичное	SP[1]	SS	FB	D1(L1,B1),D2(L2,B2) или S1(L1),S2(L2)
Сложение с очисткой	ZAP[1]	SS	F8	D1(L1,B1),D2(L2,B2) или S1(L1),S2(L2)

Таблица 1.8

Команды арифметики с плавающей точкой

Команда	Мнемонический код	Формат	Машинный код	Адреса: явные или неявные
Сложение с нормализацией (длинное)	ADR[1]	RR	2A	R1,R2
Сложение с нормализацией (длинное)	AD[1]	RX	6A	R1,D2(X2,B2) или R1,S2(X2)
Сложение с нормализацией (короткое)	AER[1]	RR	3A	R1,R2
Сложение с нормализацией (короткое)	AE[1]	RX	7A	R1,D2(X2,B2) или R1,S2(X2)
Сложение без нормализации (длинное)	AWR[1]	RR	2E	R1,R2
Сложение без нормализации (длинное)	AW[1]	RX	6E	R1,D2(X2,B2) или R1,S2(X2)
Сложение без нормализации (короткое)	AUR[1]	RR	3E	R1,R2
Сложение без нормализации (короткое)	AU[1]	RX	7E	R1,D2(X2,B2) или R1,S2(X2)
Сложение с нормализацией (расширенные операнды)	AXR[1]	RR	36	R1,R2
Сравнение (длинное)	CDR[1]	RR	29	R1,R2
Сравнение (длинное)	CD[1]	RX	69	R1,D2(X2,B2) или R1,S2(X2)
Сравнение (короткое)	CER[1]	RR	39	R1,R2
Сравнение (короткое)	CE[1]	RX	79	R1,D2(X2,B2) или R1,S2(X2)
Деление (длинное)	DDR	RR	2D	R1,R2
Деление (длинное)	DD	RX	6D	R1,D2(X2,B2) или R1,S2(X2)
Деление (короткое)	DER	RR	3D	R1,R2
Деление (короткое)	DE	RX	7D	R1,D2(X2,B2) или R1,S2(X2)
Пополам (длинное)	HDR	RR	24	R1,R2
Пополам (короткое)	HER	RR	34	R1,R2
Загрузка и проверка (длинная)	LTDR[1]	RR	22	R1,R2
Загрузка и проверка (короткая)	LTER[1]	RR	32	R1,R2
Загрузка дополнения (длинная)	LCDR[1]	RR	23	R1,R2

Продолжение табл. 1.8

Команда	Мнемонический код	Формат	Машинный код	Адреса: явные или неявные
Загрузка дополнения (короткая)	LCER[1]	RR	33	R1,R2
Загрузка (длинная)	LDR	RR	28	R1,R2
Загрузка (длинная)	LD	RX	68	R1,D2(X2,B2) или R1,S2(X2)
Загрузка отрицательная (длинная)	LNDR[1]	RR	21	R1,R2
Загрузка отрицательная (короткая)	LNER[1]	RR	31	R1,R2
Загрузка положительная (длинная)	LPDR[1]	RR	20	R1,R2
Загрузка положительная (короткая)	LPER[1]	RR	30	R1,R2
Загрузка (короткая)	LER	RR	38	R1,R2
Загрузка (короткая)	LE	RX	78	R1,D2(X2,B2) или R1,S2(X2)
Загрузка с округлением (расширенный операнд)	LRDR	RR	25	R1,R2
Загрузка с округлением (длинный операнд)	LRER	RR	35	R1,R2
Умножение (длинное)	MDR	RR	2C	R1,R2
Умножение (длинное)	MD	RX	6C	R1,D2(X2,B2) или R1,S2(X2)
Умножение (короткое)	MER	RR	3C	R1,R2
Умножение (короткое)	ME	RX	7C	R1,D2(X2,B2) или R1,S2(X2)
Умножение (расширенные операнды)	MXR	RR	26	R1,R2
Умножение (преобразование длинного операнда в расширенный)	MXDR	RR	27	R1,R2
Умножение (преобразование длинного операнда в расширенный)	MXD	RX	67	R1,D2(X2,B2) или R1,S2(X2)
Запись в память (длинная)	STD	RX	60	R1,D2(X2,B2) или R1,S2(X2)
Запись в память (короткая)	STE	RX	70	R1,D2(X2,B2) или R1,S2(X2)
Вычитание с нормализацией (длинное)	SDR[1]	RR	2B	R1,R2
Вычитание с нормализацией (длинное)	SD[1]	RX	6B	R1,D2(X2,B2) или R1,S2(X2)
Вычитание с нормализацией (короткое)	SER[1]	RR	3B	R1,R2
Вычитание с нормализацией (короткое)	SE[1]	RX	7B	R1,D2(X2,B2) или R1,S2(X2)
Вычитание без нормализации (длинное)	SWR[1]	RR	2F	R1,R2
Вычитание без нормализации (длинное)	SW[1]	RX	6F	R1,D2(X2,B2) или R1,S2(X2)
Вычитание без нормализации (короткое)	SUR[1]	RR	3F	R1,R2
Вычитание без нормализации (короткое)	SU[1]	RX	7F	R1,D2(X2,B2) или R1,S2(X2)
Вычитание с нормализацией (расширенные операнды)	SXR[1]	RR	37	R1,R2

Таблица 1.9

Команды расширенной мнемоники

Команда	Мнемонический код	Маска	Формат	Адреса: явные или неявные
Безусловный переход	B	15	RX	D2(X2,B2) или S2(X2)
Безусловный переход	BR	15	RR	R2
Нет операции	NOP	0	RX	D2(X2,B2) или S2(X2)
Нет операции	NOPR	0	RR	R2
Переход по «больше»*	BH	2	RX	D2(X2,B2) или S2(X2)
Переход по «меньше»*	BL	4	RX	D2(X2,B2) или S2(X2)
Переход по «равно»*	BE	8	RX	D2(X2,B2) или S2(X2)
Переход по «не больше»*	BNH	13	RX	D2(X2,B2) или S2(X2)
Переход по «не меньше»*	BNL	11	RX	D2(X2,B2) или S2(X2)
Переход по «не равно»*	BNE	7	RX	D2(X2,B2) или S2(X2)
Переход по переполнению**	BO	1	RX	D2(X2,B2) или S2(X2)
Переход по знаку «+»**	BP	2	RX	D2(X2,B2) или S2(X2)
Переход по знаку «—»**	BM	4	RX	D2(X2,B2) или S2(X2)
Переход по «нулю»**	BZ	8	RX	D2(X2,B2) или S2(X2)
Переход, если нет переполнения**	BNO	14	RX	D2(X2,B2) или S2(X2)
Переход по «не +»**	BNP	13	RX	D2(X2,B2) или S2(X2)
Переход по «не —»**	BNM	11	RX	D2(X2,B2) или S2(X2)
Переход по «не нуль»**	BNZ	7	RX	D2(X2,B2) или S2(X2)
Переход, если все 1***	BO	1	RX	D2(X2,B2) или S2(X2)
Переход, если есть 1 и 0 ***	BM	4	RX	D2(X2,B2) или S2(X2)
Переход, если все 0***	BZ	8	RX	D2(X2,B2) или S2(X2)
Переход, если не все 1***	BNO	14	RX	D2(X2,B2) или S2(X2)
Переход, если все 1 или все 0***	BNM	11	RX	D2(X2,B2) или S2(X2)
Переход, если не все 0***	BNZ	7	RX	D2(X2,B2) или S2(X2)

* Используется после команд сравнения.
** Используется после арифметических команд.
*** Используется после команд ТМ.

В табл. 1.10 указаны признаки результатов, устанавливаемые машинными командами.

Таблица 1.10

Признаки результатов, устанавливаемые машинными командами

Команда	Значение разряда маски			
	8	4	2	1
	Код условия			
	* 0	1	2	3
	Команды с плавающей точкой			
Сложение, вычитание, загрузка	Нуль	Меньше нуля	Больше нуля	—
Сравнение (A : B)	Равно	A меньше	A больше	—
Загрузка отрицательная	Нуль	Меньше нуля	—	—
Загрузка положительная	Нуль	—	Больше нуля	—

Продолжение табл. 1.10

Команда	Значение разряда маски			
	8	4	2	1
	Код условия			
	0	1	2	3
	Команды с фиксированной точкой			
Сложение, вычитание	Нуль	Меньше нуля	Больше нуля	Переполнение
Сложение кодов	Нуль, нет переноса	Не нуль, нет переноса	Нуль, перенос	Не нуль, перенос
Сравнение (A : B)	Равно	A меньше	A больше	—
Загрузка и проверка, сдвиги	Нуль	Меньше нуля	Больше нуля	—
Загрузка дополнения	Нуль	Меньше нуля	Больше нуля	Переполнение
Загрузка отрицательная	Нуль	Меньше нуля	—	—
Загрузка положительная	Нуль	—	Больше нуля	Переполнение
Вычитание кодов	—	Не нуль, нет переноса	Нуль, перенос	Не нуль, перенос
	Десятичная арифметика			
Сложение, вычитание десятичное	Нуль	Меньше нуля	Больше нуля	Переполнение
Сравнение десятичное (A : B)	Равно	A меньше	A больше	—
Сложение с очисткой	Нуль	Меньше нуля	Больше нуля	Переполнение
	Десятичная арифметика			
И, ИЛИ, исключающее ИЛИ	Нуль	Не нуль	—	—
Сравнение логическое (A:B)	Равно	A меньше	A больше	—
ED, EDMK	Нуль	Меньше нуля	Больше нуля	—
TM	Все 0	0 и 1	—	Все 1
TRT	Нуль	Не закончена	Закончена	—
TS	Нуль	1	—	—
	Команды ввода-вывода			
HIO	Прерывание запомнено	CSW запомнено	Операция прекращена	Устройство, канал или подканал выключены
SIO	Успешное выполнение	CSW запомнено	Канал или подканал заняты	Устройство, канал или подканал выключены
TIO	Устройство, канал, подканал доступны	CSW запомнено	Канал или подканал заняты	Канал или подканал выключены
TCH	Канал доступен	Прерывание запомнено	Монопольный режим	Канал выключен

1.3. КОМАНДЫ ПЕРИФЕРИЙНЫХ УСТРОЙСТВ

Эти команды определяют операции ввода-вывода, выполняемые периферийными устройствами. Они специфичны для каждого устройства [1]. Для записи команд ввода-вывода на языке Ассемблера используется оператор CCW (см. 1.4.1).

1.4. КОМАНДЫ АССЕМБЛЕРА

1.4.1. ОБЩЕЕ ОПИСАНИЕ

Команды Ассемблера определяют действия, которые должен выполнить транслятор при трансляции программы. Ниже в алфавитном порядке приведены команды Ассемблера и для каждой из них указаны: код, ее назначение и формат (содержимое поля названия и поля операндов).

Код команды	Назначение и формат
CCW	Определяет команду ввода-вывода длиной восемь байт. Вызывает выравнивание на границу двойного слова. **Название:** любое символическое имя или пробел. **Операнды:** четыре операнда, разделенных запятыми. Первый операнд (абсолютное выражение) определяет однобайтовый код команды ввода-вывода, второй операнд — трехбайтовый адрес области ввода-вывода, третий операнд — байт флажков команды ввода-вывода. Он задается абсолютным выражением, значение которого должно быть кратно восьми. Четвертый операнд (абсолютное выражение) определяет двухбайтовую длину области ввода-вывода.
CNOP	Выполняет установку счетчика адреса на требуемое полуслово в слове или двойном слове. В пропускаемые байты записывается машинная команда NOPR. Если перед обработкой CNOP счетчик адреса находится на границе нечетного байта, то выполняется предварительное выравнивание на границу полуслова. **Название:** имя перехода или пробел. **Операнды:** два абсолютных выражения, разделенных запятыми. Первый операнд указывает номер байта в слове или двойном слове, на который должен быть установлен счетчик адреса. Его значением могут быть числа 0, 2, 4 и 6. Второй операнд указывает, находится ли байт в слове (если он равен 4) или в двойном слове (если он равен 8).
COM	Определяет начало или продолжение общей области. **Название:** имя перехода или пробел. **Операнды:** не используются.
COPY	Вызывает включение в исходный модуль книги из подбиблиотеки Ассемблера (имя ее — A) библиотеки исходных модулей. **Название:** пробел. **Операнды:** имя включаемой книги.
CSECT	Определяет начало или продолжение программной секции. **Название:** любое символическое имя (трактуется как имя программной секции). **Операнды:** не используются.
DC	Определяет константы требуемого типа. **Название:** любое символическое имя или пробел. **Операнды:** от 1 до 32 операндов, разделенных запятыми (в Ассемблере E только один операнд). Описание операндов см. в 1.4.2.
DROP	Указывает регистры базы, определенные ранее оператором USING, которые не будут использоваться в дальнейшем в качестве регистров базы. **Название:** имя перехода или пробел. **Операнды:** от 1 до 16 операндов, разделенных запятыми. Каждый операнд должен задаваться абсолютным выражением, определяющим общий регистр.

DS — Определяет рабочее поле памяти. **Название**: любое символическое имя или пробел. **Операнды**: от 1 до 32 операндов, разделенных запятыми (в Ассемблере E только один операнд). Операнд оператора DS записывается по тем же правилам, что и операнд оператора DC (см. 1.4.2), но поле константы в операторе DS может отсутствовать, а значение модификатора длины констант типа C и X не должно превышать 65 535 (в операторе DC не более 256).

DSECT — Определяет начало или продолжение фиктивной области. **Название**: простое символическое имя, параметр или соединение параметра с другими знаками (трактуется как имя фиктивной области). **Операнды**: не используются.

EJECT — Вызывает переход на новую страницу распечатки. **Название**: имя перехода или пробел. **Операнды**: не используются.

END — Определяет конец исходного модуля. **Название**: имя перехода или пробел. **Операнды**: переместимое выражение или пробел. Значение переместимого выражения указывает точку в программе, в которую должно передаваться управление после загрузки программы.

ENTRY — Определяет символические имена данного исходного модуля, которые могут использоваться в других отдельно транслируемых модулях (входные имена). **Название**: имя перехода или пробел. **Операнды**: одно или более переместимых имен, разделенных запятыми.

EQU — Простому символическому имени, записанному в поле названия, присваивает значение выражения, записанного в поле операндов. **Название**: простое символическое имя.Оно может представляться одним параметром или параметром, соединенным с другими знаками. **Операнды**: выражение. Простые имена, используемые в этом выражении, должны быть предварительно определены.

EXTRN — Определяет символические имена, которые используются в данном модуле, но определяются в других отдельно транслируемых модулях (внешние имена). **Название**: имя перехода или пробел. **Операнды**: одно или более простых имен, разделенных запятыми.

ICTL — Определяет формат вводимых операторов. **Название**: пробел. **Операнды**: от одного до трех десятичных чисел, разделенных запятыми. Первый операнд указывает колонку начала оператора, второй — колонку конца. Следующая колонка (за колонкой конца) считается колонкой указателя продолжения. Третий операнд указывает колонку начала продолжения оператора.

ISEQ — Определяет необходимость проверки последовательности карт исходного модуля, следующих за данным оператором ISEQ. Проверкой устанавливается, возрастает ли номер данной карты относительно номера предыдущей карты. Границы поля карты, в котором содержится номер, указываются операндами оператора. Оператор ISEQ без операндов вызывает прекращение проверки. **Название**: пробел. **Операнды**: два десятичных числа, разделенных запятыми, или пробел. Первый операнд указывает начало проверяемого поля, второй — конец.

LTORG — Определяет начало области литералов, в которую помещаются литералы, встречающиеся после предыдущего LTORG или после начала программы. По оператору LTORG происходит выравнивание на границу двойного слова. Если LTORG отсутствует, то все литералы программы располагаются после первой программной секции. **Название**: любое символическое имя или пробел. **Операнды**: не используются.

ORG — Присваивает счетчику адреса значение выражения, записанного в поле операндов. **Название**: имя перехода или пробел. **Операнды**: переместимое выражение или пробел. Простые имена в выражении должны быть предварительно определены. Если операндом является пробел, то счетчику адреса присваивается значение на единицу больше значения максимального адреса, отведенного для данной программной секции к моменту обработки оператора ORG.

PRINT Определяет содержание распечатки объектного модуля. **Название:** имя перехода или пробел. **Операнды**: от одного до трех слов, разделенных запятыми: ON (распечатка выводится) или OFF (распечатка не выводится), GEN (печатаются операторы, сгенерированные для макрокоманд) или NOGEN (не печатаются операторы, сгенерированные для макрокоманд), DATA (константы печатаются полностью) или NODATA (печатаются только восемь первых байт констант).

PUNCH Перфорирует на карту знаки, записанные в поле операндов (исключая граничные апострофы). **Название**: имя перехода или пробел. **Операнды**: от 1 до 80 знаков, заключенных в апострофы.

REPRO Перфорирует карту, содержимое которой задается следующим за REPRO оператором. **Название**: имя перехода или пробел. **Операнды**: не используются.

SPACE Вызывает пропуск строк в распечатке. Количество пропускаемых строк указывается операндом. **Название**: имя перехода или пробел. **Операнды**: десятичное число или пробел. Если операндом является пробел, то пропускается одна строка.

START Определяет первую программную секцию и указывает для нее первоначальное значение счетчика адреса. **Название**: любое символическое имя (трактуется как название программной секции) или пробел. **Операнды**: самоопределенный терм или пробел. Определяет значение счетчика адреса.

TITLE Идентифицирует распечатку и карты объектного модуля и вызывает переход на новую страницу распечатки. **Название**: пробел или комбинация букв и цифр длиной не более четырех знаков. Этот набор знаков печатается в заголовке каждой страницы распечатки и перфорируется в колонках 73—76 карт объектного модуля. **Операнды**: от 1 до 100 знаков, заключенных в апострофы. Эти знаки (исключая граничные апострофы) печатаются как заголовки страниц распечатки.

USING Определяет регистры базы. **Название**: имя перехода или пробел. **Операнды**: от 2 до 17 операндов, разделенных запятыми. Первый операнд (выражение V) определяет базовое значение для регистра базы, который указывается вторым операндом (абсолютное выражение). Все операнды, начиная со второго (абсолютные выражения), задают общие регистры, которые можно использовать в качестве регистров базы. Для регистра базы, задаваемого третьим операндом, базовым значением является V + 4096, для задаваемого четвертым операндом — V + 8192 и т. д.

1.4.2. КОМАНДА DC

Каждый операнд команды DC состоит из четырех полей, следующих друг за другом: кратности, типа, модификаторов, константы (или констант). Поле кратности и модификаторов можно опускать. Если необходимо, константы, создаваемые по оператору DC, выравниваются на нужную границу в соответствии с их типом (табл. 1.11).

Кратность указывает, сколько раз должны быть построены константа или константы, определяемые данным операндом. Если кратность равна нулю и отсутствует модификатор длины, то выполняется только выравнивание значения счетчика адреса на границу согласно типу константы, указанной в операнде. Кратность задается либо десятичным самоопределенным термом без знака, либо абсолютным положительным выражением, заключенным в скобки. Все простые имена в выражении должны быть предварительно определены.

Тип константы определяет ее машинный формат. Тип задается одной буквой (см. табл. 1.11).

В поле модификаторов записываются в следующем порядке: модификаторы длины, масштаба и порядка. Модификаторы можно опускать.

Модификатор длины указывает длину поля в байтах (для Ассемблера F можно указывать в битах), отводимого для константы. Присутствие модификатора длины означает, что длина константы задана явно. Если модификатор длины опущен, то считается, что длина константы задана неявно. Если длина задана неявно, то происходит выравнивание константы соответственно ее типу (см. табл. 1.11), при явном задании длины выравнивание константы не производится. Модификатор длины записывается как Ln, если длина задается в байтах, или как L.n, если длина задается в битах, где n — десятичный самоопределенный терм без знака или абсолютное положительное выражение, заключенное в скобки. Простые имена в этом выражении должны быть предварительно определены.

Модификатор масштаба можно использовать только для констант с фиксированной или плавающей точкой. Воздействие на них модификатора масштаба показано при описании этих констант. Модификатор масштаба записывается как Sn, где n — десятичный самоопределенный терм со знаком или без него либо абсолютное выражение, заключенное в скобки. Перед скобками можно записывать знак модификатора масштаба. Простые имена в выражении должны быть предварительно определены.

Модификатор порядка можно указывать только для констант с фиксированной или плавающей точкой. Этот модификатор указывает степень числа 10, на которое нужно умножить константу перед ее преобразованием в машинный формат. Модификатор записывается как En, где n — десятичный самоопределенный терм со знаком или без него либо абсолютное выражение, заключенное в скобки. Перед скобками можно записывать знак модификатора порядка. Простые имена в выражении должны быть предварительно определены.

В поле константы непосредственно записывается константа (или константы) (см. табл. 1.11). Константы типа C, X, B, P, Z, F, H, D, E должны быть заключены в апострофы, а адресные константы типа A, Y, S, V — в круглые скобки. Если в поле константы записывается несколько констант, то они отделяются друг от друга запятыми и все сразу заключаются в апострофы или скобки.

Константа типа C— знаковая. Каждый знак, используемый для ее представления, транслируется в один байт. Каждый апостроф или знак &, определяемые в константе как знаки, должны представляться двумя апострофами или двумя знаками &.

Константа типа X — шестнадцатеричная. Каждые две цифры транслируются в один байт.

Константа типа B —двоичная. Каждые восемь цифр транслируются в один байт.

Константы типа P, Z — десятичные. Для указания дробной части используется точка. Знак плюс транслируется в код C, а знак минус — в код D.

Таблица 1.11

Типы констант

Константа	Тип	Задается	Неявная длина, байт	Максимальная длина	Выравнивание	Отбрасываемая часть	Число констант в операнде
Знаковая	C	знаками ДКОИ	По требованию	256	Байт	Правая	Одна
Шестнадцатеричная	X	шестнадцатеричными цифрами	То же	256	»	Левая	»
Двоичная	B	двоичными цифрами	»	256	»	»	»
Упакованная десятичная	P	десятичными цифрами, может указываться знак константы	»	16	»	»	Несколько
Распакованная десятичная	Z	десятичными цифрами, может указываться знак константы	»	16	»	»	»
С фиксированной точкой	F	десятичными цифрами, может указываться знак константы	4	8	Слово	»	»
С фиксированной точкой	H	десятичными цифрами, может указываться знак константы	2	8	Полуслово	»	»
С плавающей точкой	D	десятичными цифрами, может указываться знак константы	8	8	Двойное слово	Правая	»
С плавающей точкой	E	десятичными цифрами, может указываться знак константы	4	8	Слово	»	»
Действительная адресная	A	выражением (значение от -2^{31} до $2^{31}-1$)	4	2	»	Левая	»
Действительная адресная	Y	выражением (значение от -2^{15} до $2^{15}-1$)	2	2	Полуслово	»	»
Явная адресная	S	абсолютным или переместимым выражением либо двумя абсолютными выражениями в форме: выражение (выражение)	2	2	»	»	»
Внешняя адресная	V	переместимым именем	4	4	Слово	»	»

Константы типа F, H — константы с фиксированной точкой. Они записываются как десятичное число со знаком или без знака, за которым может следовать порядок. Дробная часть числа указывается точкой. Порядок записывается как Ep, где p — десятичный самоопределенный терм со знаком или без знака. Порядок указывает степень числа 10, на которое умножается константа до ее преобразования в машинный формат. Транслятор переводит константу в двоичную систему счисления и выполняет масштабирование. Модификатор масштаба указывает количество двоичных позиций дробной части, которые присоединяются к целой части двоичного числа, если масштаб положительный, или исключаются из целой части двоичного числа, если масштаб отрицательный. После масштабирования число округляется.

Константы типа D, E — константы с плавающей точкой. Они записываются так же, как константы с фиксированной точкой. Транслятор представляет константу с плавающей точкой в виде шестнадцатеричной нормализованной дроби и выполняет масштабирование. Модификатор масштаба для этих констант может быть только положительным. Он указывает число шестнадцатеричных позиций, на которые должна быть сдвинута вправо шестнадцатеричная дробь. Порядок константы при масштабировании корректируется.

Константы типа A, Y — действительные адресные константы. Они задаются абсолютным или переместимым выражением. Значение счетчика адреса, используемого в адресных константах, равно адресу первого байта, занимаемого константой в памяти. Если в одном операнде имеется несколько констант, то значение счетчика изменяется от константы к константе. Если указан коэффициент кратности и в операнде используется счетчик адреса, то его значение модифицируется для каждой копии операнда. Это верно не только для действительных адресных констант (типа A и Y), но и для явной адресной константы (типа S). Если адресная константа задается литералом, то значение счетчика адреса, использующегося в литерале, равно адресу команды, в которой записан этот литерал.

Константа типа S — явная адресная. Она определяет адрес памяти в виде регистра базы и смещения. В отношении значения счетчика адреса, используемого в константах типа S, верны замечания, сказанные о счетчике адреса для констант типа A и Y.

Константа типа V — внешняя адресная. Она определяется одним переместимым именем и предназначается для резервирования места для адреса внешнего имени, используемого для связи с другими программами.

1.5. МАКРОСРЕДСТВА АССЕМБЛЕРА

Макросредства позволяют расширять язык Ассемблера макрокомандами. По макрокоманде транслятор Ассемблера генерирует из макроопределения некоторую последовательность операторов (макрорасширение) и вставляет ее в исходную программу. Ниже приведены макросредства, которые можно использовать при составлении макроопределений.

1.5.1. ПАРАМЕТРЫ

Параметр записывается как простое символическое имя длиной от 1 до 7 букв или цифр, которому предшествует знак & (например, &KIP). Параметры заменяются значениями во время генерации и позволяют создавать из макроопределений такие макрорасширения, которые настроены на конкретные операнды макрокоманды. Параметры делятся на постоянные, системные и переменные.

Постоянные параметры можно использовать только в макроопределениях. Названия постоянных параметров определяются прототипом; значения, которыми они заменяются, задаются операндами макрокоманды.

Системные параметры можно употреблять только в макроопределениях. Значения системным параметрам присваиваются транслятором Ассемблера. Имеются следующие системные параметры: &SYSNDX, &SYSECT, &SYSLIST, &SYSPARM.

Параметр & SYSNDX обычно используется для обеспечения уникальности простых имен в макрорасширениях, генерируемых из одного макроопределения, для макрокоманд с различными операндами. Значением параметра &SYSNDX является порядковый номер обрабатываемой макрокоманды.

Параметр &SYSECT предназначен для восстановления программной секции или фиктивной области, в которой находится макрокоманда. Значением этого параметра является имя последнего оператора CSECT, START или DSECT, который записан перед макрокомандой.

Параметр &SYSLIST позволяет обращаться к операндам макрокоманды, не используя их названий. Значением параметра &SYSLIST(n) (где n — арифметическое выражение) является значение операнда макрокоманды с номером n.Если операндом n макрокоманды является список, то запись &SYSLIST(n,m) позволяет обратиться к элементу m списка, соответствующего операнду n.

С помощью параметра &SYSPARM программист может использовать в процессе генерации знаковую строку, указанную операндом SYSPARM оператора OPTION (см. 2.2.5). Например, если в операторе OPTION задан операнд SYSPARM = 'XL400' и в макроопределении имеется оператор DS &SYSPARM, то во время генерации этот оператор будет преобразован в оператор DS XL400.

Переменные параметры можно использовать как в макроопределениях, так и в исходных модулях. По типу значений переменные параметры делятся на арифметические, знаковые и логические. Значением арифметического параметра может быть целое число от -2^{31} до $2^{31} - 1$, значением знакового параметра — строка знаков от 0 до 8 знаков, значением логического параметра — логическое значение 0 («ложь») или 1 («истина»).

По месту действия переменные параметры делятся на локальные и глобальные. Значение локального параметра действует только в одном макроопределении или вне макроопределения (где он определен). Одно значение глобального параметра может использоваться в разных макроопределениях и вне макроопределений (там, где он определен). Гло-

бальные арифметические, знаковые и логические параметры определяются соответственно операторами GBLA, GBLC и GBLB, а локальные — операторами LCLA, LCLC, LCLB. Значения переменных параметров можно изменять операторами SETA, SETB, SETC. Переменные параметры могут быть индексированными. Под одним названием индексируемого параметра выступает множество независимых значений, каждое из которых соответствует определенному индексу этого параметра.

1.5.2. ИМЯ ПЕРЕХОДА

Имя перехода (один из типов символического имени) используется в качестве метки оператора, на который необходимо выполнить генераторный переход. Имя перехода записывается как точка, за которой следует от одной до семи букв или цифр, причем первым знаком за точкой должна быть буква (например, .KIB4).

1.5.3. ХАРАКТЕРИСТИКА

Характеристикой является знаковое или арифметическое значение, выражающее некоторые свойства операнда макрокоманды или символического имени. В макроопределении можно использовать характеристики только постоянных параметров, а вне макроопределений — характеристики только простых символических имен. К характеристикам операндов макрокоманды можно обращаться, используя название соответствующего постоянного параметра или с помощью системного параметра &SYSLIST. Характеристика количества N'&SYSLIST определяет число всех операндов макрокоманды. Характеристика записывается следующим образом: обозначение характеристики (табл. 1.12), за которым следует элемент, характеристика которого используется. Характеристики употребляются в качестве термов выражений макросредств.

Таблица 1.12

Обозначения характеристик

Название	Использование	Обозначение
Характеристика типа	В любом месте программы	T'
Характеристика длины	То же	L'
Характеристика масштаба	»	S'
Характеристика целой части	»	I'
Характеристика количества знаков	Только в макроопределении	K'
Характеристика количества операндов	То же	N'

Значением характеристики типа является буква (табл. 1.13).

Таблица 1.13

Значения характеристики типа

Элемент языка	Значение
Константа типа A. Неявная длина	A
Константа типа B	B
Константа типа C	C
Константа типа D. Неявная длина	D
Константа типа E. Неявная длина	E
Константа типа F. Неявная длина	F
Константа с фиксированной точкой. Явная длина	G
Константа типа H. Неявная длина	H
Константа с плавающей точкой. Явная длина	K
Константа типа P	P
Адресная константа. Явная длина	R
Константа типа S. Неявная длина	S
Константа типа V. Неявная длина	V
Константа типа X	X
Константа типа Y. Неявная длина	Y
Константа типа Z	Z
Машинная команда	I
Команда START, CSECT или DSECT	J
Макрокоманда	M
Операнд команды EXTRN	T
Команда CCW	W
Операнд макрокоманды — самоопределенный терм	N
Операнд макрокоманды опущен	O
Остальные элементы, в том числе оператор DC, кратность или модификаторы которого заданы выражением или параметром	U

Характеристика длины действительна для всех элементов, кроме тех, характеристикой типа которых являются буквы M, N, O, T или U. Значение характеристики длины равно длине (в байтах) области памяти, именуемой простым именем.

Характеристика масштаба действительна для десятичных констант, констант с плавающей и фиксированной точкой. Значением характеристики масштаба для констант с фиксированной и плавающей точкой является значение модификатора масштаба, для десятичных констант — число десятичных цифр справа от десятичной точки.

Характеристика целой части действительна для десятичных констант и констант с фиксированной и плавающей точкой. Значение характеристики целой части равно: для констант типа F и H числу двоичных цифр перед точкой ($I' = 8 \times L' - S' - 1$); для констант типа D и E числу шестнадцатеричных цифр в мантиссе без первых нулей ($I' = 2 \times (L' - 1) - S'$); для десятичных констант числу десятичных цифр перед десятичной точкой (для типа P: $I' = 2 \times L' - S' - 1$, для типа Z : $I' = L' - S'$).

Характеристика количества знаков может использоваться только в макроопределении. Она равна количеству знаков в операнде макрокоманды.

Характеристика количества операндов может использоваться только в макроопределении. Она равна количеству элементов в списке, соответствующем постоянному параметру. Если операнд макрокоманды не является списком, то значение этой характеристики принимается равной 1, а если операнд опущен, то значением характеристики считается 0. Как особый случай N'&SYSLIST равна количеству операндов в макрокоманде.

1.5.4. ЗНАКОВАЯ СТРОКА

Знаковой строкой, которая используется как терм в знаковом выражении, называется любой набор знаков, заключенный в апострофы. Апостроф или знак &, которые входят в этот набор знаков, должны изображаться двумя апострофами или двумя знаками &. В значение знаковой строки из каждых двойных апострофов попадает один апостроф, а из двойных знаков & попадут оба знака &.

1.5.5. ПОДСТРОКА ЗНАКОВ

Подстрока знаков записывается как знаковая строка, непосредственно за которой следуют два арифметических выражения, заключенные в скобки и разделенные запятой. Значение первого выражения указывает номер знака исходной знаковой строки, начиная с которого выделяется подстрока. Значение второго выражения указывает количество знаков, входящих в подстроку знаков. Подстрока знаков используется как терм в знаковом выражении.

1.5.6. ОТНОШЕНИЕ

Арифметическое отношение используется как терм в логическом выражении. Оно состоит из двух арифметических выражений, связанных одной из операций отношения: EQ (равно), NE (не равно), LT (меньше), G T (больше), LE (меньше или равно), GE (больше или равно).

Знаковое отношение — это два знаковых выражения, связанных между собой одной из операций отношения. Знаковое отношение используется в качестве терма в логическом выражении.

1.5.7. ВЫРАЖЕНИЯ

Арифметическое выражение употребляется в операнде операторов SETA, ACTR, SETB, AIF, в обозначении индекса переменного параметра, списка и параметра &SYSLIST, а также в обозначении подстроки знаков. В качестве термов выступают: самоопределенный терм; характеристики длины, масштаба, целой части, количества знаков и количества операндов; постоянный параметр, если ему соответст-

вует самоопределенный терм; переменный параметр (значением знакового параметра должны быть только десятичные цифры); параметры &SYSLIST и &SYSNDX. В качестве операций используются сложение (+), вычитание (—), умножение (*), деление (/). Допустимы круглые скобки. Значением выражения является арифметическое значение в пределах от -2^{31} до $2^{31} - 1$.

Знаковое выражение используется в операнде операторов SETB, AIF. В качестве термов выступают: знаковая строка, подстрока знаков, характеристика типа (должна быть единственным термом в выражении). Операцией является соединение (·), а значением — последовательность знаков длиной до 127 или до 255 знаков (см. 1.6).

Логическое выражение используется в операнде операторов SETB и AIF. В качестве термов применяются: логический параметр, арифметическое отношение, знаковое отношение, явное логическое значение 0 или 1, заключенное в скобки. В качестве операций допустимы AND (конъюнкция), OR (дизъюнкция), NOT (отрицание). Значением является логическое значение 1 («истина») или 0 («ложь»).

1.5.8. ОПЕРАТОРЫ МАКРОСРЕДСТВ

Оператор	Назначение и формат
ACTR	Определяет максимальное количество переходов по операторам AIF и AGO в макроопределении или в основной части программы (вне макроопределений). При отсутствии ACTR это количество переходов принимается равным 150 для Ассемблера E и 4096 для Ассемблера F. **Название:** пробел. **Операнды:** арифметическое выражение. Значение выражения указывает максимальное количество переходов.
AGO	Выполняет генераторный безусловный переход, т. е. обработка продолжается с оператора, который назван именем перехода, стоящим в поле операндов AGO. **Название:** имя перехода или пробел. **Операнды:** имя перехода.
AIF	Выполняет условный генераторный переход. **Название:** имя перехода или пробел. **Операнды:** логическое выражение, заключенное в скобки, за которым следует имя перехода. Проверяет значение логического выражения, записанного в поле операндов. Если значением является логическое значение 1 («истина»), то последовательность обработки операторов нарушается, и обработка продолжается с оператора, который назван именем перехода, стоящим в поле операндов AIF. Если значением является логическое значение 0 («ложь»), то обработка продолжается с оператора, стоящего после оператора AIF.
ANOP	По оператору ANOP никакого действия не выполняется. Этот оператор используется для того, чтобы можно было выполнить генераторный переход на тот оператор, в поле названия которого невозможно записать имя перехода. **Название:** имя перехода. **Операнды:** не используются.
GBLA, GBLB, GBLC	Определяют арифметические (GBLA), логические (GBLB) и знаковые (GBLC) глобальные переменные параметры. **Название:** пробел. **Операнды:** один или более переменных параметров, разделенных запятыми. Если параметр определяется индексируемым, то за названием параметра должен следовать заключенный в скобки индекс. Индекс в операторах GBLA, GBLB, GBLC, LCLA, LCLB, LCLC задается десятичным числом, а в остальных операторах — арифметическим выражением, не превышающим 255. Первый оператор GBLA, GBLB, GBLC устанавливает начальные значения для

переменных параметров: для арифметических — целое число 0, для логических — логическое значение 0 («ложь»), для знаковых — знаковую строку нулевой длины (т. е. отсутствие знаков).

LCLA,LCLB, LCLC — Определяют арифметические (LCLA), логические (LCLB) и знаковые (LCLC) локальные переменные параметры. **Название**: пробел. **Операнды**: один или более переменных параметров, разделенных запятыми. Локальные переменные параметры могут быть индексируемыми. Правила использования индексированных локальных и глобальных параметров совпадают (см. описание операторов GBLA, GBLB, GBLC). По операторам LCLA, LCLB, LCLC локальным параметрам устанавливаются первоначальные значения, совпадающие с первоначальными значениями глобальных параметров.

MACRO — Определяет начало макроопределения. **Название**: пробел. **Операнды**: не используются.

Макрокоманда — Определяет макроопределение, из которого в исходную программу генерируется макрорасширение, и значения, которыми заменяются постоянные параметры макроопределения. **Название**: любое символическое имя или пробел. **Операнды**: операнды, разделенные запятыми. Операндом является значение, которым заменяется постоянный параметр, или ключевое слово (название параметра без знака &), за которым следуют знак равенства и значение, заменяющее постоянный параметр. Операндом макрокоманды может быть список. Список записывается как один или несколько операндов (элементов списка), разделенных запятыми и заключенных в скобки. Если значением параметра (например, &PS1) является список, то тогда параметр &PS1 заменяется при генерации всеми знаками списка (включая скобки, в которые он заключен), а &PS1(m) заменяется элементом m этого списка.

MEND — Определяет конец макроопределения. **Название**: имя перехода или пробел. **Операнды**: не используются.

MEXIT — Прекращает генерацию макроопределения. **Название**: имя перехода или пробел. **Операнды**: не используются.

MNOTE — Порождает сообщение о ходе генерации макроопределения. **Название**: имя перехода или пробел. **Операнды**: два операнда, разделенные запятой. Первый операнд (десятичное число от 0 до 255 или знак *) указывает степень серьезности ошибки. Этот операнд может опускаться. Второй операнд определяет текст сообщения. Этим операндом может быть любой набор знаков (текст сообщения), заключенный в апострофы.

Прототип — Определяет код операции макрокоманды и названия постоянных параметров, которые используются в данном макроопределении. **Название**: постоянный параметр или пробел. **Операнды**: операнды, разделенные запятыми. Операндом является собственно постоянный параметр или постоянный параметр, за которым следуют знак равенства и стандартное значение параметра. Во время генерации постоянные параметры заменяются значениями, соответствующими им в макрокоманде.

SETA — Арифметическому параметру из поля названия присваивает значение арифметического выражения, записанного в поле операндов. **Название**: арифметический переменный параметр. **Операнды**: арифметическое выражение.

SETB — Логическому параметру из поля названия присваивает значение логического выражения, записанного в поле операндов. **Название**: логический переменный параметр. **Операнды**: логическое выражение.

SETC — Знаковому параметру из поля названия присваивает значение знакового выражения, записанного в поле операндов. **Название**: знаковый параметр. **Операнды**: знаковое выражение.

1.6. ОТЛИЧИЯ МЕЖДУ ЯЗЫКАМИ АССЕМБЛЕРОВ

Основные отличия приведены в табл. 1.14.

Таблица 1.14

Основные отличия между языками Ассемблеров

Возможности языка	ДОС ЕС		ОС
	Ассемблер E	Ассемблер F	Ассемблер F
Максимальное количество строк продолжения оператора (исключая прототип и макрокоманду)	1	2	2
Максимальное количество операндов в операторах DC и DS	1	32	32
Модификатор длины, бит	Не допускается	Допускается	Допускается
Константы типа L и Q	Не допускаются	Не допускаются	Допускаются
Максимальное количество операндов в макрокоманде и прототипе	100	200	200
Максимальное количество знаков в операнде макрокоманды	127	255	255
Максимальная длина значения знакового выражения, знаков	127	255	255
Стандартное количество переходов по операторам AIF и AGO	150	4096	4096
Операторы DXD, CXD, WXTRN, OPSYN	Не допускаются	Не допускаются	Допускаются
Минимальный объем памяти, необходимый для работы транслятора, Кбайт	14	44	44

ГЛАВА 2

УПРАВЛЯЮЩАЯ ПРОГРАММА

Основу операционной системы составляет управляющая программа, которая предназначена для автоматизации управления вычислительным процессом. Основные функции управляющей программы — подготовка операционной системы к работе, управление выполнением заданий, управление распределением и использованием ресурсов системы, связь с оператором и управление процедурами ввода-вывода. Структурно управляющая программа состоит из трех частей, выполняющих свои функции на разных этапах вычислительного процесса: программы Начальная загрузка, средств управления заданиями (программы: Управление заданиями, Инициатор одиночных программ и Внимание) и программы Супервизор.

2.1. ПОДГОТОВКА СИСТЕМЫ К РАБОТЕ

2.1.1. ПРОЦЕДУРА НАЧАЛЬНОЙ ЗАГРУЗКИ

Операционная система подготавливается к функционированию на ЭВМ совместными усилиями оператора и программы Начальная загрузка (рис. 2.1).

Функции оператора:

- Установка резидентного пакета и приведение соответствующего накопителя в состояние готовности.
- Выбор устройства связи с оператором и приведение его в состояние готовности.
- Манипуляции с кнопками ЗАГРУЗКА, ПРЕРЫВАНИЕ.
- Ввод ответов на сообщения программы Начальная загрузка.

Функции системы:

- Очистка (заполнение нулями) памяти машины.
- Загрузка в память резидентной части Супервизора (ядра Супервизора).
- Ввод и обработка директив начальной загрузки, предназначенных для настройки ядра на конкретную конфигурацию машины, и установки даты и времени дня.
- Сообщение оператору о завершении начальной загрузки.
- Вызов в фоновый раздел программы Управление заданиями и передача ей управления.

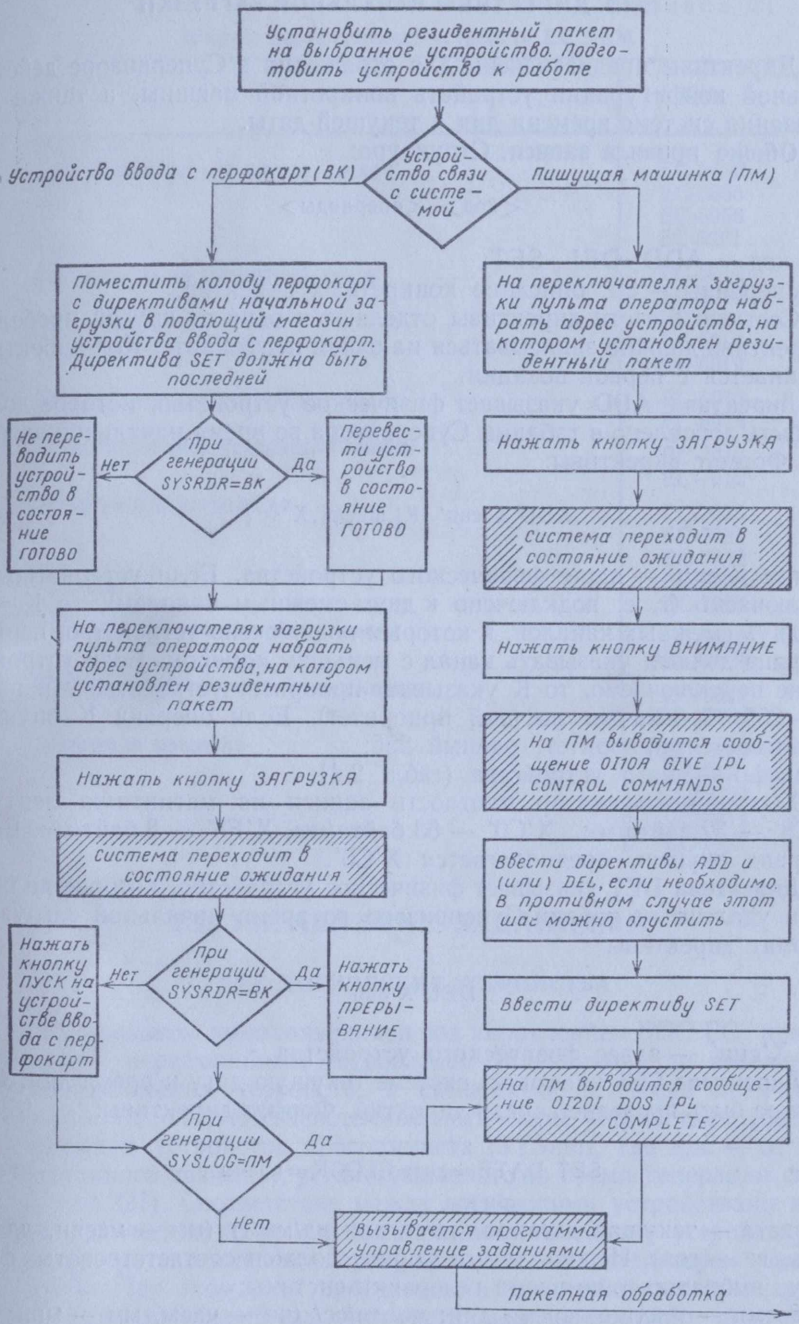

Рис. 2.1. Схема процедуры начальной загрузки (заштрихованными блоками изображены функции системы)

2.1.2. ДИРЕКТИВЫ НАЧАЛЬНОЙ ЗАГРУЗКИ

Директивы предназначены для отражения в Супервизоре действительной конфигурации устройств конкретной машины, а также для указания системе времени дня и текущей даты.

Общие правила записи. Структура:

<код> <операнды>

где код — ADD, DEL, SET;

операнды — см. описание конкретных директив.

Составные части директивы отделяются одна от другой пробелом. Директива должна записываться на одной перфокарте. Код директивы начинается с первой позиции.

Директива ADD указывает физическое устройство, которое должно быть добавлено в таблицы Супервизора во время начальной загрузки. Формат директивы:

ADD X'cuu' [K],шифр[,X'ss']

где X'cuu' — адрес физического устройства. Если устройство переключаемо (т. е. подключено к двум смежным каналам), то K = s. Из двух смежных каналов, к которым подключено устройство, первый операнд должен указывать канал с меньшим номером. Если устройство не переключаемо, то K указывает приоритет устройства в канале: 0 — 255 (0 означает высший приоритет). Если операнд K опущен, принимается приоритет, равный 255;

шифр — шифр устройства (табл. 2.1);

X'ss' — модификатор плотности записи на магнитную ленту: X'C8' — 32 байта/мм, X'C0' — 63 байта/мм, X'F8' — 8 байт/мм. Если операнд опущен, предполагается X'C0'.

Директива DEL указывает физическое устройство, которое должно быть удалено из таблиц Супервизора во время начальной загрузки. Формат директивы

DEL X'cuu'

где X'cuu' — адрес физического устройства.

Директива SET указывает системе текущую дату и время дня. Она должна быть последней среди директив. Формат директивы:

SET DATE=дата,CLOCK=время

где дата — текущая дата: мм/дд/гг или дд/мм/гг (мм — месяц, дд — день, гг — год). Используемый формат должен соответствовать формату, выбранному во время генерации системы;

время — текущее время дня: чч/мм/сс/ (чч — часы, мм — минуты, сс — секунды). Операнд требуется только в том случае, если использование таймера было указано во время генерации системы.

Таблица 2.1

Шифры периферийных устройств ЕС ЭВМ

Устройство	Шифр
Накопитель на магнитных дисках	ЕС-5052 ЕС-5055 ЕС-5056 ЕС-5061
Накопитель на магнитной ленте	ЕС-5010 ЕС-5012 ЕС-5016 ЕС-5017 ЕС-5019
Перфокарточное устройство ввода	ЕС-6012 ЕС-6016
Перфокарточное устройство вывода	ЕС-7010 ЕС-7012
Печатающее устройство	ЕС-7030 ЕС-7031 ЕС-7032 ЕС-7033 ЕС-7035
Перфоленточное устройство ввода	ЕС-6022 ЕС-7902R
Перфоленточное устройство вывода	ЕС-7022 ЕС-7902P
Пишущая машинка	ЕС-7070 ЕС-7073 ЕС-7073А ЕС-7074 ЕС-7077

2.2. УПРАВЛЕНИЕ ЗАДАНИЯМИ

2.2.1. ЛОГИЧЕСКИЕ УСТРОЙСТВА

В программах, выполняющихся под управлением ДОС ЕС, при обращении к периферийным устройствам указываются не абсолютные адреса физических устройств, а символические имена логических устройств. Используются системные логические устройства (табл. 2.2) и логические устройства программиста (SYSnnn, где nnn — от 0 до максимального значения, устанавливаемого во время генерации системы — max 221). Соответствие между логическими устройствами и адресами физических устройств (назначение логическому устройству физического) устанавливается программистом или оператором машины в задании. При этом если программиста удовлетворяют стандартные назначения периферийных устройств для каких-либо логических устройств, то в задании можно не указывать назначения для этих устройств.

Таблица 2.2

Системные логические устройства

Имя логического устройства	Имя системного файла	Назначение	Тип физического устройства
SYSRDR	IJSYSIN	Для ввода управляющих операторов задания	Перфокарточное устройство ввода, накопитель на магнитной ленте, накопитель на магнитных дисках
SYSIPT	IJSYSIN	Для ввода потока данных (системный ввод)	Перфокарточное устройство ввода, накопитель на магнитной ленте, накопитель на магнитных дисках
SYSPCH	IJSYSPH	Для вывода информации на перфокарты (системный перфоратор)	Перфокарточное устройство вывода, накопитель на магнитной ленте, накопитель на магнитных дисках
SYSLST	IJSYSLS	Для вывода информации на печать (системная печать)	Печатающее устройство, накопитель на магнитной ленте, накопитель на магнитных дисках
SYSLOG	—	Для ввода директив и вывода сообщений (связь с оператором)	Пультовая пишущая машинка, печатающее устройство
SYSLNK	IJSYSLN	Для ввода данных для Редактора	Накопитель на магнитных дисках
SYSCLB	IJSYSCL	Личная библиотека абсолютных модулей	Накопитель на магнитных дисках
SYSRLB	IJSYSRL	Личная библиотека объектных модулей	Накопитель на магнитных дисках
SYSSLB	IJSYSSL	Личная библиотека исходных модулей	Накопитель на магнитных дисках
SYSREC	IJSYSRC	Регистрационный файл, предназначенный для сбора статистики о машинных ошибках, ошибках каналов, интерфейса и устройств ввода-вывода	Накопитель на магнитных дисках

2.2.2. ПАКЕТНАЯ ОБРАБОТКА

Запуск. Пакетная обработка в фоновом разделе инициируется автоматически системой после завершения процедуры начальной загрузки. Для того чтобы пакетная обработка могла выполняться в разделе переднего плана, необходимо соблюдать следующие условия:

в системе предусмотреть пакетную обработку в разделах переднего плана;

разделу выделить не менее 10 Кбайт основной памяти;

разделу выделить необходимые физические устройства для используемых логических устройств.

Для запуска пакетной обработки в разделе переднего плана оператор должен установить связь с программой Внимание и ввести директиву BATCH с параметром, определяющим нужный раздел. После этого оператор может вводить с пишущей машинки любые директивы программы Управление заданиями. После ввода директивы КТ (см. 2.2.5) начинается обработка первого задания входного потока раздела переднего плана.

Поддержка. При пакетной обработке задания выбираются из пакета и обрабатываются одно за другим до тех пор, пока существует входной поток на устройстве SYSRDR. Идентификация конца входного потока на SYSRDR зависит от типа физического устройства, назначенного для SYSRDR. Если для него назначено устройство ввода с перфокарт, то конец входного потока определяется в том случае, когда в подающем магазине нет перфокарт и нажата кнопка КОНЕЦ КАРТОТЕКИ. Если для SYSRDR назначена магнитная лента, то конец входного потока определяется при считывании ленточной марки, если назначены диски, то при считывании записи «Конец файла» (запись с нулевой длиной данных).

При распознавании конца входного потока заданий на SYSRDR Управление заданиями выводит сообщение

```
1C00A ATTN cuu
```

и устанавливает связь с оператором для получения дальнейших директив. Чтобы начать обработку нового пакета заданий, оператор должен заправить карты нового пакета и ввести с устройства SYSLOG директиву КТ. Если оператор хочет перейти к обработке пакета, находящегося на другом устройстве, ему достаточно после названного сообщения ввести директиву нового назначения для SYSRDR.

Временная приостановка. Пакетная обработка в любом разделе может быть приостановлена директивой Управления заданиями STOP. Программа Управление заданиями остается в разделе до тех пор, пока не будет возобновлена пакетная обработка в этом разделе по директиве START или BATCH программы Внимание.

Прекращение. В фоновом разделе пакетная обработка не может быть прекращена оператором.

В разделах переднего плана пакетная обработка прекращается (раздел при этом освобождается) по директиве оператора UNBATCH, которая может быть введена с пишущей машинки в момент, когда уста-

новлена связь программы Управление заданиями с оператором. Такая связь устанавливается при распознавании конца входного потока заданий на SYSRDR. Кроме того, связь с оператором может быть установлена с помощью управляющего оператора PAUSE, вводимого последним в пакете заданий, или с помощью директивы PAUSE с параметром EOJ, введенной с помощью программы Внимание во время выполнения задания в данном разделе.

2.2.3. ВЫПОЛНЕНИЕ ОДИНОЧНОЙ ПРОГРАММЫ

Запуск осуществляется по инициативе оператора путем выдачи программе Внимание директивы START.

По директиве START F1 (или START F2) передается управление Инициатору одиночных программ, который настраивается на принятие директив оператора с пишущей машинки. Затем оператор может набрать директиву READ, указывающую Инициатору одиночных программ устройство ввода с перфокарт, с которого должны быть введены все остальные директивы. Это устройство к моменту запуска одиночной программы должно быть свободно и не назначено ни одному логическому устройству в других разделах.

С помощью директив Инициатора одиночных программ выполняется назначение периферийных устройств и сообщается системе информация о метках. Последней директивой должна быть директива EXEC. В результате ее обработки управление передается одиночной программе.

Завершение. Одиночная программа завершается по концу выполнения программы, в результате возникновения ошибочных ситуаций либо по требованию оператора. Оператор извещается об окончании выполнения одиночной программы, и раздел освобождается; если не установлен режим HOLD, то отменяются все назначения периферийных устройств.

2.2.4. ОРГАНИЗАЦИЯ СВЯЗИ ОПЕРАТОРА С СИСТЕМОЙ

Для организации экстренной связи оператора с системой предназначена программа Внимание. Связь устанавливается через пишущую машинку, назначенную для SYSLOG, путем нажатия на ней кнопки ВНИМАНИЕ. После вывода сообщения

1160A READY FOR COMMUNICATIONS

оператор может вводить с пишущей машинки директивы, которые принимаются и обрабатываются программой Внимание. Оператор сообщает системе о конце связи, вводя директиву KT.

С помощью директив программы Внимание оператор может:

запустить работу в разделах переднего плана;

перераспределить память между разделами;

прекратить выполнение задания в любом из разделов;

приостановить пакетную обработку заданий, запросив паузу после окончания выполнения текущего шага задания или задания;

указать, какой из разделов может использовать таймер.

Директивы программы Внимание реализуют функции, выполнение которых может потребоваться в любой момент времени. Эти директивы применимы для любого раздела.

2.2.5. УПРАВЛЯЮЩИЕ ОПЕРАТОРЫ И ДИРЕКТИВЫ

В зависимости от того, какая часть управляющей программы принимает и обрабатывает управляющие операторы и директивы, они подразделяются на следующие группы (табл. 2.3):

директивы и операторы программы Управление заданиями;

директивы программы Внимание;

директивы программы Инициатор одиночных программ.

Директивы Управления заданиями вводятся с логического устройства SYSRDR или SYSLOG между заданиями или шагами одного задания; директивы Внимания — с пультовой пишущей машинки, назначенной для SYSLOG, в любое время после установления связи с этой программой; директивы Инициатора одиночных программ — с пультовой пишущей машинки, назначенной для SYSLOG, или с перфокарточного устройства ввода, назначенного для SYSRDR, в момент запуска одиночной программы.

Порядок следования операторов и директив в задании. Задание начинается оператором JOB, а заканчивается оператором /&. Каждое задание содержит один или несколько шагов. Шаг задания — это последовательность операторов, подготавливающих систему к выполнению очередной программы. Шаг заканчивается управляющим оператором EXEC или RSTRT. Предшествовать этим операторам в шаге задания могут любые операторы и директивы Управления заданиями. На порядок расположения операторов в задании накладываются следующие ограничения:

за оператором DLBL должны следовать операторы EXTENT, по одному на каждый участок описываемого файла;

директиве ASSGN при назначении дисков для SYSIN, SYSLST, SYSPCH, SYSCLB должны предшествовать операторы DLBL и EXTENT.

Общие правила записи. Структура:

<признак> <код> <операнды> <комментарии>

где признак —признак управляющего оператора (//). Исключение составляют операторы /*, /&, *. Признак допускается также в следующих пяти директивах Инициатора одиночных программ: DLBL, EXTENT, TLBL, LBLTYP, EXEC;

код — мнемонический код управляющего оператора или директивы;

операнды содержат уточняющую информацию, необходимую для выполнения той или иной операции. Операнды могут отсутствовать.

Составные части оператора или директивы отделяются одна от другой по крайней мере одним пробелом. Для записи оператора могут ис-

Таблица 2.3

Директивы и операторы управляющей программы

Назначение директивы и (или) оператора	Управление заданиями		Директивы Инициатора одиночных программ	Директивы программы Внимание
	Операторы	Директивы		
Идентификация задания	// JOB	—	—	—
	/&	—	—	—
Описание файлов	// DLBL	—	[//] DLBL	—
	// EXTENT	—	[//] EXTENT	—
	// TLBL	—	[//] TLBL	—
	/*	—	—	—
Информация для оператора	*	—	—	—
Информация для программы	// DATE	—	—	—
	// LBLTYP	—	[//] LBLTYP	—
	// OPTION	—	—	—
	// UPSI	—	—	—
Управление потоком заданий	—	—	—	BATCH
	—	CANCEL	CANCEL	CANCEL
	// PAUSE	PAUSE	—	PAUSE
	—	—	READ	—
	—	—	—	START
	—	STOP	—	—
	—	UNBATCH	—	—
Установка системных параметров	—	ALLOC	—	ALLOC
	—	SET	—	—
	—	—	TIMER	TIMER
Связь с оператором	—	—	—	ALTER
	—	—	—	DSPLY
	—	—	—	DUMP
	// LOG	LOG	LOG	LOG
	—	—	—	MODE
	—	—	MSG	MSG
	// NOLOG	NOLOG	NOLOG	NOLOG
	—	KT	KT	KT
Управление системой и вводом-выводом	// ASSGN	ASSGN	ASSGN	—
	// CLOSE	CLOSE	—	—
	—	DVCDN	—	—
	—	DVCUP	—	—
	—	HOLD	HOLD	—
	// LISTIO	LISTIO	LISTIO	—
	—	MAP	MAP	MAP
	// MTC	MTC	—	—
	—	RELSE	RELSE	—
	// RESET	RESET	—	—
	—	ROD	—	—
	—	UCS	UCS	—
	—	UNA	UNA	—
Выполнение программы	// EXEC	—	// EXEC	—
	// RSTRT	—	—	—

пользоваться перфокарты продолжения. Код директивы или признак оператора записываются с первой позиции.

Директива ALLOC используется для распределения основной памяти между разделами. В директиве указывается емкость памяти, отводимой для разделов переднего плана. Фоновому разделу автоматически отводится остаток области проблемных программ. Если в директиве указан один раздел, предполагается, что размер другого раздела переднего плана остается неизменным.

Директива программ Управление заданиями и Внимание:

$$\text{ALLOC} \quad \left\{ \begin{array}{l} F1 = nK[,F2 = nK] \\ F2 = nK[,F1 = nK] \end{array} \right\}$$

где F1, F2 = nK — размер раздела переднего плана;

n — целое четное число, K — 1024.

Директива ALTER позволяет с пишущей машинки изменить от 1 до 16 байт основной памяти, начиная с указанного адреса.

Директива программы Внимание:

ALTER xxxxxx

где xxxxxx — шестнадцатеричный начальный адрес изменяемой области в основной памяти (6 знаков). Незначащие нули не опускаются. Информация для замены вводится с пишущей машинки после ввода директивы. Для изменения каждого байта вводятся два символа (0—9, A—F), которые представляют собой шестнадцатеричный эквивалент обновляемого байта памяти.

Директива и оператор ASSGN назначают конкретное периферийное устройство логическому устройству. Назначение, выполненное Управлением заданиями по оператору // ASSGN, является временным, а по директиве ASSGN постоянным. Назначение, выполненное Инициатором одиночных программ по директиве ASSGN, сохраняет свою силу, как правило, только на время выполнения одиночной программы.

Оператор и директива программы Управление заданиями и директива Инициатора одиночных программ:

$$[//]\ \text{ASSGN}\ \text{SYSxxx},\ \left\{ \begin{array}{l} X'cuu' \\ UA \\ IGN \end{array} \right\} \quad \left[,\left\{ \begin{array}{l} X'ss' \\ \\ ALT \end{array} \right\} \right] [,TEMP]$$

где SYSxxx — логическое устройство;

X'cuu' — шестнадцатеричный адрес физического устройства;

UA — указывает, что назначение для логического устройства отменяется;

IGN — назначение для логического устройства отменяется и все обращения к этому логическому устройству игнорируются. Операнд не допускается для системных логических устройств SYSRDR, SYSIPT, SYSIN, SYSCLB;

X'ss' — модификатор плотности записи на магнитную ленту: X'C8' — 32 байта/мм, X'C0' — 63 байта/мм, X'F8' — 8 байт/мм. Если операнд опущен, предполагается X'C0';

ALT — указывает, что назначается сменная магнитная лента, которая будет использоваться тогда, когда емкость основной магнитной ленты, назначенной для этого же логического устройства, исчерпается. Характеристики сменной магнитной ленты должны быть такими же, как и характеристики основной. Операнд ALT не допускается для системных логических устройств SYSIN, SYSRDR, SYSIPT;

TEMP — указывает, что назначение, производимое данной директивой, временное. Этот операнд задается только в директиве Управления заданиями.

Директива BATCH используется для запуска пакетной обработки в одном из разделов переднего плана или для продолжения пакетной обработки в любом разделе, в котором она была перед этим остановлена по директиве STOP.

Директива программы Внимание:

$$\text{BATCH}\left[\left\{\begin{matrix}\underline{\text{BG}}\\ \underline{\text{F2}}\\ \text{F1}\end{matrix}\right\}\right]$$

Директива CANCEL снимает задание. Приняв директиву CANCEL, программа Управление заданиями снимает задание, которое только что было принято из входного потока раздела. Директива CANCEL, обрабатываемая Инициатором одиночных программ, отменяет запуск одиночной программы. Раздел освобождается. Директива CANCEL, обрабатываемая программой Внимание, прекращает выполнение программы в указанном разделе и, если раздел используется для пакетной обработки, снимает задание, в котором затребовано выполнение этой программы. Если программа выполнялась в одиночном режиме, то после прекращения выполнения программы раздел освобождается.

Директива программ Управление заданиями, Внимание и Инициатор одиночных программ:

$$\text{CANCEL}\left[\left\{\begin{matrix}\underline{\text{BG}}\\ \underline{\text{F2}}\\ \text{F1}\end{matrix}\right\}\right].$$

Операнд указывается только в директиве программы Внимание.

Директива и оператор CLOSE предназначены для закрытия файлов на одном логическом устройстве. Директива CLOSE используется для закрытия:

выводных файлов, расположенных на логических устройствах, которым назначена магнитная лента;

вводных и выводных файлов, размещенных на системных логических устройствах, которым назначены диски.

Оператор CLOSE употребляется для закрытия выводных файлов, расположенных на системных логических устройствах и логических устройствах программиста, которым назначена магнитная лента.

Директива и оператор Управления заданиями:

```
[//]CLOSE SYSxxx [ { ,X' cuu'   [,X'ss'] } ]
                  { ,UA                  }
                  { ,IGN                 }
                  { ,ALT                 }
```

где SYSxxx — логическое устройство, на котором расположены файлы, подлежащие закрытию;

X'cuu' — адрес физического устройства, которое после закрытия назначается для логического устройства;

X'ss' — модификатор плотности записи на магнитную ленту. Он аналогичен операнду X'ss' в операторе (директиве) ASSGN;

UA — указывает, что после закрытия назначение логическому устройству отменяется;

IGN — после закрытия назначение логическому устройству отменяется и все последующие обращения к этому устройству игнорируются;

ALT — после закрытия основной магнитной ленты, назначенной для логического устройства, должна быть открыта сменная магнитная лента (если она назначена логическому устройству). Этот операнд допускается только для логических устройств SYSLST, SYSPCH, SYSOUT.

Оператор DATE указывает текущую дату, действительную только для задания, в котором встретился этот оператор.

Оператор Управления заданиями:

```
// DATE { мм/дд/гг }
        { дд/мм/гг }
```

где дд — день (01—31); мм — месяц (01—12); гг — год (00—99).

Директива и оператор DLBL сообщают информацию, необходимую для создания или проверки меток файла на дисках.

Оператор Управления заданиями и директива Инициатора одиночных программ:

```
[//] DLBL имя файла,['идентификация'],[дата],
     [код],[секретность]
```

где имя файла — имя файла, используемое в макрокоманде DTF, описывающей файл (1—7 символов);

идентификация содержит от 1 до 44 алфавитно-цифровых символов (в том числе пробелы), заключенных в апострофы. Если операнд опущен, в качестве идентификации файла используется имя файла;

дата указывает количество дней, необходимых для хранения файла, или абсолютную дату истечения срока хранения. Количество дней задается десятичным числом (0 — 9999), дата — в виде гг/ддд, где гг — последние две цифры года, ддд — номер дня в году. Если операнд опущен, срок хранения принимается равным 7 дням;

код идентифицирует тип организации файла: SD — последовательная организация, DA — произвольная организация, ISC — индексно-

последовательная организация (создание), ISE — индексно-последовательная организация (расширение, добавление, извлечение). Если операнд опущен, принимается SD;

секретность — значение DSF, которым определяется, что доступ к создаваемому файлу должен контролироваться оператором.

Директива DSPLY используется для отображения содержимого основной памяти. Она позволяет оператору вывести на пишущую машинку 16 байт памяти. Каждый байт выводится в виде двух шестнадцатеричных цифр (0 — F).

Директива программы Внимание:

DSPLY xxxxxx

где xxxxxx — шестнадцатеричный адрес начала отображаемой области памяти (6 знаков). Незначащие нули не опускаются.

Директива DUMP позволяет оператору запросить с пишущей машинки вывод на SYSLST областей основной памяти.

Директива программы Внимание:

DUMP операнд 1[,операнд 2]

где операнд 1 может принимать значения: S, BG, F1, F2, BGS, F1S, F2S, CEAREA, xxxxxx, yyyyyy;

S означает печать основной памяти и общих регистров;

операнд 2 может принимать значения BG, F1, F2 и указывать, SYSLST какого раздела используется для вывода. Если этот операнд опущен, предполагается BG. Если оба операнда отстствуют, то распечатывается вся основная память (кроме области Супервизора) и общие регистры.

BG (F1, F2) — печать основной памяти соответствующего раздела и общих регистров;

BGS (F1S, F2S)-печать основной памяти соответствующего раздела, области Супервизора и общих регистров;

CEAREA — печать области CE;

xxxxxx, yyyyyy — печать участка основной памяти и общих регистров. xxxxxx и yyyyyy — соответственно начальный и конечный шестнадцатеричные адреса памяти (6 знаков).

Директива DVCDN запрещает использование физического устройства. Если это устройство в момент выдачи директивы оказывается назначенным одному или нескольким логическим устройствам, то все назначения отменяются.

Директива Управления заданиями:

DVCDN X'cuu'

где X'cuu' — шестнадцатеричный адрес физического устройства, использование которого запрещается.

Директива DVCUP разрешает использование физического устройства, работа с которым была запрещена директивой DVCDN. С этого момента данное физическое устройство может использоваться для выполнения назначений логическим устройствам.

Директива Управления заданиями:

DVCUP X' cuu'

Директива и оператор EXEC организуют запуск указанной в директиве или операторе проблемной программы.

Оператор Управления заданиями и директива Инициатора одиночных программ:

[//] EXEC [имя]

где имя — имя первой фазы программы, которая должна быть вызвана для выполнения из личной или системной библиотеки абсолютных модулей. Если имя опущено, выполняется программа, отредактированная в предыдущем шаге задания.

Директива и оператор EXTENT сообщают системе информацию об участке файла на дисках.

Оператор Управления заданиями и директива Инициатора одиночных программ:

//EXTENT [SYSxxx], [том], [тип участка], [номер участка], [адрес участка], [количество дорожек], [дорожка разделения]

где SYSxxx — логическое устройство, используемое для файла. Если операнд опущен, то применяется логическое устройство, указанное в предшествующем операторе или директиве EXTENT. Если операнд опущен и во всех предшествующих операторах или директивах, то используется логическое устройство, указанное в макрокоманде DTF, описывающей файл.

Для файлов, которые описаны макрокомандой DTF с операндом DEVADDR = SYSnnn, или для всех системных файлов с именем IJSYSxx (xx — IN, PH, LN, RS, LS, SL, RL, CL или RC), указание операнда SYSnnn в директиве или операторе EXTENT не обязательно. Для файлов, имеющих последовательную (SD) или произвольную (DA) организацию, логическое устройство, указанное в операторе или директиве EXTENT, заменяет логическое устройство, указанное операндом DEVADDR = SYSnnn в макрокоманде DTF, описывающей файл;

том — регистрационный номер тома, на котором находится описываемый участок (1—6 символов). Если операнд опущен, в качестве регистрационного номера тома используется номер, указанный в предшествующем операторе или директиве EXTENT. Если операнд опущен во всех операторах или директивах EXTENT, то во время открытия файла не производится контроль на правильность установки наружного тома;

тип участка принимает значения: 1 — для размещения данных на неразделенных цилиндрах; 2 — для независимой области переполнения (индексно-последовательный файл); 4 — для размещения индекса (индексно-последовательный файл); 8 — для размещения данных на разделенных цилиндрах (только последовательный файл). Если операнд опущен, предполагается 1;

номер участка — порядковый номер участка, отведенного для одного файла (0—225). Если операнд опущен, предполагается 1;

адрес участка — относительный номер первой дорожки участка (1—5 десятичных цифр). Нумерация ведется относительно дорожки 0 на цилиндре 0. Для индексно-последовательного и произвольного файлов указание адреса участка обязательно. Для вводного файла последовательного доступа этот операнд может быть опущен (в этом случае адрес участка берется из метки файла). Формула для подсчета адреса участка (A)

$$A = k \cdot c + h,$$

где $k = \begin{cases} 10 \text{ для дисков типа ЕС-5052;} \\ 20 \text{ для дисков типа ЕС-5061;} \end{cases}$

c — номер цилиндра;

h — номер головки;

количество дорожек — число дорожек, занимаемых участком (1—5 десятичных цифр). Для файлов, разделяющих цилиндры с другими файлами, количество дорожек представляет произведение числа цилиндров, отведенных файлу, на число дорожек, занимаемых этим файлом на каждом цилиндре. Указание количества дорожек участка для вводного файла последовательного доступа не обязательно;

дорожка разделения — номер последней из дорожек, занимаемых участком на каждом из цилиндров в случае разделения цилиндров (1—2 десятичных цифры). Номер первой дорожки этого участка берется из адреса участка.

Директива HOLD устанавливает режим сохранения назначений для раздела переднего плана. Это означает, что в разделе, в котором выполняется одиночная программа, все назначения логическим устройствам будут сохранены и после завершения одиночной программы, а в разделе с пакетной обработкой не будут отменены назначения для логических устройств по директиве UNBATCH.

Директива Управления заданиями и Инициатора одиночных программ:

$$\text{HOLD} \left\{ \begin{matrix} \text{F1} \ [,\text{F2}] \\ \text{F2} \ [,\text{F1}] \end{matrix} \right\}$$

Оператор JOB указывает начало последовательности управляющих операторов нового задания.

Оператор Управления заданиями:

// JOB имя задания[информация пользователя]

где имя задания — имя, идентифицирующее задание (1—8 символов);

информация пользователя состоит из 16 символов информации, которая помещается в таблицу статистики задания (см. 8.8). Если информация пользователя указана, она должна отделяться пробелом от имени задания.

Директива и оператор LBLTYP определяют емкость основной памяти, которая должна быть отведена в области проблемной программы для обработки меток файла. Этот оператор (директива) должен указы-

ваться либо в шаге редактирования программы, либо (для самоперемещающейся программы) в шаге ее выполнения.

Оператор Управления заданиями и директива Инициатора одиночных программ:

$$[//]\ \text{LBLTYP}\ \left\{\begin{matrix}\text{TAPE}\\ \text{NSD(nn)}\end{matrix}\right\}$$

где TAPE — программа, которая обрабатывает только файлы на магнитной ленте с метками;

NSD (nn) — программа, обрабатывающая индексно-последовательные файлы, файлы прямого доступа или файлы на дисках, описанные макрокомандой DTFPH с операндом MOUNTED = ALL; nn — максимальное количество участков, используемых одним файлом (десятичное число).

Если указан параметр TAPE, то резервируется 80 байт; если указан параметр NSD(nn), резервируется (84 + (20·nn)) байт.

Директива и оператор LISTIO используются при распечатке таблицы назначений для логических устройств. По директиве LISTIO таблица назначений выводится на SYSLOG, по оператору LISTIO — на SYSLST.

Директива и оператор Управления заданиями и директива Инициатора одиночных программ:

[//] LISTIO операнд

где операнд может принимать значения: SYS, PROG, F1, F2, ALL, SYSxxx, UNITS, DOWN, UA, X'cuu' для директивы и оператора Управления заданиями и значения: BG, F1, F2, UA, ALL — для директивы Инициатора одиночных программ. Каждое из значений определяет характер печатаемой информации.

Значение	Печатаемая информация
SYS	Таблица назначений для всех системных логических устройств раздела
PROG	Таблица назначений для всех логических устройств программиста в разделе
BG	Таблица назначений для всех логических устройств фонового раздела
F1	Таблица назначений для всех логических устройств первого раздела переднего плана
F2	Таблица назначений для всех логических устройств второго раздела переднего плана
ALL	Таблица назначений для всех логических устройств всех разделов
SYSxxx	Назначение для указанного операндом логического устройства в разделе
UNITS	Таблица всех физических устройств, которые назначены в данный момент логическим устройствам
DOWN	Список всех физических устройств, использование которых запрещено
UA	Список всех физических устройств, которые в момент выдачи директивы или оператора LISTIO не назначены ни одному из логических устройств
X'cuu'	Список всех логических устройств, которым назначено физическое устройство с адресом X'cuu'

Директива и оператор LOG управляют печатью управляющих операторов и директив. По директиве LOG устанавливается режим LOG для SYSLOG: все вводимые директивы и операторы печатаются на SYSLOG; по оператору LOG — режим LOG для SYSLST: все вводимые директивы и операторы печатаются на SYSLST.

Управление заданиями и Инициатор одиночных программ организуют печать директив и операторов того раздела, в котором принимается LOG. Программа Внимание по директиве LOG организует печать директив и операторов во всех разделах.

Оператор Управления заданиями, директива Управления заданиями, программы Внимание, Инициатора одиночных программ:

[//] LOG

Директива MAP используется для вывода на печать таблицы распределения основной памяти.

Директива программ Управление заданиями, Внимание и Инициатора одиночных программ:

MAP

Распечатка распределения памяти содержит следующие графы: AREA NO.K UPPER LIMIT NAME. Ниже приводится содержимое этих граф.

Графа	Содержимое графы
AREA	Идентификатор области памяти: SP — Супервизор; BG — фоновый раздел; F2 — второй раздел переднего плана; F1 — первый раздел переднего плана. Для того раздела, которому назначен таймер, через пробел после идентификатора печатается символ «Т».
NO.K	Размер раздела в виде nK (n — целое четное десятичное число). Для идентификатора SP эта графа содержит пробелы.
UPPER LIMIT	Верхняя граница области, указанной в графе AREA.
NAME	Имя задания или NO NAME. Для идентификатора SP — пробелы, для F1 и F2 — пробелы, если в разделах нет заданий.

Директива MODE устанавливает режим обработки исправимых машинных ошибок для ЭВМ «Ряда-2». Она обеспечивает: распечатку информации о текущем режиме обработки ошибок, изменение режима, изменение пределов частоты регистрации машинных ошибок.

Директива программы ВНИМАНИЕ:

MODE операнд1 [,операнд 2] [,операнд 3] [,операнд 4] [,операнд 5]

где операнд 1 может принимать значения: STATUS, HIR, ECC, R. Если заданы HIR или ECC, то должен присутствовать хотя бы один из трех последних операндов;

операнд 2 может принимать значения: M или C;

операнд 3 может принимать значения: R, Q или TH;

операнд 4: E≠ eeee;

операнд 5: T ≠ tttt;

STATUS указывает, что на SYSLOG должна быть распечатана информация о текущем режиме регистрации ошибок. Информация печатается в следующем формате:

$$\text{HIR}, \left\{ \begin{matrix} \text{R} \\ \text{Q} \end{matrix} \right\}, \quad \text{aaaa/eeee, bbbb/tttt}$$

$$\text{ECC}, \left\{ \begin{matrix} \text{R} \\ \text{Q} \end{matrix} \right\}, \left\{ \begin{matrix} \text{M} \\ \text{C} \end{matrix} \right\}, \text{aaaa/eeee, bbbb/tttt}$$

ECC,TH,C

где aaaa — текущее значение счетчика ошибок;
eeee — предельное значение счетчика ошибок;
bbbb — интервал времени от момента установки режима до ввода директивы MODE с операндом STATUS;
tttt — предельное значение интервала времени (в часах).

Наличие операнда STATUS не допускает задания никаких других операндов в директиве;

HIR указывает, что все последующие операнды описывают режим обработки ошибок процессора, который обеспечивается аппаратурными средствами HIR, выполняющими автоматическое повторение операций процессора;

ECC указывает, что все последующие операнды описывают режим обработки ошибок основной и управляющей памяти, который обеспечивается аппаратурными средствами ECC, выполняющими автоматическое исправление одиночных ошибок. Если задан операнд ECC, то обязательно должен присутствовать операнд 2 : M или C;

R — операнд, который указывает, что избавление от ошибок должно выполняться в режиме регистрации. В зависимости от сочетания R с другими операндами директивы MODE режим регистрации устанавливается для следующих средств избавления от ошибок:

для ECC и HIR, если задан только операнд R;

для HIR, если заданы операнды HIR и R;

для ECC в основной памяти, если заданы операнды ECC, M и R;

для ECC в управляющей памяти, если заданы операнды ECC, C и R;

Q — операнд, который указывает, что избавление от ошибок должно выполняться в скрытом режиме. В зависимости от сочетания Q с другими операндами директивы MODE скрытый режим устанавливается для следующих средств избавления от ошибок:

для ECC и HIR, если задан только операнд Q;

для ECC в основной памяти, если заданы операнды ECC, M и Q;

для ECC в управляющей памяти, если заданы операнды ECC, C и Q;

M(C) — операнд, который должен быть задан, если определен операнд ECC: M — для основной памяти (C — для управляющей);

TH — операнд, действительный только при задании ECC, C. В этом случае избавление от ошибок в управляющей памяти выполняется в пороговом режиме;

E = eeee — предельное значение счетчика ошибок: 8 — 9999;

T = tttt — предельное значение интервала времени (в часах): 8 — 9999.

Директива MSG передает управление подпрограмме связи с оператором. Желая организовать в проблемной программе связь с оператором, программист составляет и включает в свою программу подпрограмму связи с оператором. Директива MSG используется для установления связи оператора с этой подпрограммой в том случае, когда она выполняется в разделе переднего плана. О наличии подпрограммы связи программист должен сообщить системе с помощью макрокоманды STXIT.

Директива программы Внимание и Инициатора одиночных программ:

$$\text{MSG} \left\{ \begin{matrix} \text{F1} \\ \text{F2} \end{matrix} \right\}$$

Директива и оператор MTC предназначены для выполнения служебных операций на магнитной ленте.

Директива Управления заданиями:

$$\text{MTC код операции,} \left\{ \begin{matrix} \text{X'cuu'} \\ \text{SYSxxx} \end{matrix} \right\} [\text{,nn}]$$

Оператор Управления заданиями:

// MTC код операции,SYSxxx [,nn]

где код операции определяет вид служебной операции, выполняемой на магнитной ленте: BSF — Шаг на файл назад; BSR — Шаг на блок назад; ERG — Стереть промежуток; FSF — Шаг на файл вперед; FSR — Шаг на блок вперед; RUN — Перемотать и разгрузить; REW — Перемотать; WTM — Записать марку;

SYSxxx — логическое устройство, которому назначена магнитная лента;

X'cuu' — шестнадцатеричный адрес физического устройства;

nn — десятичное число (1—99), указывающее количество повторений служебной операции на магнитной ленте. Если параметр опущен, операция выполняется один раз.

Директива и оператор NOLOG отменяет режим LOG, установленный оператором или директивой. Программа Внимание по директиве NOLOG прекращает печать директив и операторов во всех разделах.

Директива и оператор программ Управление заданиями, Внимание и Инициатор одиночных программ:

[//] NOLOG

Оператор OPTION устанавливает режимы работы Управления заданиями, трансляторов и системных обслуживающих программ.

Оператор Управления заданиями:

// OPTION операнды

где операнды определяют режимы системы и могут записываться в произвольном порядке.

Режимы, которые можно указывать операндами, приведены ниже.

Операнд	Действие
LOG	Устанавливает режим LOG для SYSLST, при котором все директивы и операторы Управления заданиями выводятся на SYSLST
NOLOG	Отменяет режим LOG для SYSLST
DUMP	Устанавливает режим DUMP, при котором в случае ненормального завершения проблемной программы на SYSLST (если для него есть назначение) выводятся содержимое общих регистров и областей основной памяти, используемых управляющей программой и разделом, в котором выполнялась проблемная программа
NODUMP	Отменяет режим DUMP
LINK	Режим LINK для трансляторов указывает, что объектный модуль должен быть помещен на SYSLNK. Режим LINK для Редактора и Отладки указывает, что отредактированная программа должна быть помещена на временное хранение в библиотеку абсолютных модулей. Режим LINK не допускается для разделов переднего плана, если во время генерации системы не была указана возможность работы с личной библиотекой абсолютных модулей
NOLINK	Отменяет режим LINK. Трансляторы могут также отменять режим LINK, если проблемная программа содержит ошибку, которая препятствует успешному выполнению программы
DECK	Если указан режим DECK, то трансляторы осуществляют вывод объектных модулей на SYSPCH. Если указан режим LINK, то режим DECK игнорируется трансляторами РПГ и Бэйсиный Фортран
NODECK	Отменяет режим DECK
LIST	При указании этого режима трансляторы осуществляют распечатку исходного модуля на SYSLST. Ассемблер выводит на SYSLST также распечатку объектного модуля в шестнадцатеричном виде и перечень всех ошибок в исходной программе. Трансляторы Фортрана дополнительно выводят на SYSLST распечатку распределения памяти и сообщения об ошибках в исходной программе
NOLIST	Отменяет режим LIST
LISTX	Вызывает печать трансляторами Фортран IV и ПЛ/1 объектного модуля на SYSLST. Транслятор Кобол при указании этого режима осуществляет вывод на SYSLST объектного модуля, а также глобальных таблиц литералов и распределение регистров
NOLISTX	Отменяет режим LISTX
SYM	При указании этого режима Ассемблер выводит таблицу символических имен на SYSPCH, трансляторы ПЛ/1 и Кобол — на SYSLST
NOSYM	Отменяет режим SYM
XREF	При указании этого режима трансляторы ПЛ/1, Ассемблер и Кобол выводят таблицу перекрестных ссылок на SYSLST
NOXREF	Отменяет режим XREF
ERRS	При указании этого режима трансляторы Бэйсиный Фортран, Кобол и ПЛ/1 выводят список всех ошибок в исходной программе на SYSLST
NOERRS	Отменяет режим ERRS
CATAL	Указывает, что отредактированная Редактором программа должна быть каталогизирована в библиотеку абсолютных модулей. Указание режима CATAL ав-

	томатически вызывает установку режима LINK для трансляторов. Режим CATAL не допускается для разделов переднего плана, если во время генерации системы не была указана возможность работы с личной библиотекой абсолютных модулей
STDLABEL	При указании этого режима вся последующая информация о метках записывается на цилиндр меток в область стандартной информации о метках файлов. Информация доступна любой программе в любом разделе до тех пор, пока не будет введен новый оператор // OPTION STDLABEL. Режим STDLABEL отменяется по концу задания или шага задания (восстанавливается режим USRLABEL) и при указании режима PARSTD или USRLABEL
USRLABEL	При указании этого режима вся последующая информация о метках записывается на цилиндр меток в область временной информации о метках раздела
PARSTD	При указании этого режима вся последующая информация о метках записывается на цилиндр меток в область постоянной информации о метках раздела. Информация доступна любой программе в данном разделе до тех пор, пока не будет введен новый оператор // OPTION PARSTD. Режим PARSTD отменяется по концу задания или шага задания (восстанавливается режим USRLABEL) и при указании режима USRLABEL или STDLABEL. Поиск информации о метках файла во время открытия файла производится в следующем порядке: область USRLABEL, область PARSTD, область STDLABEL
48C	Определяет 48-символьный алфавит на SYSIPT (для ПЛ/1)
60C	Определяет 60-символьный алфавит на SYSIPT (для ПЛ/1)
SYSPARM='xxxxxxxx'	Задает строку алфавитно-цифровых символов, определяющих значение параметра &SYSPARM для транслятора Ассемблер (см. 1.5.1). Длина строки не более восьми символов. Апостроф внутри строки представляется двумя апострофами. Операнд не допускается, если во время генерации системы не была указана эта возможность.

Директива и оператор PAUSE организуют паузу: приостанавливается ввод потока управляющих операторов и устанавливается связь оператора с программой Управление заданиями. Оператор организует паузу немедленно после ввода этого оператора, директива Управление заданиями — после выполнения программы, вызываемой в текущем шаге задания, а директива программы Внимание — в конце выполняющегося (или подготавливаемого) шага задания в указанном разделе.

Оператор и директива Управления заданиями:

[//] PAUSE [любые комментарии]

Директива программы Внимание:

$$\text{PAUSE}\left\{\begin{matrix}\text{BG}\\\text{F2}\\\text{F1}\end{matrix}\right\}[,\text{EOJ}]$$

где BG, F2, F1 — раздел, в котором организуется пауза;

EOJ указывает, что организуется пауза после выполнения всего задания.

Директива READ определяет устройство, с которого должны вводиться перфокарты с последующими директивами, и прекращает связь оператора с Инициатором одиночных программ.

Директива Инициатора одиночных программ:

READ X'cuu'

Директива RELSE отменяет режим HOLD. По окончании выполнения одиночной программы отменяются все назначения для логических устройств раздела.

Директива Управления заданиями и Инициатора одиночных программ:

$$\text{RELSE} \begin{Bmatrix} \text{F1[,F2]} \\ \text{F2[,F1]} \end{Bmatrix}$$

Директива и оператор RESET восстанавливают стандартные или постоянные назначения для логических устройств.

Оператор и директива Управления заданиями:

[//] RESET операнд

где операнд может принимать значения: SYS, PROG, ALL, SYSnnn.

Значения операнда определяют группу логических устройств, для которых должны восстанавливаться стандартные или постоянные назначения:

Значение операнда	Группа устройств
SYS	Для всех системных логических устройств
PROG	Для всех логических устройств программиста
ALL	Для всех логических устройств
SYSnnn	Для указанного логического устройства

Директива ROD употребляется по окончании работы с системой и перед выводом на печать регистрационного файла, для того чтобы занести в записи SDR регистрационного файла (см. 8.4) содержимое накопленных в основной памяти счетчиков об ошибках ввода-вывода.

Директива Управления заданиями:

ROD

Оператор RSTRT предназначен для запуска программы с контрольной точки. При этом имя задания, указанное в операторе JOB, должно быть идентичным имени задания, используемому при создании контрольной точки.

Оператор Управления заданиями:

// RSTRT SYSxxx,nnnn [,имя файла]

где SYSxxx — логическое устройство, на котором расположен файл контрольных точек;

nnnn — номер контрольной точки, с которой производится запуск программы. Номера контрольных точек печатаются на SYSLOG во время их создания;

имя файла — имя файла контрольных точек. Этот операнд указывается в том случае, если информация контрольной точки записана на диски. Имя файла должно быть идентичным имени файла, используемому в макрокоманде DTFPH, описывающей файл контрольных точек на дисках, и в макрокоманде CHKPT.

Директива SET устанавливает значения.

Директива Управления заданиями:

```
SET [DATE=n1][,CLOCK=n2][,UPSI=n3][,LINECT=n4]
    [,RCLST=n5] [,RCPCH=n6][,RF=n7]
```

где DATE = n1 — операнд, который устанавливает системную дату; n1 может быть записано в формате мм/дд/гг или дд/мм/гг (дд — день (01—31), мм — месяц (01—12), гг — год (00—99)). Используемый формат должен соответствовать формату, выбранному во время генерации системы;

CLOCK = n2 — операнд, определяющий системное время; n2 записывается в формате чч/мм/сс (чч — часы (00—23), мм — минуты (00—59), сс — секунды (00—59));

UPSI = n3 — операнд, устанавливающий значения бит байта программных переключателей UPSI в области связи; n3 записывается в виде строки (1—8) символов, значениями которых могут быть 0, 1 или X. Соответствующий бит байта UPSI устанавливается в 0, если значение 0; устанавливается в 1, если значение 1, и не изменяется, если значение X. Если указано меньше восьми символов, состояние оставшихся крайних справа бит в байте UPSI не изменяется;

LINECT = n4 — устанавливает количество строк, печатаемых в каждой странице на SYSLST; n4 — целое число от 30 до 99;

RCLST = n5 — устанавливает минимальную величину свободной для вывода области файла на SYSLST, если для SYSLST используются диски. Если перед выполнением очередного шага задания величина свободной области файла для SYSLST оказывается равной или меньше указанного минимального значения, на SYSLOG выводится сообщение, предупреждающее оператора о приближении конца участка. Значение n5 может изменяться от 100 до 65 535. Оно определяет количество записей, вмещающихся в свободную область файла. Рекомендуется выбирать величину свободной области таким образом, чтобы она была достаточной для одного шага задания. Если операнд опущен, используется значение, установленное при генерации;

RCPCH = n6 — определяет минимальную величину свободной для вывода области файла на SYSPCH, если для SYSPCH используются диски; n6 — десятичное число, выражающее величину этой области в количестве записей, вмещающихся в нее. Если перед выполнением очередного шага задания величина свободной области файла на SYSPCH оказывается равной указанному минимальному значению или меньше его, на SYSLOG выводится сообщение, предупреждающее оператора о приближении конца участка. n6 может принимать значения от 100 до 65 535. Этот операнд определяет количество записей, вмещающихся в свободную область файла. Рекомендуется устанавливать величину свободной области таким образом, чтобы она была достаточной для

одного шага задания. Если операнд опущен, используется значение, установленное при генерации;

RF = n7 — задает состояние регистрационного файла на SYSREC, используемого для сбора статистики об ошибках процессора, каналов, интерфейса и периферийных устройств. n7 может принимать значения YES, NO или CREATE. Если операнд опущен, предполагается YES. YES указывает, что существует активный регистрационный файл (открытие этого файла производится по первому оператору // JOB после процедуры начальной загрузки); NO — что отменяется использование регистрационного файла на SYSREC (при возникновении машинной ошибки система переводится в состояние жесткого ожидания. Операнд RF = NO недействителен для моделей «Ряда-2»); CREATE — что должен быть создан регистрационный файл на SYSREC (Создание выполняется по первому оператору //JOB после процедуры начальной загрузки).

Директива START предназначена для запуска одиночной программы в разделе переднего плана или для возобновления пакетной обработки, приостановленной по директиве STOP в любом разделе.

Директива программы Внимание:

$$\text{START} \left\{ \begin{matrix} \underline{\text{BG}} \\ \text{F1} \\ \text{F2} \end{matrix} \right\}$$

где BG вызывает ввод очередного оператора из входного потока фонового раздела. Директива START BG может быть использована только в том случае, если до этого пакетная обработка в фоновом разделе была остановлена по директиве STOP;

F1, F2 вызывает запуск обработки задания в указанном разделе переднего плана в одиночном режиме либо возобновление пакетной обработки в указанном разделе переднего плана, если она ранее была остановлена директивой STOP.

Директива STOP временно прекращает пакетную обработку в разделе, в котором эта директива выдана.

Директива Управления заданиями:

STOP

Директива TIMER предоставляет возможность назначения интервального таймера одному из разделов.

Директива программ Внимание и Инициатор одиночных программ:

$$\text{TIMER} \left\{ \begin{matrix} \underline{\text{BG}} \\ \text{F1} \\ \text{F2} \end{matrix} \right\}$$

Директива и оператор TLBL сообщают информацию, необходимую для создания или проверки меток файла на магнитной ленте.

Оператор Управления заданиями и директива Инициатора одиночных программ:

[//] TLBL имя файла, ['идентификатор'], [дата], [регистрационный номер файла], [порядковый номер тома], [порядковый номер файла], [номер поколения], [номер варианта поколения]

где имя файла — имя файла, используемое в макрокоманде DTF, описывающей файл (1—7 символов). Набор имен системных файлов строго определен;

'идентификатор' содержит от 1 до 17 алфавитно-цифровых символов, (в том числе пробелы), заключенных в апострофы. Если этот операнд опущен, то для выводного файла в качестве идентификатора используется имя файла, для вводного файла проверка идентификатора не производится;

дата для выводных файлов указывает срок хранения файла в днях (0 — 9999). Если операнд опущен, срок хранения файла принимается равным 0. В качестве даты создания для выводных файлов всегда используется текущая системная дата. Для вводных файлов этот операнд указывает дату создания файла, содержащую две последние цифры года и номер дня в году (гг/ддд). Если операнд опущен, проверка даты создания файла не производится;

регистрационный номер файла идентифицирует первую (или единственную) катушку файла на магнитной ленте (1—6 символов). Если операнд опущен, то для выводного файла используется регистрационный номер тома первой (или единственной) катушки файла, для вводного файла контроль регистрационного номера не производится;

порядковый номер тома определяет порядковый номер тома многотомного файла (1—4 цифры). Этот номер последовательно возрастает от тома к тому. Если операнд опущен, то для выводного файла его значение принимается равным 0001, для вводного файла контроль порядкового номера тома не производится;

порядковый номер файла определяет порядковый номер файла в многофайловом томе (1—4 цифры). Этот номер последовательно возрастает от файла к файлу в одном томе. Если операнд опущен, то для выводного файла его значение принимается равным 0001, контроль порядкового номера файла не производится;

номер поколения дополняет идентификатор файла (1—4 цифры). Если операнд опущен, то для выводного файла его значение принимается равным 0001, для вводного файла контроль номера поколения не производится;

номер варианта поколения дополняет номер поколения (1—2 цифры). Если операнд опущен, то для выводного файла его значение принимается равным 01, для вводного файла проверка номера варианта поколения не производится.

Директива UCS устанавливает соответствие между двоичными кодами байт в основной памяти и печатаемыми символами. Директива может использоваться для печатающих устройств, имеющих буфер UCS.

Директива Управления заданиями и Инициатора одиночных программ:

UCS SYSxxx, имя фазы [,FOLD] [,BLOCK] [,NULMSG]

где SYSxxx — логическое устройство;

имя фазы — имя фазы, содержащей 240 байт, двоичные коды которых помещаются в буфер UCS. Кроме того, фаза может также содержать контрольное сообщение в 80 байт. Эта фаза должна быть предварительно каталогизирована в библиотеку абсолютных модулей;

FOLD указывает, что сравнение двоичного кода печатаемого байта с содержимым буфера UCS производится по шести младшим двоичным битам. Если операнд опущен, то сравниваются все восемь бит;

BLOCK указывает, что подавление на устройстве печати ошибки данных вызвано тем, что в буфере UCS не был найден код, совпадающий с кодом, поступившим для печати;

NULMSG отменяет вывод на устройство печати контрольного сообщения. При отсутствии этого операнда после загрузки буфера UCS выполняется подвод нового листа и печать 80 последних байт фазы, указанной вторым операндом.

Директива UNA немедленно отменяет назначения для всех логических устройств указанного раздела, кроме SYSRES, SYSLOG и SYSREC.

Директива Управления заданиями и Инициатора одиночных программ:

UNA { F1[, F2] / F2[, F1] }

Директива UNBATCH прекращает пакетную обработку в разделе. Директива допускается только между заданиями и может вводиться только с SYSLOG после установления связи оператора с программой Управление заданиями. По директиве UNBATCH отменяются назначения для всех логических устройств раздела, кроме SYSRES, SYSLOG и SYSREC. При установленном режиме HOLD назначения для логических устройств сохраняются. Директива UNBATCH действительна только для разделов переднего плана.

Директива Управления заданиями:

UNBATCH

Оператор UPSI устанавливает значения бит байта программных переключателей UPSI в области связи.

Оператор Управления заданиями:

// UPSI строка символов

где строка символов содержит от одного до восьми символов 0, 1 или X. Каждый символ соответствует одному биту в байте UPSI: первый — биту 0, второй — биту 1 и т. д. Бит, которому соответствует символ 0, устанавливается в 0; бит, которому соответствует символ 1, устанав-

ливается в 1; бит, которому соответствует символ X, не изменяется. Если операнд содержит меньше восьми символов, состояние оставшихся крайних справа бит в байте не изменяется.

Оператор /* указывает конец файла или конец шага задания. Он должен быть последним оператором в любом файле, вводимом с SYSIPT. По этому оператору Супервизор устанавливает бит «Особый случай» в блоке ССВ.

Оператор Управления заданиями:

```
/*
```

Оператор /& указывает конец задания. По оператору /& Супервизор устанавливает бит «Особый случай» в блоке ССВ. По этому оператору восстанавливаются постоянные или стандартные назначения для логических устройств раздела.

Оператор Управления заданиями:

```
/&  [комментарии]
```

Оператор * используется для сообщения некоторой информации оператору ЭВМ. Этот оператор всегда печатается на SYSLOG. Оператор *, помещенный перед оператором // PAUSE, может запрашивать выполнение некоторых действий оператора ЭВМ во время паузы.

Оператор Управления заданиями:

```
* комментарии
```

Директива КТ вводится оператором после ввода всех необходимых директив. Она прекращает связь с управляющей программой.

Директива программ Внимание, Управление заданиями и Инициатор одиночных программ:

Нажатие кнопки КТ на пультовой пишущей машинке

2.3. ФУНКЦИИ СУПЕРВИЗОРА

Программист может запросить функции, обеспечиваемые Супервизором, с помощью макрокоманд генерации Супервизора и макрокоманд связи с Супервизором.

Макрокоманды генерации Супервизора используются в момент генерации конкретного резидента ДОС ЕС и дают возможность пользователю из большого числа функций и возможностей, предоставляемых Супервизором, выбрать наиболее оптимальные для конкретной вычислительной системы и круга ее применения (см. гл. 9).

Макрокоманды связи с Супервизором используются в любой проблемной программе, написанной на Ассемблере. Они дают возможность пользователю в удобной форме записывать запросы на выполнение функций, которые необходимы проблемной программе, но могут быть реализованы только Супервизором.

2.4. МАКРОКОМАНДЫ СВЯЗИ С СУПЕРВИЗОРОМ

Общие правила записи. Структура:

<имя> <код> <операнды> <комментарии>

где имя используется для идентификации макрокоманды;

код — мнемонический код запрашиваемой функции Супервизора;

операнды употребляются для передачи информации, необходимой для выполнения запрашиваемой функции. Они могут записываться в позиционном, ключевом или смешанном формате;

комментарии предназначены для записи описательной информации о программе.

Составные части макрокоманды отделяются одна от другой по крайней мере одним пробелом. Макрокоманды могут записываться на перфокартах продолжения.

2.4.1. МУЛЬТИВЕТВЛЕНИЕ

Макрокоманда ATTACH используется в основной задаче для подключения подзадач. Формат макрокоманды:

$$[\text{имя}]\quad \text{ATTACH}\quad \left\{\begin{matrix}\text{вход}\\(\text{r0})\end{matrix}\right\},\text{SAVE}=\left\{\begin{matrix}\text{область сохранения 1}\\(\text{r1})\end{matrix}\right\}$$

$$\left[,\text{ECB}=\left\{\begin{matrix}\text{блок ECB}\\(\text{r2})\end{matrix}\right\}\right]\left[,\text{ABSAVE}=\left\{\begin{matrix}\text{область сохранения 2}\\(\text{r3})\end{matrix}\right\}\right]$$

где вход — символический адрес точки входа в подзадачу. Для указания точки входа не должен использоваться регистр 1;

SAVE — адрес области сохранения подзадачи. Адрес должен быть на границе двойного слова. Область сохранения имеет длину 96 байт, если задан режим генерации FP=NO, или 128 байт, если задан режим генерации FP=YES (см. 9.2);

ECB — адрес связанного с этой подзадачей блока управления событиями (ECB). Блок ECB должен создаваться программистом и представляет собой область из 4 байт. Блок используется Супервизором для занесения в него информации о выполнении подзадачи. При этом байты 0, 1, 3 блока ECB не используются, байт 2 содержит признаки завершения подзадачи: бит 0 установлен в 1, когда подзадача завершена (любым способом); бит 1 установлен в 1 только в том случае, когда подзадача завершена ненормально; биты 2—7 резервируются. В подзадаче обязательно должен быть создан блок ECB, если подзадача предполагает использовать защиту ресурсов или этой подзадаче требуется синхронизация с другими задачами;

ABSAVE — адрес области сохранения подпрограммы, обрабатывающей ненормальное завершение подзадачи (72 байта). Указывается, если подзадача использует для обработки ненормального завершения подпрограмму AB (см. 2.4.8) основной задачи.

Макрокоманда DETACH отключает подзадачу. Формат макрокоманды:

$$[\text{имя}]\quad \text{DETACH}\quad \left[\text{SAVE}=\left\{\begin{array}{c}\text{область сохранения}\\(1)\end{array}\right\}\right]$$

где SAVE — адрес области сохранения отключаемой подзадачи. Задается обязательно, если DETACH выдает основная задача. Если основная задача выдает DETACH без операнда, снимается весь раздел.

2.4.2. ЗАЩИТА РЕСУРСОВ

Под ресурсами программы понимаются такие объекты, как периферийные устройства, рабочие области, константы, группы команд и т. д. Система предоставляет программисту возможность защитить свои ресурсы таким образом, чтобы несколько задач, использующих один и тот же ресурс, не могли иметь к нему одновременного доступа. Для защиты ресурсов применяются три макрокоманды: RCB, ENQ, DEQ.

По макрокоманде RCB создается блок управления ресурсом. Этот блок программист ставит в соответствие ресурсу, который необходимо защитить. Чтобы защитить ресурс, необходимо выдать макрокоманду ENQ. По этой макрокоманде в блоке управления ресурсом (RCB) делается отметка о захвате ресурса задачей. Ресурс закрепляется за задачей до тех пор, пока он не будет освобожден с помощью макрокоманды DEQ.

Макрокоманда RCB генерирует константу длиной 8 байт — блок RCB, который используется для защиты ресурса. Структура блока RCB:

Байты	Содержимое
0	Признак защиты ресурса
1—3	Резервируются
4	Байт флажков
5—7	Адрес ECB текущего владельца ресурса

Все нули в байте 0 указывают, что ресурс свободен; все единицы в байте 0 указывают, что ресурс защищен. Если бит 0 байта 4 содержит 1, то это значит, что имеются задачи, ожидающие освобождения ресурса. В это время в байтах 5—7 записан адрес блока ECB задачи, использующей ресурс. Формат макрокоманды:

[имя] RCB

Макрокоманда ENQ используется для защиты ресурсов. Задача, запросившая ресурс, соответствующий указанному блоку RCB, либо защищает его и продолжает выполняться, либо переводится в состояние ожидания, если ресурс используется другой задачей. Когда задача завершает использование ресурса, она выдает для него макрокоманду DEQ. Все другие задачи, ожидавшие освобождения ресурса, выводятся из состояния ожидания, и задача с наивысшим приоритетом получает управление.

Каждая подзадача, выдающая макрокоманду ENQ, должна иметь блок ECB, и этот блок не должен применяться для управления никакими другими событиями, пока используется ресурс. Если задача завершается, предварительно не освободив защищенных ею ресурсов, то любая другая задача, пытающаяся захватить эти ресурсы, завершается ненормально. Формат макрокоманды:

$$[\text{имя}]\quad \text{ENQ}\quad \left\{\begin{array}{c}\text{блок RCB}\\(0)\end{array}\right\}$$

где блок RCB — символический адрес блока управления ресурсом.

Макрокоманда DEQ освобождает ресурс. Освободить ресурс может только задача, захватившая этот ресурс по макрокоманде ENQ. Когда защита ресурса отменяется, в байт 0 блока RCB записывается код X'00', и все задачи, ожидающие этот ресурс, выводятся из состояния ожидания. Управление передается задаче с наивысшим приоритетом. Формат макрокоманды:

$$[\text{имя}]\quad \text{DEQ}\quad \left\{\begin{array}{c}\text{блок RCB}\\(0)\end{array}\right\}$$

2.4.3. СВЯЗЬ МЕЖДУ ЗАДАЧАМИ

В случае, когда задачи в разделе зависят одна от другой, синхронизацию их выполнения обеспечивают макрокоманды WAIT, WAITM, POST.

Макрокоманда WAIT приостанавливает выполнение задачи до тех пор, пока не произойдет событие, которое описано блоком управления событиями. После выполнения события управление получает команда, следующая за макрокомандой WAIT. Формат макрокоманды:

$$[\text{имя}]\quad \text{WAIT}\quad \left\{\begin{array}{c}\text{блок ECB}\\(1)\end{array}\right\}$$

где блок ECB — символический адрес блока управления событиями.

Макрокоманда WAITM организует мультиожидание. Она приостанавливает выполнение задачи до тех пор, пока не произойдет любое из ожидаемых событий. Формат макрокоманды:

$$[\text{имя}]\quad \text{WAITM}\quad \left\{\begin{array}{c}\text{ECB1,ECB2, . . .}\\\text{адрес таблицы}\\(1)\end{array}\right\}$$

где ECB1, ECB2, ... — символические адреса блоков управления событиями, по которым выполняется ожидание;

адрес таблицы — символический адрес таблицы слов, представляющих собой адреса блоков ECB.

Макрокоманда POST вызывает принудительный вывод из состояния ожидания событий. По макрокоманде POST делается отметка о событии в блоке управления событиями ECB, после чего одна или несколько задач, ждущих отметки в этом блоке, выводятся из состояния ожидания. Формат макрокоманды:

$$[\text{имя}]\ \text{POST}\ \left\{\begin{matrix}\text{блок ECB}\\(1)\end{matrix}\right\}\left[,\text{SAVE}=\left\{\begin{matrix}\text{область сохранения}\\(0)\end{matrix}\right\}\right]$$

где блок ECB — символический адрес блока управления событиями, в котором нужно сделать отметку;

SAVE — адрес области сохранения задачи, которую нужно вывести из состояния ожидания. Обычно эта задача ждет отметки в заданном ECB. Другие задачи, ждущие отметки в этом ECB, не выводятся из состояния ожидания, пока не будет выдана снова макрокоманда POST. Если операнд опущен, то из состояния ожидания выводятся все задачи, ждущие отметки в этом ECB.

2.4.4. ЗАГРУЗКА ФАЗ

Для загрузки фаз проблемной программы в основную память из личной или из системной библиотеки абсолютных модулей используются макрокоманды FETCH и LOAD.

Макрокоманда FETCH предназначена для загрузки в память фазы и передачи ей управления. Фаза загружается по адресу, определенному при редактировании. Управление передается по адресу, указанному в макрокоманде, или, если второй операнд отсутствует, — по адресу, определенному в качестве точки входа при редактировании фазы. Формат макрокоманды:

$$[\text{имя}]\ \text{FETCH}\ \left\{\begin{matrix}\text{фаза}\\(1)\end{matrix}\right\}\left[,\left\{\begin{matrix}\text{вход}\\(0)\end{matrix}\right\}\right]$$

где фаза — имя загружаемой фазы (1—8 символов);

вход — символический адрес точки входа в загружаемую фазу.

Макрокоманда LOAD загружает в память фазу без передачи ей управления. Фаза загружается по адресу, указанному в макрокоманде, или, если второй операнд отсутствует, — по адресу, определенному в качестве адреса загрузки при редактировании. Формат макрокоманды:

$$[\text{имя}]\ \text{LOAD}\ \left\{\begin{matrix}\text{фаза}\\(1)\end{matrix}\right\}\left[,\left\{\begin{matrix}\text{адрес загрузки}\\(0)\end{matrix}\right\}\right]$$

где фаза — имя загружаемой фазы (1—8 символов);

адрес загрузки — символический адрес загрузки фазы.

По окончании выполнения макрокоманды LOAD регистр 1 содержит адрес точки входа загруженной фазы.

2.4.5. ДОСТУП К ОБЛАСТИ СВЯЗИ

Ниже приведено содержимое той части области связи, которая может использоваться проблемной программой:

Байты	Содержимое
0—7	Календарная дата
8—9	Адрес начала области проблемных программ
10—11	Адрес начала защиты ключом 1
12—22	Область пользователя. Используется проблемными программами
23	Программные переключатели (UPSI)
24—31	Имя задания
32—35	Адрес конца раздела
36—39	Адрес конца фазы
40—43	Адрес конца программы
44—45	Длина области меток

Макрокоманда COMRG используется для получения адреса области связи соответствующего раздела. После выполнения макрокоманды адрес находится в регистре 1. Формат макрокоманды:

[имя] COMRG

Макрокоманда MVCOM модифицирует область связи. Формат макрокоманды:

[имя] MVCOM смещение, длина, $\left\{\begin{array}{c}\text{адрес}\\(0)\end{array}\right\}$

где смещение — смещение первого изменяемого байта области связи относительно начала области связи;

длина — количество байт (1—12), которые необходимо переслать в область связи;

адрес — символический адрес данных, пересылаемых в область связи.

2.4.6. ОТМЕНА ВРЕМЕННЫХ НАЗНАЧЕНИЙ

Эти макрокоманды обеспечивают эффективное использование периферийных устройств: закончив работу с устройством, программа может освободить его для применения в другом разделе.

Макрокоманда RELEASE отменяет временные назначения для логических устройств программиста. Эта макрокоманда действительна только для заданий, выполняющихся в режиме пакетной обработки. Формат макрокоманды:

[имя] RELEASE (SYSxxx,...) [, область сохранения]

где (SYSxxx,...) — список логических устройств программиста, для которых нужно отменить назначения (максимум 16 устройств);

область сохранения — символический адрес области сохранения (8 байт), в которой запоминается содержимое регистров 0 и 1 перед использованием их в макрокоманде. Содержимое регистров 0 и 1 сохраняется только в том случае, если записан второй операнд.

2.4.7. РАБОТА С ТАЙМЕРОМ

Макрокоманда GETIME используется для получения текущего времени дня. Формат макрокоманды:

[имя] GETIME $\left[\left\{\begin{array}{c}\text{STANDARD}\\\hline\text{BINARY}\\\text{TU}\end{array}\right\}\right]$

где STANDARD указывает, что время дня помещается в регистр 1 в виде упакованного десятичного числа (ччммсс);

BINARY — время дня помещается в регистр 1 в виде двоичного целого числа в секундах;

TU — время дня помещается в регистр 1 в виде двоичного целого числа единиц по 1/300 с.

Макрокоманда SETIME устанавливает таймер на заданную величину. Наибольший интервал времени, который можно задать, равен 55 924 с, что эквивалентно 15 ч 32 мин 4 с. Формат макрокоманды:

$$[\text{имя}]\quad \text{SETIME}\ \left\{\begin{matrix}\text{секунды}\\(1)\end{matrix}\right\}\left[,\left\{\begin{matrix}\text{блок TECB}\\(r)\end{matrix}\right\}\right]$$

где секунды — количество секунд (десятичное число), определяющее длину интервала времени;

блок TECB — символический адрес блока событий таймера.

Макрокоманда TECB строит блок TECB — блок событий таймера (4 байта), в котором Супервизор производит отметку об истечении интервала времени. Структура блока TECB:

байты 0, 1, 3 — резервируются,
байт 2 — индикация события (бит 0).

Единичное значение бита 0 байта 2 (бита ожидания) говорит о том, что интервал времени, заданный в макрокоманде SETIME, истек. Формат макрокоманды:

имя TECB

Макрокоманда WAIT приостанавливает выполнение программы до тех пор, пока не истечет время, заданное макрокомандой SETIME. Формат макрокоманды:

$$[\text{имя}]\quad \text{WAIT}\ \left\{\begin{matrix}\text{блок TECB}\\(1)\end{matrix}\right\}$$

где блок TECB — символический адрес блока событий таймера.

2.4.8. ОБРАБОТКА ПРЕРЫВАНИЙ ПРОБЛЕМНОЙ ПРОГРАММОЙ

При работе в ДОС ЕС отдельные прерывания могут обрабатываться проблемной программой. Для того чтобы Супервизор при возникновении таких прерываний передавал управление соответствующей подпрограмме проблемной программы, к моменту возникновения прерывания должна быть установлена связь Супервизора с этой подпрограммой.

Макрокоманда STXIT устанавливает связь Супервизора с подпрограммами проблемной программы, предназначенными для обработки прерываний, а также сообщает Супервизору о прекращении такой связи.

Установка связи с подпрограммой обработки прерываний по таймеру (подпрограмма IT) действительна только для раздела, в котором разрешено использование таймера. В системе с мультизадачным режимом связь с подпрограммой IT может устанавливать только основная задача. Установка связи с подпрограммой обработки прерываний по программным ошибкам (подпрограмма PC) возможна во всех разделах. В системе с мультизадачным режимом каждая подзадача и основная

задача могут иметь свою собственную подпрограмму для обработки программных прерываний. Устанавливать связь с подпрограммой обработки прерываний по требованию оператора (подпрограмма ОС) имеет право только основная задача.

Выполнение подпрограммы обработки ненормальных завершений программы (подпрограмма AB), указанной в макрокоманде STXIT, начинается по любому завершению программы, кроме завершения по макрокомандам CANCEL, DUMP, EOJ и макрокоманде DETACH. Устанавливать связь с подпрограммой AB может каждый раздел. В системе с мультизадачным режимом устанавливать связь с подпрограммой AB может как основная задача, так и подзадача, или подзадача может использовать для обработки ненормальных завершений подпрограмму AB основной задачи. Когда подпрограмма AB получает управление, регистр 0 содержит код завершения программы или задачи; список кодов завершения приведен ниже.

Код завершения	Причина завершения
17	Основная задача, имеющая подключенные подзадачи, выдала макрокоманду CANCEL
18	Основная задача, имеющая подключенные подзадачи, выдала макрокоманду DUMP
19	Оператор ответил CANCEL на сообщение об ошибке ввода-вывода
1A	Ошибка ввода-вывода
1B	Ошибка канала
1C	Подзадача выдала макрокоманду CANCEL ALL
1D	Основная задача завершена
1E	Неизвестна задача, выдавшая макрокоманду ENQ
1F	Машинная ошибка
20	Программная ошибка
21	Недействительная команда SVC
22	Не найдена фаза
23	Задача выдала макрокоманду CANCEL
24	Вмешательство оператора
25	Недействительный адрес основной памяти
26	Нет назначения для логического устройства
27	Логическое устройство не определено
30	Чтение после /& на SYSRDR или SYSIPT
31	Переполнение таблицы ошибок ввода-вывода
32	Недействительный адрес на дисках
33	Отсутствует команда подвода для дисков

Формат макрокоманды:

$$[\text{имя}]\ \text{STXIT}\ \left\{\begin{matrix}\text{AB}\\ \text{PC}\\ \text{IT}\\ \text{OC}\end{matrix}\right\},\left\{\begin{matrix}\text{вход}\\ (0)\end{matrix}\right\},\left\{\begin{matrix}\text{область сохранения}\\ (1)\end{matrix}\right\}$$

где AB указывает, что устанавливается или прекращается связь с подпрограммой обработки ненормального завершения проблемной программы;

PC — устанавливается или прекращается связь с подпрограммой обработки программных ошибок;

IT — устанавливается или прекращается связь с подпрограммой обработки прерываний по таймеру;

ОС — устанавливается или прекращается связь с подпрограммой обработки прерываний по требованию оператора;

вход — символический адрес точки входа в подпрограмму обработки прерывания;

область сохранения — символический адрес области сохранения (72 байта), в которую Супервизор записывает старое PSW и содержимое общих регистров 0—15. Каждая подпрограмма проблемной программы должна иметь отдельную область сохранения.

Чтобы отменить связь с подпрограммами обработки прерываний, в макрокоманде STXIT должны быть опущены все операнды, кроме первого.

Макрокоманда EXIT осуществляет возврат из подпрограмм обработки прерываний по программным ошибкам (подпрограмма PC), прерываний по таймеру (подпрограмма IT) и прерываний по требованию оператора (подпрограмма ОС). Возврат в прерванную программу осуществляется к той команде, которая расположена непосредственно за командой, при выполнении которой произошло прерывание. Перед возвратом к прерванной программе из области сохранения восстанавливается запомненное PSW и содержимое общих регистров 0—15. Формат макрокоманды:

$$[\text{имя}]\quad \text{EXIT}\quad \left\{\begin{matrix}\text{PC}\\ \text{IT}\\ \text{OC}\end{matrix}\right\}$$

2.4.9. ВЫПОЛНЕНИЕ ОПЕРАЦИЙ ВВОДА-ВЫВОДА

Макрокоманда CCB создает блок управления командами ввода-вывода — блок CCB. Структура блока CCB:

Байты 0 — 1 — остаточный счетчик;
Байты 2 — 3 — байты связи (табл. 2.4);
Байты 4 — 5 — байты состояния из CSW;
значение бит байта 4 : 0 — внимание; 1 — модификатор; 2 — устройство управления кончило; 3 — занято; 4 — канал кончил; 5 — устройство кончило; 6 — сбой в устройстве; 7 — особый случай;
значение бит байта 5 : 0 — программно управляемое прерывание; 1 — неверная длина; 2 — программная ошибка; 3 — защита; 4 — ошибка в данных в канале; 5 — ошибка управления канала; 6 — ошибка управления интерфейса; 7 — ошибка зацепления;
Байты 6 — 7 —шифр логического устройства: SYSRDR = 0000, SYSIPT = 0001; SYSPCH = 0002, SYSLST = 0003, SYSLOG = 0004, SYSLNK = 0005, SYSRES = 0006, SYSSLB = 0007, SYSRLB = 0008, SYSREC = 000A, SYSCLB = 000B, SYS000 = 0100, SYS001 = 0101, ..., SYS243 = 01F3;
Байты 8 — 11 — адрес команды CCW, соответствующей этому CCB;
Байт 12 — используется программами физической СУВВ;
Байты 13 — 15 — адрес CCW из CSW, записанный по сигналу «Канал кончил», или адрес дополнительной подпрограммы канала для устройств телеобработки;
Байты 16 — 23 — байты команды CCW, которые добавляются к блоку CCB для запроса информации уточненного состояния.

Таблица 2.4

Байты связи

Бит	Название бита	Условия установки бита в 1	Значение в ССВ	Значение маски
		Байт 2		
0	Бит ожидания	Операция ввода-вывода завершена	—	X'80'
1	Бит конца файла	С SYSRDR или SYSIPT был прочитан оператор /* или /& (одновременно в 1 устанавливается бит «Особый случай» байта 4)	—	X'40'
2	Бит ошибки ввода-вывода	Операция ввода-вывода завершена с ошибкой. Устанавливается в 1 при возникновении ошибки, если существует подпрограмма обработки ошибок или соответствующая ошибка допустима. Программная ошибка ввода-вывода, ошибка защиты, ошибка из-за отвергнутой команды или ошибка оборудования магнитной ленты всегда ведут к снятию программы	—	X'20'
	Бит допустимости ошибки ввода-вывода	Проблемная программа допускает завершение операции ввода-вывода с ошибкой. При возникновении ошибки и неудачных попытках ее устранения будет установлен в 1 бит ошибки ввода-вывода	X'1000'	X'10'
4	Бит допустимости ошибки данных на дисках	Проблемная программа допускает завершение операции ввода-вывода на дисках с ошибкой данных	X'0800'	X'08'
5	Бит «Устройство кончило»	Проблемная программа требует, чтобы операция ввода-вывода завершилась по сигналу «Устройство кончило»	X'0400'	X'04'
6	Бит допустимости ошибки данных при чтении с магнитной ленты, ошибки оборудования при выводе на перфокарты, ошибки данных при выполнении команд чтения или записи на дисках	Проблемная программа допускает завершение операции чтения на магнитной ленте с ошибкой данных, операции вывода на перфокарты с ошибкой оборудования, операции чтения или записи на дисках с ошибкой данных	X'0200'	X'02'
7	Бит наличия подпрограммы обработки ошибок	Проблемная программа указывает, что у нее имеется подпрограмма для обработки ошибок ввода-вывода	X'0100'	X'01'
		Байт 3		
0	Бит ошибки данных в области счетчика на дисках	Операция ввода-вывода на дисках завершена с ошибкой данных в области счетчика	—	X'80'
1	Бит переполнения дорожки	Во время операции ввода-вывода на дисках возникло переполнение дорожки	—	X'40'
2	Бит конца цилиндра	Во время операции ввода-вывода на дисках был достигнут конец цилиндра	—	X'20'

Продолжение табл. 2.4

Бит	Название бита	Условия установки бита в 1	Значение в ССВ	Значение маски
3	Бит ошибки данных при чтении с магнитной ленты, ошибки оборудования при выводе на перфокарты, ошибки данных на дисках	Операция чтения с магнитной ленты завершена с ошибкой данных, операция вывода на перфокарты завершена с ошибкой оборудования, операция на дисках завершена с ошибкой данных	—	X'10'
4	Бит необычного состояния	Во время выполнения операции ввода-вывода обнаружена необычная последовательность команд канала или на дисках не найдена запись	—	X'08'
5	Бит повторения поиска записи	Проблемная программа требует, чтобы при ненахождении записи операция поиска повторялась. Если этот бит не установлен, то при ненахождении записи в 1 устанавливается бит необычного состояния	—	X'04'
6	Бит ошибки данных при записи на диски или бит кода 9 на управляющей перфоленте печати	Во время выполнения операции записи на диски возникла ошибка данных. Во время выполнения операции вывода на печать на управляющей перфоленте опознан код 9	—	X'02'
7	Бит повторения	Проблемная программа указывает, что при возникновении ошибки повторное выполнение операции должно начинаться с той команды, при выполнении которой возникла ошибка. Если этот бит имеет значение 0, то операция повторяется с начала канальной программы	X'0001'	X'01'

Примечание. В графе «Значение в ССВ» указывается значение третьего операнда макрокоманды ССВ для установки бита в 1, в графе «Значение маски» — значение маски для проверки состояния бита.

Блок ССВ предназначен для передачи от проблемной программы Супервизору информации, необходимой для выполнения операции ввода-вывода. Этот же блок используется Супервизором для передачи информации проблемной программе о результатах выполнения операции. Формат макрокоманды:

[имя] ССВ SYSxxx, адрес CCW [,X'nnnn'] [, адрес]

где SYSxxx — символическое имя логического устройства, на котором должна выполняться операция ввода-вывода;

адрес CCW — символическое имя первой команды CCW в цепочке команд, которая используется для выполнения операции ввода-вывода, соответствующей этому блоку ССВ;

X'nnnn' — шестнадцатеричная константа, используемая для установления бит в байтах связи блока ССВ;

адрес — символический адрес области, в которую заносится информация уточненного состояния устройства.

Макрокоманда EXCP указывает Супервизору о необходимости начать операцию ввода-вывода. Формат макрокоманды:

$$[\text{имя}]\ \ \text{EXCP}\ \ \left\{\begin{matrix}\text{блок CCB}\\(1)\end{matrix}\right\}$$

где блок CCB — символический адрес блока CCB.

Макрокоманда WAIT приостанавливает выполнение проблемной программы до тех пор, пока операция ввода-вывода, начатая по макрокоманде EXCP, не завершится. Формат макрокоманды:

$$[\text{имя}]\ \ \text{WAIT}\ \ \left\{\begin{matrix}\text{блок CCB}\\(1)\end{matrix}\right\}$$

2.4.10. ЗАВЕРШЕНИЕ ПРОГРАММ (ПОДЗАДАЧ)

Макрокоманда EOJ выдается в конце выполнения программы (подзадачи), чтобы сообщить Супервизору о нормальном завершении программы (подзадачи). Формат макрокоманды:

имя EOJ

Макрокоманда CANCEL снимает задание (подзадачу). Выдача макрокоманды CANCEL в основной задаче или выдача макрокоманды CANCEL с операндом ALL в любой задаче (как в основной, так и в подзадаче) вызывает снятие всего задания. Если выдается макрокоманда CANCEL без операнда и подзадаче, то снимается только подзадача. Формат макрокоманды:

[имя] CANCEL [ALL]

Макрокоманда DUMP прекращает выполнение программы или подзадачи и выполняет разгрузку памяти. Формат макрокоманды:

[имя] DUMP

2.4.11. РАЗГРУЗКА ОСНОВНОЙ ПАМЯТИ

Макрокоманда DUMP описана в 2.4.10.

Макрокоманда PDUMP разгружает указанную область основной памяти. Формат макрокоманды:

$$[\text{имя}]\ \ \text{PDUMP}\ \ \left\{\begin{matrix}\text{адрес 1}\\(r1)\end{matrix}\right\},\left\{\begin{matrix}\text{адрес 2}\\(r2)\end{matrix}\right\}$$

где адрес 1 — символический начальный адрес разгружаемой области;

адрес 2 — символический конечный адрес разгружаемой области.

2.4.12. СОЗДАНИЕ КОНТРОЛЬНОЙ ТОЧКИ

Макрокоманда CHKPT создает контрольную точку. Формат макрокоманды:

```
[имя]  CHKPT SYSnnn, {адрес повторного запуска}
                     {     (r1)              },

[{адрес конца}],[{Л-указатель}],
[{    (r2)  }]  [{     (r3)  }]
[{Д-указатель}],[{имя файла  }]
[{     (r4)  }]  [{     (r5)  }]
```

где SYSnnn — логическое устройство, на котором располагается файл контрольной точки;

адрес повторного запуска — символический адрес в проблемной программе, с которого начнется выполнение программы при повторном запуске;

адрес конца — символический адрес последнего байта в области проблемной программы, до которого включительно запоминается состояние основной памяти в момент создания контрольной точки;

Л-указатель — символический адрес области, содержащей адреса таблиц установки магнитных лент — две адресные константы. Первая константа указывает на таблицу установки магнитных лент, обрабатываемых логической системой управления вводом-выводом. Структура этой таблицы: 2 байта — количество строк в таблице (равно количеству файлов на магнитной ленте, для которых требуется установка при повторном запуске), далее строки длиной по 4 байта. Каждая строка — адресная константа, определяющая адрес DTF для устанавливаемого файла. Вторая константа указывает на таблицу установки магнитных лент, обрабатываемых физической системой управления вводом-выводом. Структура таблицы: 2 байта — количество строк в таблице, равное количеству файлов на магнитной ленте, для которых требуется установка при повторном запуске; далее следуют строки длиной по 6 байт. Структура строки: 2 байта — шифр логического устройства, на котором расположен соответствующий файл (из блока ССВ); 2 байта — количество ленточных марок от начала до той точки, в которой она находится в момент создания контрольной точки (в двоичном виде); 2 байта — количество физических записей между предыдущей ленточной маркой и точкой, в которой находится магнитная лента в момент создания контрольной точки (в двоичном виде);

Д-указатель — символический адрес таблицы установки пакетов дисков. Структура таблицы: 2 байта — количество строк в таблице, равное количеству логических устройств, для которых требуется контролировать правильность установки пакетов; далее следуют строки длиной по 4 байта. Структура строки: 2 байта — шифр логического устройства, на которое должен быть установлен пакет дисков (из ССВ); 2 байта — X'0000';

имя файла — символический адрес макрокоманды DTFPH, описывающей файл контрольной точки на дисках.

ГЛАВА 3

УПРАВЛЕНИЕ ДАННЫМИ

3.1. ОРГАНИЗАЦИЯ ДАННЫХ

3.1.1. ФОРМАТЫ ЛОГИЧЕСКИХ ЗАПИСЕЙ

Записи фиксированной длины (формат F). Логическая запись является записью фиксированной длины, если длина всех логических записей в пределах файла одна и та же. Записи фиксированной длины могут быть блокированными или неблокированными (рис. 3.1). Число логических записей во всех блоках, кроме последнего, обычно одинаково. В последнем блоке может быть меньше логических записей (усеченный блок).

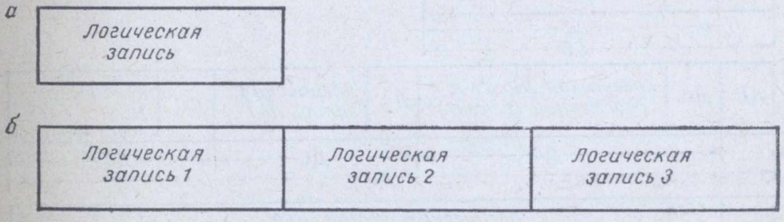

Рис. 3.1. Запись фиксированной длины (формат F):
а — неблокированная; *б* — блокированные (коэффициент блокирования равен 3)

Записи переменной длины (формат V). Логическая запись является записью переменной длины, если длина записей файла различна, и она указывается в каждой записи. Записи переменной длины могут быть блокированными или неблокированными (рис. 3.2).

Расширенные записи (формат V) — это логические записи переменной длины, которые могут записываться в одном или нескольких последовательно расположенных физических блоках. Расширенные записи могут быть блокированными или неблокированными. Если записи неблокированные, то физический блок может содержать полную логическую запись или сегмент логической записи, если блокированные — несколько логических записей или полную логическую запись и (или) сегмент логической записи (рис. 3.3).

Записи неопределенной длины (формат U). Любая логическая запись, которая не является ни записью фиксированной длины, ни записью переменной длины (нерасширенной или расширенной), считается записью неопределенной длины. Записи неопределенной длины могут быть только неблокированными.

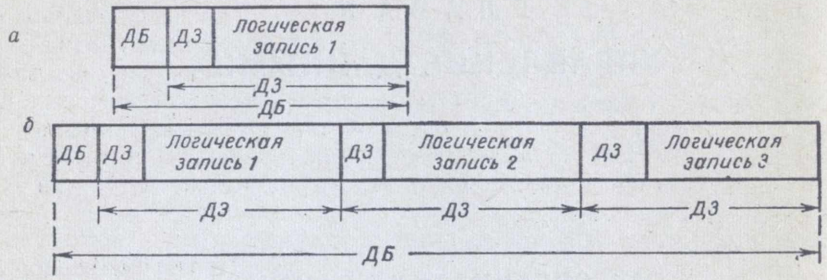

Рис. 3.2. Записи переменной длины (формат V):

а — неблокированная; *б* — блокированные (коэффициент блокирования равен 3); ДБ — длина блока, включая и длину 4-байтового поля ДБ, которое содержит длину физического блока в первых двух байтах поля; ДЗ — длина записи, включая и длину 4-байтового поля ДЗ, которое содержит длину логической записи в первых двух байтах поля

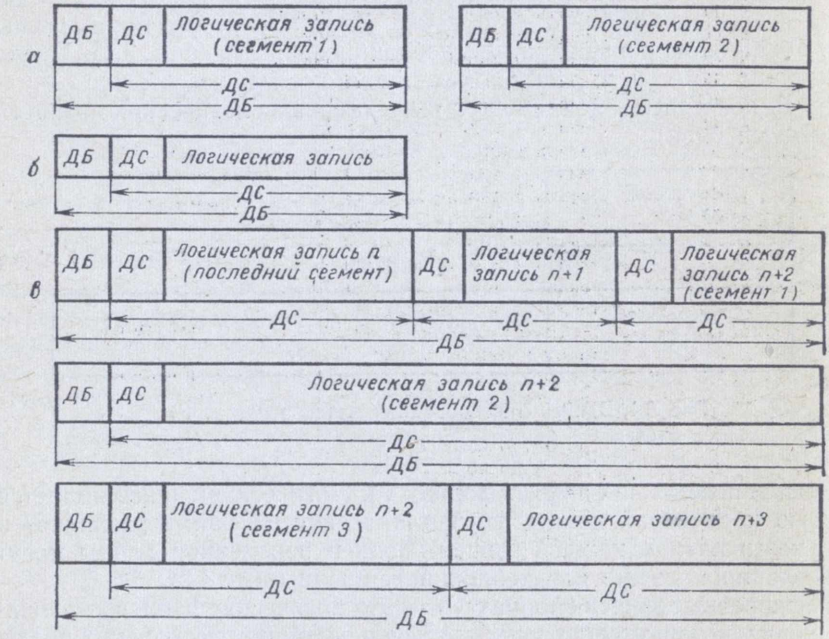

Рис. 3.3. Расширенные записи (формат V):

а — неблокированная запись, расположенная в двух физических блоках; *б* — неблокированная запись, расположенная в одном физическом блоке; *в* — блокированные записи; ДБ — длина блока, включая и длину 4-байтового поля ДБ, которое содержит длину физического блока в первых двух байтах поля, ДС — длина сегмента (или всей логической записи), включая и длину 4-байтового поля ДС, которое содержит длину сегмента в первых двух байтах поля и указатель положения сегмента (первый, промежуточный, последний сегмент логической записи) в следующих двух байтах поля

3.1.2. УПРАВЛЯЮЩИЕ СИМВОЛЫ

Эти символы управляют форматом листа на устройствах печати, снабженных управляющей перфолентой, или определяют карман для укладки отперфорированных карт. Они задают действие, которое необходимо выполнить при печати записи (печать и пропуск заданного числа строк, печать и прогон бумаги до заданного кода на управляющей перфоленте и т. п.) или при перфорации карты (выбор кармана для укладки карты). Управляющий символ должен быть: для логической записи фиксированной или неопределенной длины — первым символом; для записи переменной длины или расширенной записи — пятым символом. Управляющие символы могут быть трех типов: A, D или C (табл. 3.1—3.3).

Таблица 3.1

Управляющие символы типа A

Управляющий символ	Выполняемое действие
X'01'	Печать
X'09'	Печать и пропуск одной строки
X'11'	Печать и пропуск двух строк
X'19'	Печать и пропуск трех строк
X'89'	Печать и прогон до кода 1
X'91'	Печать и прогон до кода 2
X'99'	Печать и прогон до кода 3
X'A1'	Печать и прогон до кода 4
X'A9'	Печать и прогон до кода 5
X'B1'	Печать и прогон до кода 6
X'B9'	Печать и прогон до кода 7
X'C1'	Печать и прогон до кода 8
X'C9'	Печать и прогон до кода 9
X'D1'	Печать и прогон до кода 10
X'D9'	Печать и прогон до кода 11
X'E1'	Печать и прогон до кода 12
X'0B'	Пропуск одной строки
X'13'	Пропуск двух строк
X'1B'	Пропуск трех строк
X'8B'	Прогон до кода 1
X'93'	Прогон до кода 2
X'9B'	Прогон до кода 3
X'A3'	Прогон до кода 4
X'AB'	Прогон до кода 5
X'B3'	Прогон до кода 6
X'BB'	Прогон до кода 7
X'C3'	Прогон до кода 8
X'CB'	Прогон до кода 9
X'D3'	Прогон до кода 10
X'DB'	Прогон до кода 11
X'E3'	Прогон до кода 12
X'03'	Холостой ход
X'01'	Перфорация и выбор кармана 1
X'41'	Перфорация и выбор кармана 2

Примечание. Тип A не используется в программах и макрокомандах перезаписи при перфорации карт.

Таблица 3.2

Управляющие символы типа С

Управляющий символ	Выполняемое действие
+ (X'4E' или X'50')	Печать
Пробел	Печать и пропуск одной строки
0	Пропуск двух строк, печать и пропуск одной строки
—	Пропуск трех строк, печать и пропуск одной строки
1—9 или J — R	Прогон до кода 1—9, печать и пропуск одной строки (если 1 или J, то прогон до кода 1; если 2 или K, прогон до кода 2,...; если 9 или R, прогон до кода 9)

Примечание. Тип С не используется в логической СУВВ.

Таблица 3.3

Управляющие символы типа D

Управляющий символ	Выполняемое действие
Пробел	Пропуск одной строки и печать
0	Пропуск двух строк и печать
—	Пропуск трех строк и печать
+(X'4E' или X'50')	Печать
1	Прогон до кода 1 и печать
2	Прогон до кода 2 и печать
3	Прогон до кода 3 и печать
4	Прогон до кода 4 и печать
5	Прогон до кода 5 и печать
6	Прогон до кода 6 и печать
7	Прогон до кода 7 и печать
8	Прогон до кода 8 и печать
9	Прогон до кода 9 и печать
A	Прогон до кода 10 и печать
B	Прогон до кода 11 и печать
C	Прогон до кода 12 и печать
V	Перфорация и выбор кармана 1
W	Перфорация и выбор кармана 2

3.1.3. ФОРМАТЫ ДАННЫХ НА ПЕРФОЛЕНТЕ

Формат физических записей. Все компоненты ДОС ЕС, обеспечивающие работу с перфолентой, допускают физические записи неопределенного формата. В конце каждой физической записи неопределенного формата присутствует символ «Конец записи», являющийся признаком конца физической записи. Он набирается оператором на пульте управления устройства ввода. В качестве символа «Конец записи» может использоваться любой символ перфоленты, за исключением символа «Забой».

Физические записи фиксированного формата допускаются программами логической системы управления вводом-выводом (СУВВ) для любых кодов перфоленты и макрокомандами перезаписи для кодов без служебных символов. Программы перезаписи не обрабатывают физические записи фиксированного формата.

Количество строк на перфоленте может быть больше количества перфорируемых символов данных за счет строк, необходимых для служебных символов (например, признаков регистров). По этой причине длина физических записей на перфоленте может быть различной, даже если перфорируются записи фиксированного формата. Тем не менее они ничем не разделяются, и возможность разграничения записей должна быть предусмотрена программой пользователя.

Формат логических записей. Логическая СУВВ для перфоленты допускает только неблокированные записи фиксированной или неопределенной длины.

Программы и макрокоманды перезаписи допускают блокированные записи фиксированной или неопределенной длины, но для выполнения деблокирования в конце каждой логической записи должен присутствовать символ «Конец логической записи». Кроме того, программы перезаписи предполагают, что логические записи состоят из полей. Если записи неопределенной длины, то поля можно перфорировать не полностью, так как в конце каждого поля должен присутствовать символ «Конец поля».

3.1.4. ФОРМАТ И ЕМКОСТЬ ДОРОЖКИ ДИСКОВ

Формат дорожки. Каждая дорожка содержит: индексную точку, собственный адрес, запись 0, записи данных и промежутки (рис. 3.4). Индексная точка определяет физическое начало дорожки, собственный адрес — местоположение дорожки на пакете и ее состояние. Запись 0

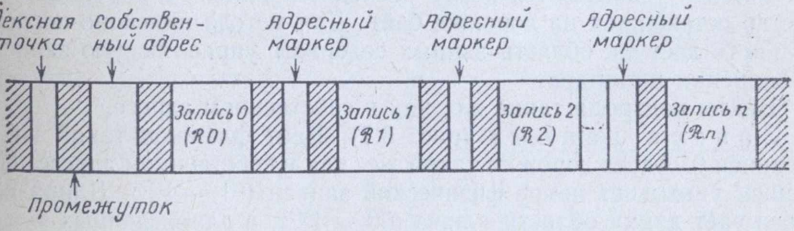

Рис. 3.4. Формат дорожки диска

используется для описания дорожки. Адресный маркер показывает начало физической записи (блока), кроме записи 0. Запись 1, запись 2, ..., запись n — это физические записи данных. Промежутки разделяют элементы дорожки.

Собственный адрес. На каждой дорожке имеется один собственный адрес (рис. 3.5). Байт флажков описывает состояние дорожки. После инициализации пакета биты 0—5 содержат нули, а биты 6 и 7 — следующую информацию: 00 — для недефектной основной и не-

назначенной запасной дорожки, 01 — для назначенной запасной дорожки, 10 — для дефектной основной дорожки, 11 — для дефектной запасной дорожки. Биты 6 и 7 автоматически включаются в байт флажков всех физических записей дорожки.

Адрес дорожки состоит из шестнадцатеричных номеров цилиндра и головки.

Байты контроля используются при циклическом контроле чтения и записи. Они записываются автоматически и в основную память не считываются.

З а п и с ь 0 (рис. 3.6). Байт флажков описывает состояние дорожки (биты 6 и 7 из байта флажков собственного адреса) и характеристику физической записи (биты 0 и 1): бит 0 равен 0 для четных записей и 1 для нечетных записей, бит 1 равен 0 для непереходящих записей и 1 для переходящих записей.

Байт флажков	Адрес дорожки				Байты контроля	
	Номер цилиндра CC		Номер головки HH			
1	2	3	4	5	6	7

Рис. 3.5. Структура собственного адреса

Адрес дорожки в области счетчика или совпадает с адресом дорожки собственного адреса, или указывает адрес запасной либо основной дорожки.

Номер записи и длина ключа равны 0, а длина данных — 8. Байты контроля такие же, как в собственном адресе.

Адрес дорожки в области данных либо совпадает с адресом дорожки в области счетчика, либо равен 0. Для метода прямого доступа в области данных указываются номер последней записи на дорожке и количество оставшихся на дорожке байт. Для метода индексно-последовательного доступа область данных содержит управляющую запись переполнения цилиндра.

Байты контроля такие же, как в собственном адресе.

З а п и с и д а н н ы х (рис. 3.7). Байт флажков такой же, как в записи 0. Адрес дорожки такой же, как в собственном адресе. Номер записи указывает номер физической записи (01 — FF). Длина ключа указывает длину области ключа (00 — FF), а длина данных — длину области данных (0000 — FFFF). Байты контроля такие же, как в собственном адресе.

Емкость дорожки. Максимальная емкость дорожки достигается в случае, если на дорожке, кроме записи 0, содержится только одна физическая запись без области ключа, занимающая всю дорожку. Служебная информация — адресные маркеры, области счетчика, байты циклического контроля, промежутки — увеличивает фактическую длину физических записей, уменьшая тем самым количество «полезных» байт, которые можно использовать для ключей и данных. Формулы для расчета фактической длины физических записей приведены в табл. 3.4.

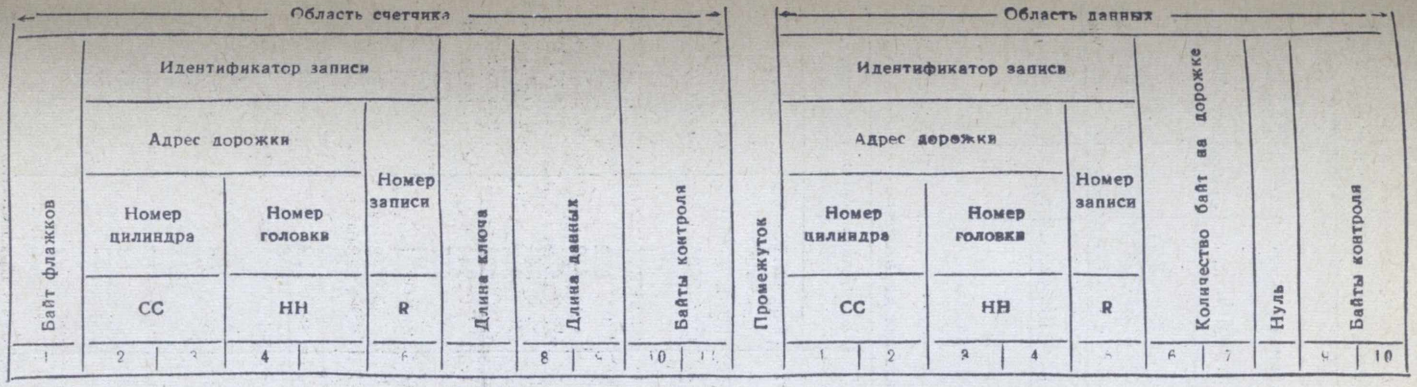

Рис. 3.6. Структура записи описания дорожки (записи 0)

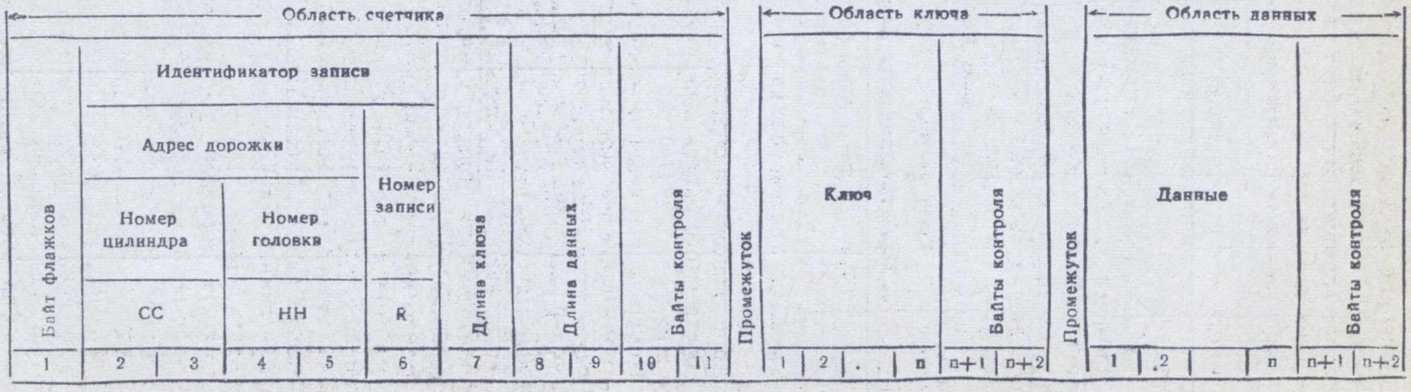

Рис. 3.7. Структура записей данных

Таблица 3.4

Фактическая длина физической записи

Шифр устройства	Максимальная емкость дорожки, байт	Фактическая длина, байт			
		любой записи данных, кроме последней		последней записи данных	
		без ключа	с ключом	без ключа	с ключом
ЕС-5052 ЕС-5055 ЕС-5056	3625	$61+\frac{537\cdot ДД}{512}$	$81+\frac{537(ДК+ДД)}{512}$	ДД	20+ДК+ДД
ЕС-5061	7294	$101+\frac{2137\cdot ДД}{2048}$	$146+\frac{2137(ДК+ДД)}{2048}$	ДД	45+ДК+ДД

Примечание. ДД — длина области данных, ДК — длина области ключа.

Таблица 3.5

Максимальная длина записей на дорожке

Количество записей на дорожке	Длина записи без ключа, байт, для устройств		Длина записи с ключом, байт, для устройств	
	ЕС-5052, ЕС-5055, ЕС-5056	ЕС-5061	ЕС-5052, ЕС-5055, ЕС-5056	ЕС-5061
1	3625	7294	3605	7249
2	1740	3520	1720	3476
3	1131	2298	1111	2254
4	830	1693	811	1649
5	651	1332	632	1288
6	632	1092	512	1049
7	447	921	428	877
8	384	793	364	750
9	334	694	315	650
10	295	615	275	571
11	263	550	244	506
12	236	496	217	452
13	213	450	194	407
14	193	411	174	368
15	177	377	158	333
16	162	347	143	304
17	149	321	130	277
18	138	298	119	254
19	127	276	108	233
20	118	258	99	215
21	109	241	90	198
22	102	226	82	183
23	95	211	76	168
24	88	199	69	156
25	82	187	63	144
26	77	176	58	133
27	72	166	53	123
28	67	157	48	114
29	63	148	44	105
30	59	139	40	96

Зная максимальную емкость дорожки и фактическую длину записей, можно вычислить количество записей заданного формата, вмещающихся на дорожке (K):

$$K = 1 + \frac{\text{Максимальная емкость дорожки} - \text{Фактическая длина последней записи}}{\text{Фактическая длина записи (непоследней)}}.$$

Для рационального использования емкости дорожки полезно знать соотношение между количеством записей на дорожке и максимальной длиной записей (табл. 3.5).

Емкость пакета дисков является произведением емкости дорожки (в байтах) на количество дорожек на пакете (табл. 3.6).

Таблица 3.6

Характеристика емкости пакета дисков

Характеристика	Шифр устройства	
	ЕС-5052, ЕС-5055, ЕС-5056	ЕС-5061
Количество основных цилиндров на пакете	200	200
Количество дорожек на цилиндре	10	20
Количество дорожек на пакете	2000	4000
Максимальная емкость дорожки, байт	3625	7294
Максимальная емкость цилиндра, байт	36250	145880
Максимальная емкость пакета, байт	7250000	29176000

3.1.5. СТРУКТУРА ФАЙЛОВ НА МАГНИТНОЙ ЛЕНТЕ И ДИСКАХ

Файл — это поименованная, организованная совокупность логически связанных данных. Единица носителя данных (катушка магнитной ленты, пакет дисков), на которой размещаются файлы, называется томом. Файлы могут располагаться на томах следующим образом:

однотомный файл — один файл на одном томе;

многотомный файл — один файл на нескольких томах;

многофайловый том — несколько файлов на одном томе;

многофайловая/многотомная совокупность — несколько файлов (все или часть которых разбивается на секции файла) на нескольких томах.

Для идентификации томов и файлов и защиты их от постороннего вмешательства или разрушения используются м е т к и. Они содержат информацию, необходимую программам системы управления вводом-выводом для распознавания, поиска и защиты данных. Метки могут присутствовать на магнитной ленте (рис. 3.8) и должны присутствовать на дисках (рис. 3.9).

Файлы на магнитной ленте могут содержать стандартные метки с фиксированной длиной (80 байт) и определенным форматом (рис. 3.10).

Файлы на магнитной ленте могут иметь также нестандартные метки, содержимое и формат которых полностью определяются пользователем (рис. 3.11). Можно использовать файлы без меток (рис. 3.12), когда на

ленте находятся только записи файлов, разделенных ленточными марками. В начале катушки может присутствовать ленточная марка, чтобы обеспечить возможность чтения файла в обратном направлении.

Файлы на дисках. Каждый файл располагается на одном или на нескольких участках. Участок — это совокупность дорожек одного или

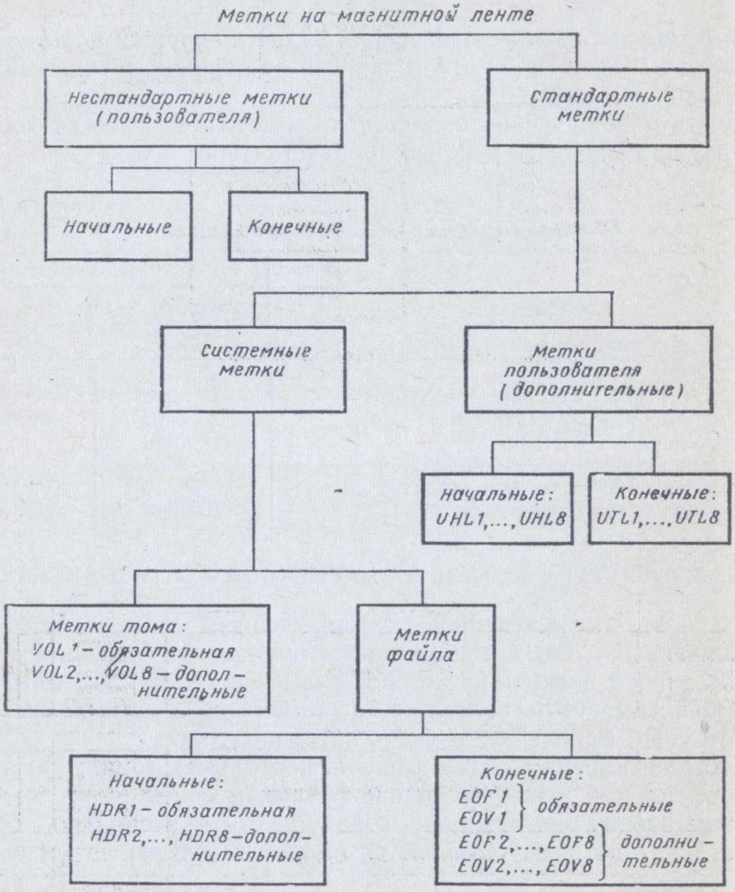

Рис. 3.8. Типы меток на магнитной ленте

нескольких последовательных цилиндров, которые предназначены для размещения всех записей файла или его части.

Для неразделенных цилиндров участок, занимающий несколько цилиндров, является непрерывным, т. е. он занимает все дорожки выделенных цилиндров. Для разделенных цилиндров участок занимает лишь определенные дорожки на последовательных цилиндрах и не является непрерывным. В обоих случаях границы участка указываются с помощью одного управляющего оператора EXTENT (см. 2.2). Во время

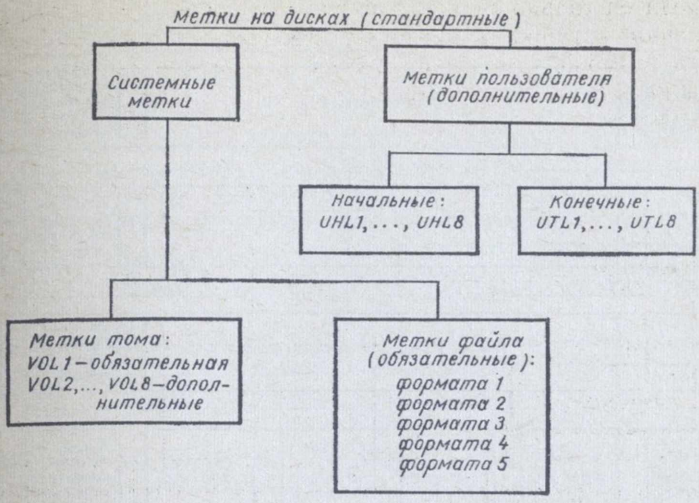

Рис. 3.9. Типы меток на дисках

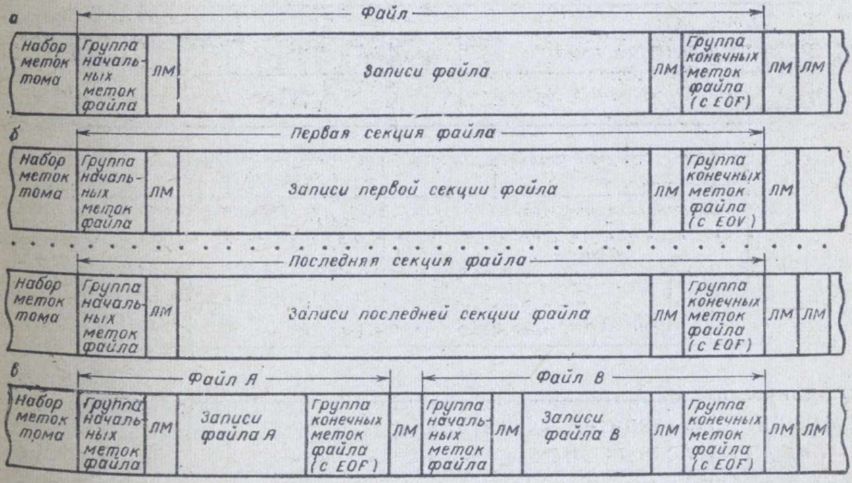

Рис. 3.10. Структура файлов со стандартными метками:

а — однотомный файл; *б* — многотомный файл; *в* — многофайловый том; ЛМ — обязательная ленточная марка: набор меток тома — VOL1, [VOL2], ..., [VOL8]; группа начальных меток файла — HDR1, [HDR2], ..., [HDR8], [UHL1], ..., [UHL8]; группа конечных меток файла — {[EOF1, [EOF2], ..., [EOF8]} / {[EOV1, [EOV2], ..., [EOV8]}, [UTL1], ..., [UTL8]

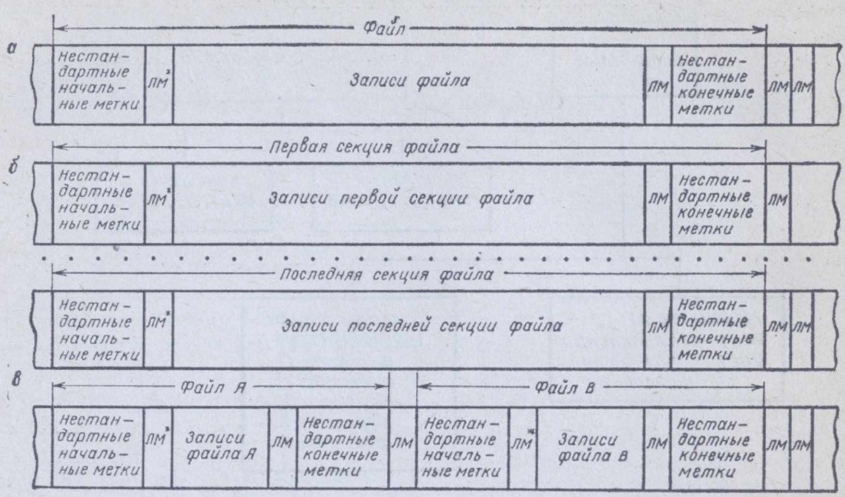

Рис. 3.11. Структура файлов с нестандартными метками:

а — одногомный файл; *б* — многотомный файл; *в* — многофайловый том; ЛМ* — необязательная ленточная марка; ЛМ — обязательная ленточная марка

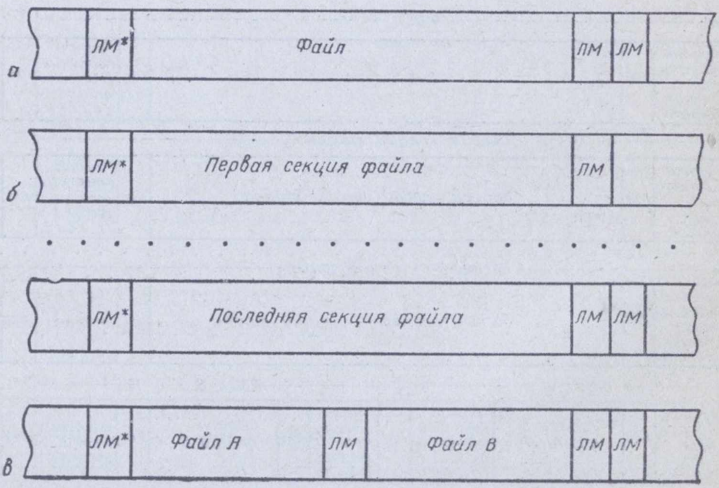

Рис. 3.12. Структура файлов без меток:

а — одногомный файл; *б* — многотомный файл; *в* — многофайловый том; ЛМ* — необязательная ленточная марка; ЛМ — обязательная ленточная марка

создания файла информация об участках записывается в системных метках.

Системные метки файлов, расположенных на пакете дисков, сосредоточены в оглавлении тома, а метки пользователя конкретного файла — на первой дорожке первого участка, занимаемого файлом. Формат и число меток, описывающих файл, определяются организацией файла и количеством участков файла.

Последовательные файлы и файлы прямого доступа содержат только данные, индексно-последовательные файлы сопровождаются еще и индексами, с помощью которых обеспечивается произвольный доступ к данным.

Системные метки файлов, записанных на пакете дисков, образуют оглавление тома, в котором содержится вся информация о занятости пакета. Адрес оглавления тома указывается меткой тома, а само оглавление описывается меткой формата 4, в которой помимо информации о расположении оглавления тома находятся также сведения о запасных дорожках пакета дисков. Обычно оглавление тома располагается на цилиндре 0 (для рабочих пакетов) или на цилиндре 199 (для резидентных пакетов). Размер оглавления не превышает одного цилиндра. Местоположение оглавления тома указывается во время инициализации пакета (см. 6.1). Распечатку оглавления тома можно выполнить с помощью программы Вывести оглавление тома (см. 6.3).

3.1.6. ОБРАБОТКА МЕТОК

Системные метки обрабатываются программами логической СУВВ. Информация, необходимая для обработки системных меток, сообщается системе с помощью управляющих операторов: для файлов на дисках — в операторах DLBL и EXTENT, для файлов на магнитных лентах — в операторе TLBL (см. 2.2).

Метки пользователя читаются и записываются программами логической СУВВ, а обрабатываются подпрограммой пользователя.

После чтения и перед записью каждой метки пользователя логическая СУВВ передает подпрограмме пользователя следующую информацию:

регистр 0 — букву O, если обрабатываются начальные метки файла; букву F, если обрабатываются конечные метки файла, следующие за метками EOF; букву V, если обрабатываются конечные метки файла, следующие за метками EOV;

регистр 1 — адрес прочитанной метки вводного файла или адрес области, где может строиться метка выводного файла.

После создания метки выводного файла подпрограмма пользователя должна сообщать ее адрес в регистре 0.

Подпрограмма пользователя должна возвращать управление логической СУВВ с помощью макрокоманды LBRET (см. 3.3.8):

LBRET 1, если обработанная метка последняя;

LBRET 2, если предполагается обработка следующих меток;

LBRET 3, если обновляется метка вводного файла на дисках.

Нестандартные метки читаются, записываются и обрабатываются подпрограммами пользователя.

Для вводных файлов логическая СУВВ передает подпрограмме пользователя следующую информацию:

регистр 0 — букву O, если должны обрабатываться начальные метки;

регистр 1 — шестнадцатеричное представление логического устройства.

После обработки конечных меток подпрограмма пользователя должна сообщать системе в регистре 0:

буквы EV, если обнаружен конец тома;

буквы EF, если обнаружен конец файла.

Для выводных файлов логическая СУВВ передает подпрограмме пользователя следующую информацию:

регистр 0 — букву O, если обрабатываются начальные метки; букву F, если обрабатываются конечные метки в конце файла; букву V, если обрабатываются конечные метки в конце секции многотомного файла (не последней);

регистр 1 — шестнадцатеричное представление логического устройства.

Подпрограмма пользователя должна возвращать управление логической СУВВ с помощью макрокоманды LBRET 2 (см. 3.3.8).

3.2. СИСТЕМА УПРАВЛЕНИЯ ВВОДОМ-ВЫВОДОМ

Совокупность программ, обеспечивающих обмен информацией между основной памятью и периферийными устройствами, называется системой управления вводом-выводом. Эти программы выполняют непосредственно чтение и запись, блокирование и деблокирование записей, обработку меток и ошибок. Существуют два уровня работы СУВВ: физический и логический. Каждый уровень реализуется совокупностью программ, называемых соответственно физической системой управления вводом-выводом и логической системой управления вводом-выводом.

3.2.1. ФИЗИЧЕСКАЯ СИСТЕМА УПРАВЛЕНИЯ ВВОДОМ-ВЫВОДОМ

Назначение и возможности. Физическая СУВВ:

- Обеспечивает выполнение операций ввода-вывода, осуществляющих непосредственно ввод физических записей с периферийных устройств в основную память и вывод их из основной памяти на периферийные устройства.
- Управляет выполнением канальных программ, подготовленных программистом и определяющих операцию ввода-вывода, обеспечивая при этом запуск канальной программы, ведение очереди запросов на выполнение операций ввода-вывода, если устройство занято, обработку завершения выполнения канальной программы, обнаруженных ошибок ввода-вывода, а также обработку прерываний ввода-вывода.
- Выдает программе пользователя информацию о результате выполнения операции ввода-вывода.

Используемые макрокоманды. CCB — создание блока управления командами ввода-вывода, EXCP — запрос выполнения канальной программы, WAIT — ожидание окончания операции ввода-вывода (см. 2.4.9).

Для файлов на магнитной ленте и дисках, обрабатываемых физической СУВВ, могут использоваться следующие макрокоманды логической СУВВ: DTFPH — описание файла для физической СУВВ, OPEN или OPENR — открытие файла, CLOSE или CLOSER — закрытие файла, LBRET — возврат управления из подпрограммы обработки меток пользователя, FEOV — запрос конца тома для логических устройств программиста, SEOV — запрос конца тома для системных устройств. Необходимость в использовании перечисленных макрокоманд возникает тогда, когда требуется обрабатывать метки для файлов на магнитных лентах и дисках либо должна обеспечиваться защита для файлов на дисках, либо файл контрольных точек должен создаваться на дисках.

Ниже описываются операнды макрокоманды DTFPH и макрокоманда SEOV, остальные макрокоманды логической СУВВ описаны в 3.3.8.

Макрокоманда DTFPH:

[CCWADDR = имя] — символический адрес канальной программы, используемой в CCB;

[DEVADDR = SYSxxx] — используемое логическое устройство, может быть задано символическое имя любого логического устройства (см. табл. 2.2.1);

DEVICE = $\left\{\begin{matrix}\text{TAPE}\\ \text{шифр}\end{matrix}\right\}$ — шифр конкретного устройства ввода-вывода, если описываемый файл на дисках (см. табл. 3.8), или TAPE — для файлов на магнитной ленте;

[HDRINFO = YES] — печать информации начальной метки файла на SYSLOG при открытии файла со стандартными метками на магнитной ленте, а также имени файла, логического устройства и адреса физического устройства при обнаружении конца тома;

[LABADDR = имя] — символический адрес подпрограммы пользователя, которой передается управление для обработки меток пользователя;

[MOUNTED = $\left\{\begin{matrix}\text{ALL}\\ \text{SINGLE}\end{matrix}\right\}$] — доступность участков для обработки: ALL — участки доступны одновременно, SINGLE — участки доступны последовательно;

TYPEFLE = $\left\{\begin{matrix}\text{INPUT}\\ \text{OUTPUT}\end{matrix}\right\}$ — тип описываемого файла: INPUT — вводной, OUTPUT — выводной;

[XTNTXIT = имя] — символический адрес подпрограммы пользователя, которой при открытии файла передается управление для обработки информации о каждом доступном участке файла. При этом в регистре 1 передается адрес области, которая содержит информацию об участке (см. табл. 3.13).

Макрокоманда	Символы стандартных имен модулей и их значения				
	Префикс имени	Четвертый символ		Пятый символ	
		Значение	Операнды	Значение	Операнды
CDMOD	IJC	F	RECFORM=FIXUNB	A	CTLCHR=ASA
		V	RECFORM=VARUNB	Y	CTLCHR=YES
		U	RECFORM=UNDEF	C	CONTROL=YES
				Z	CTLCHR и CONTROL опущены
PRMOD	IJD	F	RECFORM=FIXUNB	A	CTLCHR=ASA
		V	RECFORM=VARUNB	Y	CTLCHR=YES
		U	RECFORM=UNDEF	C	CONTROL=YES
				Z	CONTROL и CTLCHR опущены
PTMOD	IJE	S	SCAN=YES	T	TRANS=YES
		Z	SCAN опущен	Z	TRANS опущен
DIMOD	IJJ	F	Всегда	C	Всегда
MTMOD (вводные и выводные файлы)	IJF	F	RECFORM={FIXUNB / FIXBLK}	B	READ=BACK
		V	RECFORM={VARUNB / VARBLK}	Z	READ=FORWARD или READ опущен
		U	RECFORM=UNDEF		
		S	RECFORM={SPNUNB / SPNBLK}		
MTMOD (рабочие файлы)	IJF	W	Всегда	E	ERROPT=YES
				Z	ERROPT опущен

Таблица 3.7

модулей ввода-вывода

в зависимости от операндов макрокоманд xxMOD

Шестой символ		Седьмой символ		Восьмой символ	
Значение	Операнды	Значение	Операнды	Значение	Операнды
I	TYPEFLE=INPUT	I	IOAREA2=YES и WORKA опущен	0	DEVICE=7010 или 6016
O	TYPEFLE=OUTPUT			2	DEVICE=7012
H	TYPEFLE=INPUT, RDONLY=YES	W	WORKA=YES и IOAREA2 опущен	3	DEVICE=6012
N	TYPEFLE=OUTPUT, RDONLY=YES	B	IOAREA2=YES и WORKA=YES	4	DEVICE=7010 и CRDERR=RETRY
		Z	IOAREA2 и WORKA опущены	5	DEVICE=7012 и CRDERR=RETRY
P	PRINTOV=YES	I	IOAREA2=YES	V	WORKA=YES и RDONLY=YES
Z	PRINTOV опущен	Z	IOAREA2 опущен	W	WORKA=YES
				Y	RDONLY=YES
				Z	WORKA и RDONLY опущены
F	RECFORM=FIXUNB, SCAN=YES и TYPEFLE=INPUT	Z	DEVICE=6022 или 7902R	Z	Всегда
U	RECFORM=UNDEF, SCAN=YES и TYPEFLE=INPUT	2	DEVICE=7022 или 7902P		
Z	SCAN опущен и (или) TYPEFLE=OUTPUT				
B	TYPEFLE=OUTPUT	I	IOAREA2=YES	C	RDONLY=YES
I	TYPEFLE=INPUT	Z	IOAREA2 опущен	D	RDONLY опущен
C	CKPTREC=YES	W	WORKA=YES	M	ERREXT=YES и RDONLY=YES
Z	CKPTREC опущен	Z	WORKA опущен	N	ERREXT=YES
				Y	RDONLY=YES
				Z	ERREXT и RDONLY опущены
N	NOTEPNT=YES	Z	Всегда	M	ERREXT=YES и RDONLY=YES
S	NOTEPNT=POINTS			N	ERREXT=YES
Z	NOTEPNT опущен			Y	RDONLY=YES
				Z	RDONLY и ERREXT опущены

Символы стандартных имен модулей и их значения

Макрокоманда	Префикс имени	Четвертый символ		Пятый символ	
		Значение	Операнды	Значение	Операнды
SDMOD (вводные и выводные файлы)	IJG	C	для SDMODFx с HOLD=YES	U	для SDMODxU
				I	для SDMODxI
		F	для SDMODFx без HOLD	O	для SDMODxO
		R	для SDMODUx с HOLD=YES		
		U	для SDMODUx без HOLD		
		P	для SDMODVx с RECFORM={SPNUNB / SPNBLK} и HOLD=YES		
		Q	для SDMODVx с RECFORM={SPNUNB / SPNBLK} без HOLD		
		S	для SDMODVx с HOLD=YES		
		V	для SDMODVx без HOLD		
SDMOD (рабочие файлы)	IJG	T	HOLD=YES	C	ERROPT=YES и ERREXT=YES
		W	HOLD опущен	F	ERROPT=YES
				Z	ERROPT и ERREXT опущены
DAMOD	IJI	F	RECFORM=FIXUNB	A	AFTER=YES
DAMODV		B	RECFORM=UNDEF	Z	AFTER опущен
		S	RECFORM=SPNUNB		
		V	RECFORM=VARUNB		
ISMOD	IJH	A	RECFORM=BOTH и IOROUT=ADD или ADDRTR	A	IOROUT=ADDRTR
				I	IOROUT=ADD
		B	RECFORM=FIXBLK и IOROUT=ADD или ADDRTR	R	IOROUT=RETRVE
				L	IOROUT=LOAD
		U	RECFORM=FIXUNB и IOROUT=ADD или ADDRTR		
		Z	RECFORM опущен и IOROUT=LOAD или RETRVE		

Продолжение табл. 3.7

В зависимости от операндов макрокоманд xxMOD

Шестой символ		Седьмой символ		Восьмой символ	
Значение	Операнды	Значение	Операнды	Значение	Операнды
C	ERROPT=YES и ERREXT=YES	M	TRUNCS=YES и FEOVD=YES	B	CONTROL=YES и RDONLY=YES
E	ERROPT=YES	T	TRUNCS=YES	C	CONTROL=YES
Z	ERROPT и ERREXT опущены	W	FEOVD=YES	Y	RDONLY=YES
		Z	TRUNCS и FEOVD опущены	Z	CONTROL и RDONLY опущены
N	NOTEPNT=YES	C	CONTROL=YES	T	UPDATE=YES и RDONLY=YES
R	NOTEPNT=POINTRW	Z	CONTROL опущен	U	UPDATE=YES
Z	NOTEPNT опущен			Y	RDONLY=YES
				Z	UPDATE и RDONLY опущены
E	IDLOC=YES и FEOVD=YES	H	ERREXT=YES и RELTRK=YES	W	HOLD=YES и RDONLY=YES
I	IDLOC=YES	P	ERREXT=YES	Y	RDONLY=YES
R	FEOVD=YES	R	RELTRK=YES	X	HOLD=YES
Z	FEOVD и IDLOC опущены	Z	RELTRK и ERREXT опущены	Z	RDONLY и HOLD опущены
B	TYPEFLE=RANSEQ	B	CORINDX=YES и HOLD=YES	F	CORDATA=YES, ERREXT=YES и RDONLY=YES
G	IOAREA2=YES и TYPEFLE=SEQNTL или IOROUT=LOAD	C	CORINDX=YES	G	CORDATA=YES и ERREXT=YES
		O	HOLD=YES	O	CORDATA=YES и RDONLY=YES
R	TYPEFLE=RANDOM	Z	CORINDX и HOLD опущены	P	CORDATA=YES
S	TYPEFLE=SEQNTL			S	ERREXT=YES и RDONLY=YES
Z	TYPEFLE и IOAREA2 опущены			T	ERREXT=YES
				Y	RDONLY=YES
				Z	CORDATA, ERREXT и RDONLY опущены

Макрокоманда SEOV обеспечивает переключение на сменное устройство для вводных системных файлов на магнитной ленте, т. е. если логическому устройству SYSLST или SYSPCH назначена магнитная лента. Формат макрокоманды:

[имя] SEOV имя файла

3.2.2. ЛОГИЧЕСКАЯ СИСТЕМА УПРАВЛЕНИЯ ВВОДОМ-ВЫВОДОМ

Логическая СУВВ предоставляет пользователю несколько стандартных способов организации данных в файлы и методов обработки организованных данных.

Способ организации данных в совокупности с методом их обработки образует определенный метод доступа. В ДОС ЕС существуют три метода доступа: последовательный, прямой и индексно-последовательный.

Метод последовательного доступа обеспечивает обработку записей в последовательном порядке. Предполагается, что записи в файле организованы последовательно. Последовательная обработка применяется к файлам на магнитной ленте, перфокартах, дисках, печати, пишущей машинке и перфоленте.

Метод прямого доступа обеспечивает обработку записей в произвольном порядке и используется только для файлов на дисках.

Метод индексно-последовательного доступа обеспечивает обработку записей в последовательном и произвольном порядке и используется только для файлов на дисках.

Для каждого из методов доступа программисту на Ассемблере предоставляется набор макрокоманд: декларативных и императивных. Декларативные макрокоманды предназначены для описания файлов (макрокоманды DTFxx) и описания модулей ввода-вывода (макрокоманды xxMOD), а императивные — для запроса функций по обработке файла, выполняемых модулями ввода-вывода.

Императивные макрокоманды, а также макрокоманды описания файлов включаются программистом непосредственно в исходную программу на Ассемблере. Модули ввода-вывода обычно транслируются отдельно, образуя самостоятельные поименованные объектные модули, которые объединяются с программой пользователя на этапе редактирования. Модули ввода-вывода, реализующие наиболее часто используемые функции, поставляются пользователям в библиотеке объектных модулей. Возможности этих модулей однозначно определяются стандартными именами, которые присваиваются модулям во время трансляции на основании операндов макрокоманд xxMOD (табл. 3.7).

3.3. МЕТОД ПОСЛЕДОВАТЕЛЬНОГО ДОСТУПА

3.3.1. ФАЙЛЫ НА ПЕРФОКАРТАХ

Назначение и возможности. Метод последовательного доступа для файлов на перфокартах:

- Обрабатывает вводные файлы с неблокированными записями фиксированной длины и выводные файлы с неблокированными записями

фиксированной или переменной длины. Поле длины записи и управляющий символ не выводятся на перфокарты.

- Совмещает обработку записи с выполнением операции ввода-вывода, если используются две области ввода-вывода и (или) рабочая область.
- Помещает все перфорируемые карты в заданный карман или управляет выбором кармана для каждой перфокарты с помощью управляющего символа или макрокоманды CNTRL.
- Обрабатывает ошибки перфорации, обеспечивая в случае ошибки перфорации две возможности: снятие задания или повторение вывода перфокарты.
- Определяет конец вводного файла и передает управление подпрограмме пользователя для завершения обработки файла.

Используемые макрокоманды. Декларативные: DTFCD — описание файла на перфокартах, CDMOD — описание модуля ввода-вывода для файлов на перфокартах; императивные (см. 3.3.8).

Ниже описываются операнды для каждой декларативной макрокоманды.

Макрокоманда DTFCD:

$\left[\text{BLKSIZE} = \left\{\begin{matrix} n \\ 80 \end{matrix}\right\}\right]$ — длина области ввода-вывода (логической записи), включая управляющий символ выбора кармана, если он используется, и поле длины записи в случае записей переменной длины;

[CONTROL = YES] — использование макрокоманды CNTRL для выбора приемного кармана. Если операнд CONTROL задан, то должен отсутствовать операнд CTLCHR;

[CRDERR = RETRY] — повторение вывода перфокарты в случае ошибки перфорации. Если операнд опущен, задание автоматически снимается при ошибке перфорации;

$\left[\text{CTLCHR} = \left\{\begin{matrix} \text{YES} \\ \text{ASA} \end{matrix}\right\}\right]$ — наличие управляющего символа выбора кармана в записях выводного файла: YES — управляющие символы типа A, ASA — управляющие символы типа D. Если операнд CTLCHR задан, то должен отсутствовать операнд CONTROL;

DEVADDR = SYSxxx — используемое логическое устройство: SYSRDR, SYSIPT, SYSPCH или SYSnnn;

DEVICE = шифр — шифр используемого устройства ввода-вывода (см. табл. 3.8);

[EOFADDR = имя] — символический адрес в программе пользователя, по которому передается управление при обнаружении конца вводного файла, т. е. при считывании карты, содержащей символы /*, /& для SYSIPT или SYSDR или /* (для логических устройств программиста) в двух первых колонках;

[IOAREA1 = имя — символический адрес первой области ввода-вывода. Длина ее должна быть не меньше длины, указанной в операнде BLKSIZE;

Таблица 3.8

Периферийные устройства, работа с которыми обеспечивается программами СУВВ

Устройство	Шифр устройства	Макрокоманды СУВВ
Накопитель на магнитных дисках	ЕС-5052 ЕС-5055 ЕС-5056 ЕС-5061	DTFDA, DAMOD, DTFSD, SDMODxx DAMODV, DTFPH DTFIS, ISMOD
Накопитель на магнитной ленте	ЕС-5010 ЕС-5012 ЕС-5016 ЕС-5017 ЕС-5019	DTFMT MTMOD
Перфокарточное устройство ввода	ЕС-6012 ЕС-6016	DTFCD CDMOD
Перфокарточное устройство вывода	ЕС-7010 ЕС-7012	DTFCD CDMOD
Печатающее устройство	ЕС-7030 ЕС-7031 ЕС-7032 ЕС-7033 ЕС-7035	DTFPR PRMOD
Перфоленточное устройство ввода	ЕС-6022 ЕС-7902R	DTFPT PTMOD
Перфоленточное устройство вывода	ЕС-7022 ЕС-7902P	DTFPT PTMOD
Пишущая машинка	ЕС-7070 ЕС-7073 ЕС-7073А ЕС-7074 ЕС-7077	DTFGN

[IOAREA2 = имя] — символический адрес второй области ввода-вывода. Длина ее должна быть равна длине первой области ввода-вывода;

[IOREG = (r)] — номер регистра, в который загружается адрес записи вводного файла или адрес области для построения очередной записи выводного файла, если используются две области ввода-вывода и не используется рабочая область;

[MODNAME = имя] — имя модуля ввода-вывода, который будет использоваться для обработки файла, описанного данной макрокомандой;

[RDONLY = YES] — использование повторно-входимого модуля ввода-вывода;

[RECFORM = формат] — формат записей описываемого файла: FIXUNB — неблокированные фиксированной длины, VARUNB — неблокированные переменной длины (только для выводных файлов), UNDEF — неопределенной длины (только для выводных файлов). Если операнд отсутствует, предполагается RECFORM = FIXUNB;

[RECSIZE = (r)] — номер регистра, в который пользователь должен загружать длину выводной записи неопределенной длины;

[SEPASMB = YES] — отдельная от программы пользователя трансляция макрокоманды описания файла;

$\left[\text{SSELECT} = \left\{\begin{matrix} n \\ 1 \\ - \end{matrix}\right\}\right]$ — номер приемного кармана, в который должны укладываться перфорируемые карты (n равно 1 или 2);

$\left[\text{TYPEFLE} = \left\{\begin{matrix} \text{INPUT} \\ \text{OUTPUT} \end{matrix}\right\}\right]$ — тип описываемого файла: INPUT — вводной, OUTPUT — выводной;

[WORKA = YES] — использование рабочей области для обработки записей файла.

Макрокоманда CDMOD:

[CONTROL = YES] — выполнение функций макрокоманды CNTRL;

[CRDERR = RETRY] — повторение вывода перфокарты в случае ошибки перфорации;

$\left[\text{CTLCHR} = \left\{\begin{matrix} \text{YES} \\ \text{ASA} \end{matrix}\right\}\right]$ — обработка выводных файлов с записями, имеющими управляющие символы типа A (YES) или типа D (ASA);

DEVICE = шифр — шифр используемого устройства ввода-вывода. Указывается тот же шифр, что и в описании файла;

[IOAREA2 = YES] — использование двух областей ввода-вывода для обработки записей файла;

[RDONLY = YES] — обеспечение повторной входимости модуля ввода-вывода;

[RECFORM = формат] — формат записей обрабатываемых файлов: FIXUNB — неблокированные фиксированной длины, VARUNB — неблокированные переменной длины, UNDEF — неопределенной длины. Если операнд отсутствует, предполагается RECFORM = =FIXUNB;

[SEPASMB = YES] — отдельная от программы пользователя трансляция макрокоманды описания модуля ввода-вывода;

$\left[\text{TYPEFLE} = \left\{\begin{matrix} \text{INPUT} \\ \text{OUTPUT} \end{matrix}\right\}\right]$ — тип обрабатываемых файлов: INPUT — вводной, OUTPUT — выводной;

[WORKA = YES] — использование рабочей области для обработки записей файла.

3.3.2. ФАЙЛЫ НА ПЕЧАТИ

Назначение и возможности. Метод последовательного доступа для файлов на печати:

- Обрабатывает файлы, организуя их вывод на печать, с неблокированными записями фиксированной или переменной длины или с записями неопределенной длины. Поле длины записи и управляющий символ печати не печатаются.
- Совмещает обработку с выполнением операции вывода, если используются две области вывода и (или) рабочая область.
- Выполняет операции управления форматом листа (пропуск строк или прогон бумаги до соответствующего кода управляющей перфоленты), заданные управляющим символом печати или макрокомандой CNTRL.
- Определяет конец листа и передает управление подпрограмме пользователя, которая может напечатать желаемую информацию в конце текущего листа и (или) в начале следующего листа.

Используемые макрокоманды. Декларативные: DTFPR — описание файла на печати, PRMOD — описание модуля ввода-вывода для файлов на печати; императивные (см. 3.3.8). Ниже описываются операнды для каждой декларативной макрокоманды.

Макрокоманда DTFPR:

$\left[\text{BLKSIZE} = \left\{\begin{matrix} \text{n} \\ \underline{121} \end{matrix}\right\}\right]$ — длина области вывода (логической записи), включая управляющий символ печати, если он используется, и поле длины записи в случае записей переменной длины;

[CONTROL = YES] — использование макрокомады CNTRL для управления форматом печати. Если операнд CONTROL задан, то должен отсутствовать операнд CTLCHR;

$\left[\text{CTLCHR} = \left\{\begin{matrix} \text{YES} \\ \text{ASA} \end{matrix}\right\}\right]$ — наличие управляющего символа печати в записях: YES — управляющие символы типа A, ASA — управляющие символы типа D. Если операнд CTLCHR задан, то должен отсутствовать операнд CONTROL;

DEVADDR = SYSxxx — используемое логическое устройство: SYSLST, SYSLOG или SYSnnn;

[DEVICE = шифр] — шифр используемого устройства вывода (см. табл. 3.8). Если операнд отсутствует, предполагается устройство EC-7030;

[PRINTOV = YES] — использование макрокоманды PRTOV;

[TRACK = RIGNT] — печать логической записи на правой полосе бумаги для устройства EC-7031;

IOAREA1 = имя, [IOAREA2 = имя], [IOREG = (r), [MODNAME = имя], [RDONLY = YES], [RECFORM = формат], [RECSIZE = (r)], [SEPASMB = YES], [WORKA = YES] — см. 3.3.1.

Макрокоманда PRMOD:

[DEVICE = шифр] — шифр используемого устройства вывода

(см. табл. 3.8). Указывается тот же шифр, что и в описании файла. Если операнд отсутствует, предполагается устройство EC-7030;

[PRINTOV = YES] — выполнение функций макрокоманды PRTOV;

$$[\text{CONTROL} = \text{YES}],\ \left[\text{CTLCHR} = \begin{Bmatrix}\text{YES}\\ \text{ASA}\end{Bmatrix}\right],\quad [\text{IOAREA2} = \text{YES}],$$

[RECFORM = формат], [RDONLY = YES], [SEPASMB = YES], [WORKA = YES] — см. 3.3.1.

3.3.3. ФАЙЛЫ НА ПЕРФОЛЕНТЕ

Назначение и возможности. Метод последовательного доступа для файлов на перфоленте:

- Обрабатывает файлы с неблокированными записями фиксированной или неопределенной длины, обеспечивая ввод логических записей с перфоленты и вывод их на перфоленту. Записи неопределенной длины должны разграничиваться символом «Конец записи» (символом EOR): для вводных файлов код этого символа набирается на пульте устройства, для выводных файлов символ EOR перфорируется в конце каждой записи.
- Обрабатывает пяти-, шести-, семи- и восьмидорожковые перфоленты, использующие одно-или двухрегистровый код: для вводных файлов выполняет перекодировку информации из кода перфоленты в ДКОИ, удаляя при этом служебные символы; для выводных файлов — из ДКОИ в код перфоленты, вставляя при этом служебные символы. Перекодировка осуществляется с помощью таблиц, поставляемых пользователем.
- Обрабатывает однотомные и многотомные файлы и многофайловые тома.
- Совмещает обработку с выполнением операций ввода-вывода, если используются две области ввода-вывода.
- Обрабатывает ошибки ввода-вывода, обеспечивая возможность игнорировать ошибку и продолжать обработку, пропускать ошибочную запись или обрабатывать ошибку с помощью подпрограммы пользователя.
- Определяет записи неверной длины для вводных файлов и обеспечивает возможность обрабатывать их с помощью подпрограммы пользователя.
- Определяет конец вводного файла и передает управление подпрограмме пользователя для завершения обработки файла.

Используемые макрокоманды. Декларативные: DTFPT — описание файла на перфоленте, PTMOD — описание модуля ввода-вывода для файлов на перфоленте; императивные — см. 3.3.8.

Ниже описываются операнды для каждой декларативной макрокоманды.

Макрокоманда DTFPT:

BLKSIZE=n — длина логической записи для записей фиксированной длины и длина наибольшей физической записи для записей не-

определенной длины. Для вводных записей неопределенной длины указывается длина наибольшей физической записи плюс единица;

[DELCHAR = X'nn'] — код символа «Забой» (пробивки по всем дорожкам), использующегося для исправления ошибочных символов; указывается только для выводных устройств, имеющих аппаратуру для исправления ошибочных символов;

DEVADDR = SYSnnn — используемое логическое устройство;

[DEVICE = шифр] — шифр используемого устройства ввода-вывода (см. табл. 3.8). Если операнд отсутствует, предполагается устройство EC-6022;

[EOFADDR = имя] — символический адрес в программе пользователя, по которому передается управление при обнаружении конца вводного файла, т. е. при считывании не менее 25 см чистой перфоленты, если нажата клавиша «Конец картотеки»;

[EORCHAR = X'nn'] — код символа EOR, являющегося концом записи, при выводе файлов с записями неопределенной длины. Он не должен совпадать ни с одним из символов записи;

$$\left[\text{ERROPT} = \begin{Bmatrix}\text{IGNORE}\\ \text{SKIP}\\ \text{имя}\end{Bmatrix}\right]$$ — режим обработки ошибок ввода-вывода: IGNORE — ошибка игнорируется и обработка продолжается. Для вводных файлов запись с ошибкой также предоставляется пользователю для обработки; SKIP — ошибочная запись пропускается, считывается следующая запись. Для выводных файлов этот параметр не используется; имя — символический адрес подпрограммы пользователя, которой передается управление при обнаружении ошибки ввода-вывода. Для вводных файлов этот операнд используется только для устройств, имеющих аппаратуру исправления ошибочных символов;

IOAREA1 = имя — символический адрес первой области ввода-вывода. Ее длина должна быть не меньше длины, указанной в операнде BLKSIZE или OVBLKSZ, если операнд OVBLKSZ используется;

[FSCAN = имя] — символический адрес таблицы, используемой для выделения цифровой информации в записях выводных файлов. Во всех байтах таблицы, адреса которых относительно ее начала равны кодам буквенных символов, должен стоять признак буквенного регистра. Все остальные байты — X'00';

[FTRANS = имя] — символический адрес таблицы перекодировки цифровой информации в записях вводных файлов. Таблица должна удовлетворять требованиям команды TR;

[LSCAN = имя] — символический адрес таблицы, используемой для выделения буквенной информации в записях выводных файлов. Во всех байтах таблицы, адреса которых относительно ее начала равны кодам цифровых символов, должен стоять признак цифрового регистра. Все остальные байты — X'00';

[LTRANS = имя] — символический адрес таблицы перекодировки буквенной информации в записях вводных файлов. Таблица должна удовлетворять требованиям команды TR;

[OVBLKSZ = n] — длина области ввода-вывода для записей фиксированной длины. Если этот операнд опущен, то длина области ввода-вывода определяется операндом BLKSIZE;

[RECFORM = формат] — формат записей описываемого файла: FIXUNB — неблокированные фиксированной длины, UNDEF — неопределенной длины. Если операнд отсутствует, предполагается RECFORM = FIXUNB;

[RECSIZE = (r)] — номер регистра, в который пользователь должен загружать длину выводной логической записи или в который помещается длина вводной логической записи при обработке записей неопределенной длины;

[SCAN = имя] — символический адрес таблицы, используемой для определения служебных символов в записях вводных файлов. В байтах таблицы, адреса которых относительно ее начала равны кодам служебных символов, должен находиться один из следующих кодов: X'04' — для признака цифрового регистра, X'08' — для признака буквенного регистра, X'0C' — для символов удаления, X'10' — для недопустимых символов;

[TRANS = имя] — символический адрес таблицы перекодировки информации в записях вводных файлов (без признаков регистров) или выводных файлов;

[WLRERR = имя] — символический адрес подпрограммы пользователя, которой передается управление при считывании записей неверной длины;

[IOAREA2 = YES], [IOREG = (r)], [MODNAME = имя], [SEPASMB=YES], $\left[\text{TYPEFLE} = \left\{\frac{\text{INPUT}}{\text{OUTPUT}}\right\}\right]$ — см. 3.3.1.

Макрокоманда PTMOD:

[DEVICE = шифр] — шифр используемого устройства ввода-вывода. Указывается тот же шифр, что и в описании файла. Если операнд отсутствует, предполагается устройство ЕС-6022;

[RECFORM = формат] — формат записей обрабатываемых файлов: FIXUNB — неблокированные фиксированной длины, UNDEF — неопределенной длины. Если операнд отсутствует, предполагается RECFORM = FIXUNB;

[SCAN = YES] — перекодировка информации, содержащей служебные символы в записях вводных и выводных файлов;

[TRANS = YES] — перекодировка информации, не содержащей служебных символов в записях вводных файлов;

[SEPASMB = YES], $\left[\text{TYPEFLE} = \left\{\frac{\text{INPUT}}{\text{OUTPUT}}\right\}\right]$ — см. 3.3.1.

3.3.4. ФАЙЛЫ НА ПИШУЩЕЙ МАШИНКЕ

Назначение и возможности. Метод последовательного доступа для файлов на пишущей машинке:

- Обрабатывает файлы с неблокированными записями фиксированной или неопределенной длины, обеспечивая ввод логических записей с пишущей машинки и вывод их на пишущую машинку.

• Обеспечивает возможность повторять ввод записи сначала, если была допущена ошибка при наборе информации на клавиатуре пишущей машинки.

Используемые макрокоманды. Декларативная: DTFCN — описание файла на пишущей машинке; императивные — см. 3.3.8.

Ниже описываются операнды для декларативной макрокоманды.

Макрокоманда DTFCN:

$\left[\text{BLKSIZE} = \left\{\frac{80}{n}\right\}\right]$ — длина области ввода-вывода (логической записи). Длина ее не должна превышать 256 байт;

DEVADDR = SYSxxx — используемое логическое устройство: SYSLOG или SYSnnn;

[RECFORM = формат], [RECSIZE = (r)] — см. 3.3.3;

IOAREA1 = имя, [MODNAME = имя], $\left[\text{TYPEFLE} = \left\{\frac{\text{INPUT}}{\text{OUTPUT}}\right\}\right]$,

[WORKA = YES] — см. 3.3.1.

3.3.5. ФАЙЛЫ, НЕЗАВИСИМЫЕ ОТ УСТРОЙСТВ

Назначение и возможности. Метод последовательного доступа для файлов, независимых от устройств:

• Обрабатывает файлы с неблокированными записями фиксированной длины, обеспечивая ввод записей с системных устройств и вывод их на системные устройства.

• Обрабатывает системные метки для файлов на магнитной ленте и дисках, а также файлы на магнитной ленте без меток (нестандартные метки и метки пользователя применять не разрешается).;

• Организует совмещение обработки с выполнением операций ввода-вывода, если используются две области ввода-вывода.

• Обрабатывает ошибки ввода-вывода, обеспечивая для файлов на магнитной ленте и дисках возможность игнорировать ошибку и продолжать обработку, или пропускать ошибочную запись, или обрабатывать ее подпрограммой пользователя.

• Определяет записи неверной длины для вводных файлов и обеспечивает возможность обрабатывать их подпрограммой пользователя.

• Определяет конец вводного файла и передает управление подпрограмме пользователя для завершения обработки файла.

• Обрабатывает укороченные блоки для файлов с блокированными записями фиксированной и переменной длины или с расширенными записями.

• При открытии файла выполняет перемотку магнитной ленты к точке загрузки, если режим перемотки не запрещен.

• При закрытии файла или при обнаружении конца тома выполняет только перемотку магнитной ленты к точке загрузки или перемотку с разгрузкой, если режим перемотки с разгрузкой задан.

Используемые макрокоманды. Декларативные: DTFDI — описание файла, независимого от устройства; DIMOD — описание модуля вво-

да-вывода для файлов, независимых от устройств; императивные — см. 3.3.8.

Ниже описываются операнды для каждой декларативной макрокоманды.

Макрокоманда DTFDI:

DEVADDR = SYSxxx — используемое логическое устройство: для вводных файлов SYSRDR или SYSIPT, для выводных — SYSPCH или SYSLST. Для SYSRDR и SYSIPT могут быть назначены устройство ввода с перфокарт, магнитная лента, диски; для SYSPCH — устройство вывода на перфокарты, магнитная лента, диски; для SYSLST — печать, магнитная лента, диски;

[EOFADDR = имя] — символический адрес в программе пользователя, по которому передается управление при обнаружении конца вводного файла: при считывании ленточной марки с магнитной ленты; записи EOF с дисков; перфокарты, содержащей символы /* и /& в двух первых колонках;

$$\left[\text{ERROPT} = \left\{\begin{matrix}\text{IGNORE}\\ \text{SKIP}\\ \text{имя}\end{matrix}\right\}\right]$$ — режим обработки ошибок ввода для магнитной ленты и дисков: IGNORE — ошибка игнорируется, запись предоставляется пользователю для обработки; SKIP — ошибочная запись пропускается, считывается следующая запись и обработка продолжается; имя — символический адрес подпрограммы пользователя, которой передается управление при обнаружении ошибки ввода;

[RECSIZE = n] — длина записей файла. Для вводных файлов (SYSRDR или SYSIPT) максимальный размер записи — 80 байт, для выводных: 121 байт для SYSLST и 81 — для SYSPCH. Такая же длина предполагается, если операнд отсутствует;

[IOAREA1 = имя, [IOAREA2 = имя], [IOREG = (r)], [MODNAME = имя], [RDONLY = YES], [SEPASMB = YES] — см. 3.3.1; [WLRERR = имя] — см. 3.3.3.

Макрокоманда DIMOD:

[IOAREA2 = YES], [RDONLY = YES], [SEPASMB = YES],

$$\left[\text{TYPEFLE} = \left\{\begin{matrix}\text{INPUT}\\ \underline{\text{OUTPUT}}\end{matrix}\right\}\right]$$ — см. 3.3.1.

3.3.6. ФАЙЛЫ НА МАГНИТНОЙ ЛЕНТЕ

Назначение и возможности. Метод последовательного доступа для файлов на магнитной ленте:

- Обрабатывает вводные, выводные и рабочие файлы, обеспечивая ввод логических записей с магнитной ленты в основную память и вывод их из основной памяти на магнитную ленту. Вводные и выводные файлы могут состоять из записей любого допустимого формата (см. 3.1.1), рабочие файлы — из неблокированных записей фиксированной или неопределенной длины.

- Для вводных и выводных файлов обрабатывает однотомные и многотомные файлы, многофайловые тома и многофайловую/многотомную совокупность (рабочие файлы могут быть только однотомными).
- Обрабатывает файлы с системными метками, или с системными метками и метками пользователя, или с нестандартными метками, или без меток. Системные метки обрабатываются программами СУВВ, а метки пользователя и нестандартные метки — с помощью подпрограммы пользователя.
- Для вводных и выводных файлов определяет конец тома и организует переход к обработке нового тома; может использоваться сменное устройство.
- Обрабатывает однотомные вводные и рабочие файлы, читая магнитную ленту в обратном направлении.
- Пропускает записи контрольных точек, если они содержатся во вводном файле.
- Совмещает обработку с выполнением операций ввода-вывода для вводных и выводных файлов, если используются две области ввода-вывода и (или) рабочая область.
- Выполняет операции управления магнитной лентой, обеспечивая подвод магнитной ленты к требуемому блоку.
- Обрабатывает ошибки данных, обнаруживаемые при чтении, а также любые другие ошибки ввода-вывода, обеспечивая возможность игнорировать ошибку, пропускать ошибочный блок или обрабатывать с помощью подпрограммы пользователя.
- Определяет записи неверной длины для вводных файлов и обеспечивает возможность обрабатывать их с помощью подпрограммы пользователя.

Используемые макрокоманды. Декларативные: DTFMT — описание файла на магнитной ленте, MTMOD — описание модуля ввода-вывода для файлов на магнитной ленте; императивные — см. 3.3.8.

Ниже описываются операнды для каждой декларативной макрокоманды.

Макрокоманда DTFMT:

$$\left[\text{BLKSIZE} = \left\{\begin{matrix}12\\18\\n\end{matrix}\right\}\right]$$ — длина области ввода-вывода (блока).

Для записей переменной длины и расширенных записей она должна включать длину описателя блока (4 байта). Минимальный размер физической записи 12 байт, а для выводных файлов с расширенными записями — 18 байт;

[CKPTREC = YES] — обход записей контрольных точек при обработке вводных файлов;

DEVADDR = SYSxxx — используемое логическое устройство: SYSRDR, SYSIPT, SYSPCH, SYSLST или SYSnnn;

[EOFADDR = имя] — символический адрес в программе пользователя, по которому передается управление при обнаружении конца вводного или рабочего файла; для файлов со стандартными метками —

при считывании ленточной марки и метки EOF1, для файлов с нестандартными метками или без меток — при считывании ленточной марки, для системных файлов на устройствах SYSRDR и SYSIPT — при считывании записи /* или /&;

[ERREXT = YES] — использование макрокоманды ERET (см. 3.3.8) для возврата в модуль ввода-вывода из подпрограммы пользователя, которая обрабатывает любые ошибки ввода-вывода. Этот операнд может быть указан только при наличии операнда ERROPT;

$$\left[\text{ERROPT} = \begin{Bmatrix}\text{IGNORE}\\ \text{SKIP}\\ \text{имя}\end{Bmatrix}\right]$$ — при наличии операнда ERREXT — режим обработки ошибок данных, возникающих при чтении, и других неустранимых ошибок ввода-вывода, в противном случае — режим обработки только ошибок данных, возникающих при чтении: IGNORE — ошибка игнорируется и продолжается обычная обработка. Для вводных файлов записи блока предоставляются пользователю для обработки; SKIP — ошибочный блок пропускается, считывается следующий блок и обработка продолжается с первой записи этого блока; имя — символический адрес подпрограммы пользователя, которой передается управление при обнаружении ошибки ввода-вывода;

[FILABL = тип] — тип меток для вводных и выводных файлов: STD — файл со стандартными метками, NO — файл без меток, NSTD — файл с нестандартными метками. Если операнд отсутствует, предполагается FILABL = NO;

[HDRINFO = YES] — печать начальной метки файла на SYSLOG при открытии файла со стандартными метками, а также печать имени файла, логического устройства и адреса физического устройства при обнаружении конца тома;

[IOREG = (r)] — номер регистра, в который загружается адрес записи вводного файла или адрес области для построения очередной записи выводного файла. Этот операнд должен задаваться в следующих случаях: определены две области ввода-вывода, а рабочая область не используется; блокированные записи обрабатываются непосредственно в области ввода-вывода; записи неопределенной или переменной длины считываются в обратном направлении;

[LABADDR = имя] — символический адрес подпрограммы пользователя, которой передается управление для обработки нестандартных меток или меток пользователя;

$$\left[\text{NOTEPNT} = \begin{Bmatrix}\text{POINTS}\\ \text{YES}\end{Bmatrix}\right]$$ — использование для обработки рабочего файла макрокоманд POINTS, NOTE, POINTR и POINTW: POINTS, если применяется только макрокоманда POINTS; YES, если применяются все перечисленные макрокоманды;

$$\left[\text{READ} = \begin{Bmatrix}\text{FORWARD}\\ \text{BACK}\end{Bmatrix}\right]$$ — направление, в котором считывается магнитная лента: FORWARD — прямое направление, BACK — обратное направление;

[RECFORM = формат] — формат записей описываемого файла: FIXUNB — неблокированные фиксированной длины, FIXBLK — блокированные фиксированной длины, VARUNB — неблокированные переменной длины, VARBLK — блокированные переменной длины, SPNUNB — неблокированные расширенные, SPNBLK — блокированные расширенные, UNDEF — неопределенной длины. Если операнд отсутствует, предполагается RECFORM = FIXUNB. Для рабочих файлов может быть указано только FIXUNB или UNDEF;

$\left[\text{RECSIZE} = \left\{\begin{matrix}(r)\\ n\end{matrix}\right\}\right]$ — номер регистра, в который пользователь должен загружать длину выводной записи или в который помещается длина считанной записи при обработке расширенных записей или записей неопределенной длины. Если обрабатываются блокированные записи фиксированной длины, этот операнд определяет длину логической записи;

$\left[\text{REWIND} = \left\{\begin{matrix}\text{UNLOAD}\\ \text{NORWD}\end{matrix}\right\}\right]$ — режим перемотки магнитной ленты: UNLOAD — перемотка ленты к точке загрузки при выполнении макрокоманды OPEN, перемотка и разгрузка при выполнении макрокоманды CLOSE или при достижении конца тома; NORWD — запрещается перемотка ленты;

[TPMARK = NO] — игнорирование записи ленточной марки перед данными первого на ленте файла без меток или между группой начальных нестандартных меток и данными для файлов с нестандартными метками;

$\left[\text{TYPEFLE} = \left\{\begin{matrix}\text{INPUT}\\ \text{OUTPUT}\\ \text{WORK}\end{matrix}\right\}\right]$ — тип описываемого файла: INPUT — вводной, OUTPUT — выводной, WORK — рабочий;

[VARBLD = (r)] — номер регистра, в котором содержится длина свободного места в области вывода при построении блокированных записей переменной длины непосредственно в области вывода;

[WORKA = YES] — использование рабочей области для обработки записей файла. Для расширенных записей обязателен;

IOAREA1 = имя, [IOAREA2 = имя], [MODNAME = имя], [RDONLY = YES], [SEPASMB = YES] — см. 3.3.1, [WLRERR = = имя] — см. 3.3.3.

Макрокоманда MTMOD:

[CKPTREC = YES] — пропуск записей контрольных точек для вводных файлов. Для рабочих файлов этот операнд не требуется;

[ERREXT = YES] — выполнение макрокоманды ERET (см. 3.3.8) и обработка не только ошибок данных, возникающих при чтении, но и любых других ошибок ввода-вывода. Для рабочих файлов этот операнд может быть указан только при наличии операнда ERROPT;

[ERROPT = YES] — обработка не только ошибок данных, возникающих при чтении, но и других неустранимых ошибок ввода-вывода. Требуется только для рабочих файлов;

$\left[\text{NOTEPNT} = \left\{\begin{matrix}\text{YES}\\ \text{POINTS}\end{matrix}\right\}\right]$ — при наличии POINTS — выполнение макрокоманды POINTS, а при наличии YES — выполнение макрокоманд POINTS, POINTW, POINTR или NOTE. Требуется только для рабочих файлов;

$\left[\text{READ} = \left\{\begin{matrix}\underline{\text{FORWARD}}\\ \text{BACK}\end{matrix}\right\}\right]$ — считывание ленты в прямом (FORWARD) или прямом и обратном (BACK) направлениях. Для рабочих файлов этот операнд не требуется;

[RECFORM = формат] — формат записей обрабатываемых файлов: FIXUNB, FIXBLK — фиксированной длины, VARUNB, VARBLK — переменной длины, SPNUNB, SPNBLK — расширенные, UNDEF — неопределенной длины. Одним модулем обрабатываются как блокированные, так и неблокированные записи. Если операнд отсутствует, предполагается RECFORM=FIXUNB, RECFORM= = FIXBLK. Для рабочих файлов он не требуется;

$\left[\text{TYPEFLE} = \left\{\begin{matrix}\underline{\text{INPUT}}\\ \underline{\text{OUTPUT}}\\ \text{WORK}\end{matrix}\right\}\right]$ — тип обрабатываемых файлов: INPUT — вводной или выводной, OUTPUT — вводной или выводной, WORK — рабочий;

[RDONLY = YES], [SEPASMB = YES], [WORKA = YES] — см. 3.3.1.

3.3.7. ФАЙЛЫ НА ДИСКАХ

Назначение и возможности. Метод последовательного доступа для файлов на дисках:

- Обрабатывает вводные, выводные и рабочие файлы, обеспечивая ввод логических записей в основную память с диска и вывод их из основной памяти на диск. Вводные и выводные файлы могут состоять из записей любого допустимого формата (см. 3.1.1), рабочие файлы — из неблокированных записей фиксированной или неопределенной длины.
- Для вводных и выводных файлов обрабатывает однотомные и многотомные файлы, многофайловые тома, многофайловую/многотомную совокупность (рабочие файлы могут быть только однотомными.)
- Обрабатывает файлы с системными метками и для вводных и выводных файлов обеспечивает возможность обрабатывать метки пользователя с помощью подпрограммы пользователя.
- Для вводных и выводных файлов обрабатывает конец тома и организует переход к обработке нового тома.
- Обрабатывает файлы, состоящие не более чем из 255 участков; участки могут быть с разделенными и неразделенными цилиндрами.
- Совмещает обработку с выполнением операций ввода-вывода, если используются две области ввода-вывода и/или рабочая область.

- Контролирует доступ к защищенным файлам.
- Выполняет функцию захвата дорожки при обработке вводных и рабочих файлов с обновлением.
- Обрабатывает укороченные блоки для файлов с блокированными записями фиксированной и переменной длины или с расширенными записями.
- Выполняет операции управления, заданные макрокомандой CNTRL, а также макрокоманды, обеспечивающие для рабочего файла доступ к требуемой записи.
- Обрабатывает ошибки данных, возникающие при чтении и записи, а также любые другие ошибки ввода-вывода, обеспечивая возможность игнорировать ошибку, или пропускать ошибочный блок, или обрабатывать его подпрограммой пользователя.
- Определяет записи неверной длины и обеспечивает возможность обрабатывать их с помощью подпрограммы пользователя.
- Определяет конец вводного или рабочего файла и передает управление подпрограмме пользователя для завершения обработки файла.

Используемые макрокоманды. Декларативные: DTFSD — описание последовательного файла на дисках, SDMODxx — описание модулей ввода-вывода для последовательных файлов на дисках: SDMODFI — для вводных файлов с записями фиксированной длины, SDMODFO — для выводных файлов с записями фиксированной длины, SDMODFU — для вводных файлов с обновлением записей фиксированной длины, SDMODVI — для вводных файлов с записями переменной длины и расширенными записями, SDMODVO — для выводных файлов с записями переменной длины и расширенными записями, SDMODVU — для вводных файлов с обновлением записей переменной длины и расширенных записей, SDMODUI — для вводных файлов с записями неопределенной длины, SDMODUO — для выводных файлов с записями неопределенной длины, SDMODUU — для вводных файлов с обновлением записей неопределенной длины, SDMODW — для рабочих файлов. Императивные макрокоманды — см. 3.3.8

Ниже описываются операнды для каждой декларативной макрокоманды.

Макрокоманда DTFSD:

$\left[\text{BLKSIZE} = \left\{\frac{10}{n}\right\}\right]$ — длина области ввода-вывода (блока). Для записей переменной длины и расширенных записей она должна включать длину описателя блока (4 байта). Для выводных файлов она должна включать 8 байт для построения счетчика. Длина блока не должна превышать емкость дорожки используемого устройства;

[CONTROL = YES] — использование макрокоманды CNTRL;

[DELETFL = NO] — сохранение меток формата 1 и 3 при выполнении макрокоманды CLOSE для рабочего файла;

[DEVADDR = SYSxxx] — используемое устройство: SYSRDR, SYSIPT, SYSPCH, SYSLST, SYSRES, SYSLNK, SYSREC, SYSRLB, SYSRLB, SYSSLB или SYSnnn;

[DEVICE = шифр] — шифр используемого устройства ввода-вывода (см. табл. 3.8). Если операнд отсутствует, предполагается устройство ЕС-5052.

[EOFADDR = имя] — символический адрес в программе пользователя, по которому передается управление при обнаружении конца вводного или рабочего файла: при считывании записи EOF либо записи /* или /&, если для системных файлов SYSRDR, SYSIPT назначены диски:

[ERREXT = YES] — обработка модулем не только ошибок данных, возникающих при чтении и записи, но и других неустранимых ошибок ввода-вывода. При этом возврат в модуль из подпрограммы пользователя (обработка ошибок ввода-вывода и/или записей неверной длины) может осуществляться с помощью макрокоманды ERET (см. 3.3.8). Этот операнд указывается только при наличии операнда ERROPT;

$$\left[\text{ERROPT} = \begin{Bmatrix}\text{IGNORE}\\ \text{SKIP}\\ \text{имя}\end{Bmatrix}\right]$$ — режим обработки ошибок данных, возникающих при чтении и записи, и других неустранимых ошибок ввода-вывода, если задан операнд ERREXT, в противном случае — режим обработки только ошибок данных, возникающих при чтении и записи: IGNORE — ошибка игнорируется, если файлы вводные, то записи блока поступают для обработки, если файлы выводные — считается, что блок записан верно; SKIP — ошибочный блок пропускается, считывается следующий блок и обработка продолжается с первой записи этого блока, на выводе выполняется режим IGNORE; имя — символический адрес подпрограммы пользователя, которой передается управление при обнаружении ошибки ввода-вывода;

[FEOVD = YES] — использование макрокоманды FEOVD, по которой запрашивается конец тома до действительного его достижения;

[HOLD = YES] — использование функций захвата дорожки для обработки вводных и рабочих файлов с обновлением;

IOAREA1 = имя — символический адрес первой области ввода-вывода. Длина ее должна быть не меньше длины, указанной в BLKSIZE. Область ввода-вывода должна начинаться с границы полуслова;

[IOREG = (r)] — номер регистра, в который загружается адрес записи вводного файла или адрес области для построения очередной записи выводного файла, если применяются две области ввода-вывода и не используется рабочая область или если блокированные записи обрабатываются в области ввода-вывода;

[LABADDR = имя] — символический адрес подпрограммы пользователя, которой передается управление для обработки меток пользователя;

$$\left[\text{NOTEPNT} = \begin{Bmatrix}\text{POINTRW}\\ \text{YES}\end{Bmatrix}\right]$$ — использование для обработки рабочего файла макрокоманд NOTE, POINTW, POINTR, POINTS: POINTRW, если применяются макрокоманды NOTE, POINTW или

POINTR; YES — если используются макрокоманды NOTE, POINTW, POINTR или POINTS;

[TRUNCS = YES] — использование макрокоманды TRUNC для записи укороченных блоков или чтения укороченных блоков, если обрабатываются файлы с блокированными записями фиксированной длины;

[UPDATE = YES] — обновление записей вводных или рабочих файлов;

[VERIFY = YES] — проверка правильности записи информации на диски;

[IOAREA2 = имя], [MODNAME = имя], [RDONLY = YES], [SEPASMB = YES] — см. 3.3.1;

[RECFORM = формат], $\left[\text{RECSIZE} = \begin{Bmatrix}(\text{r})\\ \text{n}\end{Bmatrix}\right]$, $\left[\text{TYPEFLE} = \begin{Bmatrix}\text{INPUT}\\ \text{OUTPUT}\\ \text{WORK}\end{Bmatrix}\right]$, [VARBLD = (r)], [WORKA = YES] — см. 3.3.6; [WLRERR = имя] — см. 3.3.3.

Макрокоманды SDMODxx:

[CONTROL = YES] — выполнение функций макрокоманды CNTRL;

[ERREXT = YES] — использование макрокоманды ERET (см. 3.3.8) для возврата в модуль ввода-вывода из подпрограммы пользователя, которая осуществляет обработку любых ошибок ввода-вывода. Операнд ERREXT может указываться только при наличии операнда ERROPT;

[ERROPT = YES] — обработка не только ошибок данных, возникающих при чтении и записи, но и других неустранимых ошибок ввода-вывода;

[HOLD = YES] — выполнение функций захвата дорожки при обработке вводных и рабочих файлов с обновлением. Используется для макрокоманд SDMODFU, SDMODVU, SDMODUU и SDMODW;

[FEOVD = YES] — выполнение функций макрокоманды FEOVD. Не используется только для макрокоманды SDMODW;

$\left[\text{NOTEPNT} = \begin{Bmatrix}\text{POINTRW}\\ \text{YES}\end{Bmatrix}\right]$ — при наличии POINTRW — выполнение макрокоманд NOTE, POINTW, POINTR, а при наличии YES — макрокоманд NOTE, POINTS, POINTR и POINTW. Используется только для рабочих файлов, т. е. для макрокоманды SDMODW;

[RECFORM = формат] — формат записей обрабатываемых файлов: SPNUNB, SPNBLK — переменной длины и расширенные. Одним модулем обрабатываются как блокированные, так и неблокированные записи. Используется для макрокоманд SDMODVI, SDMODVO и SDMODVU;

[TRUNCS = YES] — обработка укороченных блоков для файлов с записями фиксированной длины, т.е. вводные файлы могут содержать укороченные блоки, а для выводных файлов может выдаваться макро-

команда TRUNC для записи укороченных блоков. Используется для макрокоманд SDMODFI, SDMODFO и SDMODFU;

[UPDATE = YES] — обновление записей при обработке рабочих файлов.

Используется для макрокоманды SDMODW;

[RDONLY = YES], [SEPASMB = YES] — см. 3.3.1.

3.3.8. ИМПЕРАТИВНЫЕ МАКРОКОМАНДЫ

Список императивных макрокоманд методов последовательного доступа представлен в табл. 3.9, а их описание и формат даются ниже.

Таблица 3.9

Список императивных макрокоманд

Макрокоманда	Файлы							Назначение макрокоманды
	на перфокартах	на печати	на перфоленте	на пишущей машинке	независимые от устройств	на магнитной ленте	на дисках	
OPEN(R)	×	×	×	×	×	×	×	Открытие файла
LBRET						×	×	Возврат управления из подпрограмм пользователя (обработка меток пользователя или нестандартных меток)
GET	×	×	×	×	×	×	×	Получение логической записи
PUT	×	×	×	×	×	×	×	Запись логической записи
RELSE						×	×	Освобождение области ввода
TRUNC						×	×	Усечение выводного блока
CNTRL	×	×				×	×	Управление устройством
ERET						×	×	Возврат управления из подпрограммы пользователя (обработка ошибок ввода-вывода и/или записей неверной длины)
PRTOV	×							Управление переходом на следующий лист
READ						×	×	Чтение физической записи (блока)
WRITE						×	×	Запись физической записи (блока)
CHECK						×	×	Ожидание завершения операции ввода-вывода
NOTE						×	×	Получение идентификации записи
POINTR						×	×	Установка файла для чтения
POINTW						×	×	Установка файла для записи
POINTS						×	×	Установка файла в начало
FREE							×	Освобождение захваченной дорожки
FEOV						×		Запрос конца тома для магнитной ленты
FEOVD							×	Запрос конца тома для дисков
CLOSE(R)	×	×	×	×	×	×	×	Закрытие файла

Примечание. Знак «×» обозначает использование макрокоманды для обработки соответствующего файла.

Макрокоманда OPEN(R) открывает файлы, подготавливая их к обработке. Она обязательна для файлов на дисках и магнитных лентах, а также для файлов на печати, для выводных файлов на перфокартах и перфоленте, если используются две области вывода, и для файлов, независимых от устройств. Для несамоперемещающихся программ используется OPEN, а для самоперемещающихся — OPENR. Последняя обязательна для всех файлов. Одной макрокомандой можно открыть до шестнадцати файлов. Формат макрокоманды:

$$[\text{имя}]\ \begin{Bmatrix}\text{OPEN}\\ \text{OPENR}\end{Bmatrix}\ \begin{Bmatrix}\text{имя файла 1}\\ (\text{г1})\end{Bmatrix}\left[,\begin{Bmatrix}\text{имя файла 2}\\ (\text{г2})\end{Bmatrix},\ldots,\begin{Bmatrix}\text{имя файла n}\\ (\text{гn})\end{Bmatrix}\right]$$

где имя файла — здесь и далее — символический адрес имени файла.

Макрокоманда LBRET возвращает управление СУВВ из подпрограмм обработки меток пользователя или подпрограмм обработки нестандартных меток. Формат макрокоманды:

$$[\text{имя}]\ \text{LBRET}\ \begin{Bmatrix}1\\ 2\\ 3\end{Bmatrix}$$

где значение «1» указывается, если необходимо прекратить обработку остальных меток пользователя или если созданная метка пользователя для выводного файла на магнитной ленте является последней;

значение «2» указывается, если после записи или считывания очередной метки пользователя требуется вернуть управление подпрограмме для проверки или создания очередной метки пользователя;

значение «3» допускается только для файлов на дисках: оно указывается, если должно выполняться обновление меток пользователя для вводного файла, в остальном — аналогично значению «2».

Макрокоманда GET предоставляет пользователю для обработки в области ввода или в рабочей области очередную логическую запись. Формат макрокоманды:

$$[\text{имя}]\ \text{GET}\ \begin{Bmatrix}\text{имя файла}\\ (1)\end{Bmatrix}\left[,\begin{Bmatrix}\text{имя рабочей области}\\ (0)\end{Bmatrix}\right]$$

Макрокоманда PUT помещает в файл очередную логическую запись, подготовленную пользователем в области вывода или в рабочей области, а также обновляет запись вводного файла. Формат макрокоманды:

$$[\text{имя}]\ \text{PUT}\ \begin{Bmatrix}\text{имя файла}\\ (1)\end{Bmatrix}\left[,\begin{Bmatrix}\text{имя рабочей области}\\ (0)\end{Bmatrix}\right]$$

Макрокоманда RELSE пропускает оставшиеся записи текущего блока, по следующей за ней макрокоманде GET пользователю предоставляется первая логическая запись следующего блока. RELSE используется только для вводных файлов на магнитной ленте и дисках с блокированными записями (фиксированной, переменной длины или расширенными записями). Формат макрокоманды:

$$[\text{имя}]\ \text{RELSE}\ \begin{Bmatrix}\text{имя файла}\\ (1)\end{Bmatrix}$$

Макрокоманда TRUNC записывает построенный к этому моменту в области вывода блок на магнитную ленту или на диски и освобождает

область вывода для построения нового блока. TRUNC используется только для выводных файлов на магнитной ленте и дисках с блокированными записями (фиксированной или переменной длины). Формат макрокоманды:

$$[\text{имя}]\quad \text{TRUNC}\ \left\{\begin{matrix}\text{имя файла}\\(1)\end{matrix}\right\}$$

Макрокоманда CNTRL выполняет операции управления на устройстве вывода на перфокарты, печати, магнитной ленте и дисках. Формат макрокоманды:

$$[\text{имя}]\ \text{CNTRL}\ \left\{\begin{matrix}\text{имя файла}\\(1)\end{matrix}\right\},\ \text{код}\ [,n]\ [,m]$$

где код — код выполняемой операции управления (см. табл. 3.10);

n — номер кармана (1 или 2) для укладки перфорируемых карт, или количество строк (1, 2 или 3), пропускаемых перед печатью логической записи, или код управляющей перфоленты (1—12), до которого должен производиться прогон бумаги перед печатью логической записи;

m — количество пропускаемых строк (1, 2 или 3) или код управляющей перфоленты (1—12), до которого должен производиться прогон бумаги после печати логической записи.

Для файлов на магнитной ленте и дисках используются только первых два операнда.

Таблица 3.10

Код операции управления для макрокоманды CNTRL

Устройство	Код операции	Операция
Накопитель на магнитной ленте	REW	Перемотать
	RUN	Перемотать и разгрузить
	ERG	Стереть промежуток
	WTM	Записать марку
	FSR	Шаг на блок вперед
	FSF	Шаг на файл вперед
	FSL	Пропустить расширенную логическую запись в прямом направлении
	BSR	Шаг на блок назад
	BSF	Шаг на файл назад
	BSL	Пропустить расширенную логическую запись в обратном направлении
Перфокарточное устройство вывода	SS	Выбрать карман 1 или 2
Печатающее устройство	SP	Пропустить одну, две или три строки
	SK	Прогнать до кода
Накопитель на магнитных дисках	SEEK	Установить к указанной дорожке

Макрокоманда ERET осуществляет возврат управления в модуль из подпрограмм пользователя после обработки ошибок ввода-вывода

и/или записей неверной длины. Получив управление, модуль выполняет действия, определенные операндами. Формат макрокоманды:

$$[\text{имя}]\quad \text{ERET}\quad \left\{\begin{matrix}\text{SKIP}\\ \text{IGNORE}\\ \text{RETRY}\end{matrix}\right\}$$

где SKIP — пропускается ошибочный блок;
IGNORE — игнорируется ошибка, записи ошибочного блока предоставляются для обработки;
RETRY — повторяется выполнение операции ввода-вывода.

Макрокоманда PRTOV вызывает операцию, которая должна быть выполнена в случае обнаружения конца печатного листа. Формат макрокоманды:

$$[\text{имя}]\quad \text{PRTOV}\quad \left\{\begin{matrix}\text{имя файла}\\ (1)\end{matrix}\right\},\ \text{код}\ \left[,\left\{\begin{matrix}\text{имя подпрограммы}\\ (0)\end{matrix}\right\}\right]$$

где код — код управляющей перфоленты (9 или 12), при достижении которого обнаруживается конец печатного листа;
имя подпрограммы — символический адрес подпрограммы пользователя, которой передается управление при обнаружении конца печатного листа. Если этот операнд опущен, то при достижении конца печатного листа осуществляется прогон бумаги до кода 1 управляющей перфоленты.

Макрокоманда READ считывает очередной блок или его часть из рабочего файла в область ввода. Формат макрокоманды:

$$[\text{имя}]\quad \text{READ}\quad \left\{\begin{matrix}\text{имя файла}\\ (1)\end{matrix}\right\},\text{SQ},\left\{\begin{matrix}\text{область}\\ (0)\end{matrix}\right\}\left[,\left\{\begin{matrix}\text{длина}\\ (\text{r})\\ \text{S}\end{matrix}\right\}\right]$$

где SQ — последовательный тип считывания;
область — символический адрес области ввода;
длина — длина считываемого блока или его части (для записей неопределенной длины). Если считывается весь блок, указывается S.

Макрокоманда WRITE записывает блок в рабочий файл из области вывода. Формат макрокоманды:

$$[\text{имя}]\quad \text{WRITE}\quad \left\{\begin{matrix}\text{имя файла}\\ (1)\end{matrix}\right\},\left\{\begin{matrix}\text{SQ}\\ \text{UPDATE}\end{matrix}\right\},\left\{\begin{matrix}\text{область}\\ (0)\end{matrix}\right\}\left[,\left\{\begin{matrix}\text{длина}\\ (\text{r})\end{matrix}\right\}\right]$$

где SQ — последовательный тип записи для магнитной ленты и дисков;
UPDATE — обновление данных ранее записанного блока;
область — символический адрес области вывода;
длина — длина записываемого блока для записей неопределенной длины.

Макрокоманда CHECK ожидает завершения передачи данных, начатой предыдущей макрокомандой READ или WRITE для рабочих файлов на магнитной ленте и дисках. Формат макрокоманды:

$$[\text{имя}]\quad \text{CHECK}\ \left\{\begin{matrix}\text{имя файла}\\ (1)\end{matrix}\right\}$$

Макрокоманда NOTE предоставляет в регистре 1 идентификацию только что считанного или записанного блока рабочего файла. Для

файлов на магнитной ленте — 0bbb, где bbb — количество блоков от точки загрузки до последнего записанного или считанного блока, а для дисков — 0CHR, где C — номер цилиндра, H — номер дорожки, R — номер записи на дорожке. Формат макрокоманды:

$$[\text{имя}]\ \text{NOTE}\ \left\{\begin{matrix}\text{имя файла}\\(1)\end{matrix}\right\}$$

Макрокоманда POINTR устанавливает рабочий файл для чтения блока, указанного вторым операндом. Формат макрокоманды:

$$[\text{имя}]\ \text{POINTR}\ \left\{\begin{matrix}\text{имя файла}\\(1)\end{matrix}\right\},\left\{\begin{matrix}\text{адрес}\\(0)\end{matrix}\right\}$$

где адрес — символический адрес 4-байтового поля основной памяти, содержащего идентификацию (0bbb или 0CHR) требуемого блока.

Макрокоманда POINTW устанавливает рабочий файл для помещения в него нового блока вслед за блоком, указанным вторым операндом. Формат макрокоманды:

$$[\text{имя}]\ \text{POINTW}\ \left\{\begin{matrix}\text{имя файла}\\(1)\end{matrix}\right\},\left\{\begin{matrix}\text{адрес}\\(0)\end{matrix}\right\}$$

где адрес — символический адрес 4-байтового поля основной памяти, содержащего идентификацию (0bbb или 0CHR) требуемого блока.

Макрокоманда POINTS устанавливает рабочий файл в начало. Для файлов на магнитной ленте осуществляется установка ленты перед первым блоком данных, для файлов на дисках — установка файла к нижней границе первого участка. Формат макрокоманды:

$$[\text{имя}]\ \text{POINTS}\ \left\{\begin{matrix}\text{имя файла}\\(1)\end{matrix}\right\}$$

Макрокоманда FREE вызывает освобождение захваченной по макрокоманде READ дорожки рабочего файла с обновлением, если считанная запись не обновляется. Если считанная запись обновляется, то захваченная дорожка для вводных и рабочих файлов с обновлением освобождается автоматически. Формат макрокоманды:

$$[\text{имя}]\ \text{FREE}\ \left\{\begin{matrix}\text{имя файла}\\(1)\end{matrix}\right\}$$

Макрокоманды FEOV/FEOVD запрашивают конец тома для файлов на магнитной ленте (FEOV) или для файлов на дисках (FEOVD). Выдача макрокоманд означает, что обработка записей одного тома считается завершенной, а остальные записи этого же файла должны считываться или записываться на следующий том файла. Макрокоманды могут использоваться только для логических устройств программиста. В случае системных устройств для этих целей используется макрокоманда SEOV (см. 3.2.1). Формат макрокоманды:

$$[\text{имя}]\ \left\{\begin{matrix}\text{FEOV}\\\text{FEOVD}\end{matrix}\right\}\ \left\{\begin{matrix}\text{имя файла}\\(1)\end{matrix}\right\}$$

Макрокоманда CLOSE(R) закрывает файлы, запрещая их дальнейшую обработку. Она обязательна для файлов на дисках и магнитных лентах, а также для выводных файлов на перфокартах, для выводных файлов на перфоленте, если используются две области вывода, и для

файлов, независимых от устройств. Для несамоперемещающихся программ используется CLOSE, а для самоперемещающихся — CLOSER, которая обязательна для всех файлов. Одной макрокомандой можно закрыть до шестнадцати файлов. Формат макрокоманды:

$$[\text{имя}]\ \begin{Bmatrix}\text{CLOSE}\\ \text{CLOSER}\end{Bmatrix}\ \begin{Bmatrix}\text{имя файла 1}\\ (\text{r1})\end{Bmatrix}\left[,\begin{Bmatrix}\text{имя файла 2}\\ (\text{r2})\end{Bmatrix},\ldots,\begin{Bmatrix}\text{имя файла n}\\ (\text{rn})\end{Bmatrix}\right]$$

3.4. МЕТОД ПРЯМОГО ДОСТУПА

Назначение и возможности. Метод прямого доступа:

- Обеспечивает произвольную обработку записей файлов на дисках. Файлы могут состоять из неблокированных записей фиксированной или переменной длины, расширенных записей, а также записей неопределенной длины. Записи могут быть с ключами или без ключей.
- Обрабатывает однотомные и многотомные файлы и многофайловые тома.
- Обрабатывает системные метки и обеспечивает обработку меток пользователя с помощью подпрограммы пользователя.
- Осуществляет поиск записи по идентификатору на дорожке и по ключу на дорожке или цилиндре. Для указания адреса дорожки может использоваться относительная (десятичная зонированная или шестнадцатеричная) или абсолютная адресация. Идентификатор обрабатываемой или следующей записи предоставляется пользователю.
- Ведет учет свободного места, т. е. незанятого записями для каждой дорожки, помещая в запись 0 каждой дорожки информацию о свободной части дорожки, и использует ее при добавлении новых записей в файл.
- Выполняет операции управления, заданные макрокомандой CNTRL.
- Обрабатывает ошибки ввода-вывода.

Используемые макрокоманды. Декларативные: DTFDA — описание файла метода прямого доступа, DAMOD — описание модуля ввода-вывода метода прямого доступа для обработки файлов с записями фиксированной или неопределенной длины, DAMODV — описание модуля ввода-вывода метода прямого доступа для обработки файлов с записями переменной длины или с расширенными записями; императивные: OPEN(R) — открытие файла, LBRET — возврат управления из подпрограммы обработки меток пользователя (см. 3.3.8) или подпрограммы обработки информации об участках, READ — чтение физической записи, WRITE — запись физической записи, WAITF — ожидание завершения передачи данных, CNTRL — установка к указанной дорожке, FREE — освобождение захваченной дорожки, CLOSE(R) — закрытие файла.

Ниже описываются операнды для каждой декларативной макрокоманды и императивные макрокоманды.

Макрокоманда DTFDA:

AFTER=YES — использование макрокоманды WRITE AFTER или WRITE RZERO;

BLKSIZE = n — длина области ввода-вывода. Она должна включать длину данных и длину ключа, если записи содержат ключи и для

обработки файла используются макрокоманды, передающие ключ и данные. Если используется макрокоманда WRITE AFTER или WRITE RZERO, она должна включать еще 8 байт для построения счетчика. В случае расширенных записей она должна включать 8 байт для описателей блока и сегмента, но не должна учитывать длину ключа и счетчика;

[DEVADDR = SYSnnn] — используемое логическое устройство;

[DSKXTNT = n] — максимальное количество участков (максимум 255), которые определены для файлов. Присутствие этого операнда для неблокированных записей фиксированной, переменной или неопределенной длины означает, что используется относительная адресация, в противном случае — абсолютная адресация. Для расширенных записей этот операнд требуется всегда. Для указания относительной адресации требуется также операнд RELTYP;

ERRBYTE = имя — символический адрес двухбайтового поля-указателя, в которое после выполнения макрокоманды READ или WRITE помещается информация о результате ее выполнения. Причины, вызывающие установку в единицу бит поля-указателя, приведены в табл. 3.11;

Таблица 3.11

Поле-указатель макрокоманды DTFDA

Бай.	Бит	Причина установки бита в 1	Пояснение
0	0	—	Бит не используется
0	1	Запись неверной длины	Обнаружена запись неверной длины при выполнении операции ввода-вывода
0	2	Ошибка, не связанная с передачей данных	Ошибочный блок не может быть ни прочитан, ни записан
0	3	—	Бит не используется
0	4	Недостаточно места	СУВВ определяет, что оставшегося на дорожке места недостаточно для размещения записи
0	5	—	Бит не используется
0	6	—	То же
0	7	Обращение к записи за границами участка	При относительной адресации адрес записи находится вне границ участков файла
1	0	Ошибка данных в области счетчика	Неустранимая ошибка
1	1	—	Бит не используется
1	2	Конец цилиндра	При многодорожечном поиске не обнаружена запись на цилиндре
1	3	Ошибка данных при чтении ключа или данных	Неустранимая ошибка
1	4	Запись не найдена	Не найдена запись при поиске ее на дорожке
1	5	Конец файла	Считана запись конца файла
1	6	Конец тома	Идентификатор следующей записи превышает границу тома
1	7	—	Бит не используется

[ERREXT = YES] — получение индикации в поле-указателе не только об ошибках данных, возникающих при чтении и записи, но и о других неустранимых ошибках ввода-вывода;

[FEOVD = YES] — обработка конца тома для вводных файлов. Операнд указывается только для вводных файлов, которые были созданы методом последовательного доступа с использованием макрокоманды FEOVD;

[HOLD = YES] — использование функций захвата дорожки при обработке файлов;

[IDLOC = имя] — символический адрес поля, в которое помещается идентификатор (ID) записи: идентификатор обработанной записи, если используются макрокоманды READ KEY, WRITE KEY и производится поиск на цилиндре; идентификатор следующей записи, если используются макрокоманды READ ID, WRITE ID, READ KEY или WRITE KEY и производится поиск на дорожке. Идентификатор записи поставляется в том же формате, что и в поле SEEKADR;

IOAREA1 = имя — символический адрес области ввода-вывода. Длина области должна быть не меньше длины, указанной в BLKSIZE;

[KEYARG = имя] — символический адрес поля основной памяти, в которое пользователь помещает ключ записи для макрокоманды READ или WRITE, если обращение к записи осуществляется с помощью ключа;

[KEYLEN = n] — длина ключа записей, если записи содержат ключи. Длина ключа записей предполагается равной нулю, если операнд опущен;

[READID = YES] — использование макрокоманды READ ID;

[READKEY = YES] — использование макрокоманды READ KEY;

[RECFORM = формат] — формат записей описываемого файла: FIXUNB — неблокированные фиксированной длины, VARUNB — неблокированные переменной длины, SPNUNB — неблокированные расширенные, UNDEF — неопределенной длины. Если операнд опущен, предполагается RECFORM = FIXUNB;

[RECSIZE = (r)] — номер регистра, в который пользователь должен загружать длину выводной записи или в который помещается длина считанной записи при обработке записей неопределенной длины;

$\left[\text{RELTYPE} = \begin{Bmatrix}\text{DEC}\\ \text{HEX}\end{Bmatrix}\right]$ — формат идентификатора записи при относительной адресации: DEC — десятичный зонированный, а HEX — шестнадцатеричный;

SEEKADR = имя — символический адрес поля-указателя дорожки, в которое пользователь должен поместить идентификатор записи, если производится поиск по идентификатору, или адрес дорожки, начиная с которой необходимо производить поиск по ключу. Формат поля-указателя дорожки приведен в табл. 3.12;

[SRCHM = YES] — поиск записи по ключу на цилиндре;

[TRLBL = YES] — использование конечных меток пользователя;

$\text{TYPEFLE} = \begin{Bmatrix}\text{INPUT}\\ \text{OUTPUT}\end{Bmatrix}$ — тип обработки меток файла:

Таблица 3.12

Поле-указатель дорожки, определяемое операндом SEEKADR

Байты	Обозначение байта	Возможные значения байта	Пояснение
		Относительная адресация (адрес дорожки в десятичном зонированном формате)	
0—7	TTTTTTTT	0—16777215	Номер дорожки относительно первой дорожки файла
8—9	RR	0—99	Номер записи относительно первой записи дорожки, если обращение к записи производится по идентификатору. При обращении по ключу номер должен быть равен нулю
		Относительная адресация (адрес дорожки в шестнадцатеричном формате)	
0—2	TTT	0—FFFFFF	Номер дорожки относительно первой дорожки файла
3	R	0—FF	Номер записи относительно первой записи дорожки, если обращение к записи производится по идентификатору. При обращении по ключу номер должен быть равен нулю
		Абсолютная адресация (адрес дорожки только в шестнадцатеричном формате)	
0	M	00—DD	Номер тома, на котором находится требуемая запись. Тома должны быть пронумерованы последовательно. Номер тома используется для определения логического устройства, на котором установлен том
1—2	BB	0000	Всегда равны нулю
3—4	CC	0000—00C7	Второй байт указывает номер цилиндра, на котором находится запись
5—6	HH	0000—0013	Номер головки указывается во втором байте. Максимально допустимый номер зависит от типа дисков
7	R	00—FF	Порядковый номер записи на дорожке. Он должен быть равен нулю, если обращение к записи производится по ключу

INPUT — метки считываются и проверяются, OUTPUT — метки записываются;

[WRITEID = YES] — использование макрокоманды WRITE ID;

[WRITEKY = YES] — использование макрокоманды WRITE KEY;

[XTNTXIT = имя] — символический адрес подпрограммы пользователя, которой передается управление для обработки информации об участках файла. При этом СУВВ помещает в регистр 1 адрес области, содержащей информацию об участке; формат этой области приведен в табл. 3.13. После обработки этой информации управление возвращается СУВВ с помощью макрокоманды LBRET;

[CONTROL = YES], [DEVICE = шифр], [LABADDR = имя], [VERIFY = YES] — см. 3.3.7;

Таблица 3.13

Структура области, содержащей информацию об участке

Байты	Содержимое области
0	Код типа участка (такой же, как и в операторе EXTENT, которым был определен обрабатываемый участок)
1	Последовательный номер участка
2—5	Нижняя граница участка
6—9	Верхняя граница участка
10—11	Логическое устройство
12—13	Не используются

[MODNAME = имя], [RDONLY = YES] [SEPASMB = YES] — см. 3.3.1.

Макрокоманды DAMOD и DAMODV:

[AFTER = YES] — выполнение функций макрокоманд WRITE RZERO или WRITE AFTER;

[ERREXT = YES] — получение в поле-указателе информации не только об ошибках данных, возникающих при чтении и записи, но и о других неустранимых ошибках ввода-вывода;

[FEOVD = YES] — обработка конца тома для вводных файлов. Операнд указывается только для вводных файлов, которые были созданы последовательным методом доступа с использованием макрокоманды FEOVD;

[HOLD = YES] — выполнение функций захвата дорожки;

[IDLOC = YES] — предоставление пользователю идентификатора записи;

[RECFORM = формат] — формат записей обрабатываемых файлов: UNDEF — неопределенной длины или неблокированные фиксированной длины для DAMOD, FIXUNB — неблокированные фиксированной длины для DAMOD, SPNUNB — неблокированные расширенные или неблокированные переменной длины для DAMODV, VARUNB — неблокированные переменной длины для DAMODV. Если операнд опущен, предполагается RECFORM = FIXUNB для DAMOD, а для DAMODV — RECFORM = VARUNB;

[RELTRK = YES] — использование как относительной, так и абсолютной адресации. Если операнд опущен, то допускается только абсолютная адресация;

[DRONLY = YES], [SEPASMB = YES] — см. 3.3.1.

Макрокоманда OPEN(R) открывает файлы, подготавливая их к обработке. Все используемые участки, даже на разных томах, делаются доступными для обработки одновременно. Для несамоперемещающихся программ употребляется OPEN, для самоперемещающихся — OPENR. Одной макрокомандой можно открыть до шестнадцати файлов. Формат макрокоманды:

$$[\text{имя}]\ \left\{\begin{matrix}\text{OPEN}\\ \text{OPENR}\end{matrix}\right\}\left\{\begin{matrix}\text{имя файла 1}\\ (\text{r1})\end{matrix}\right\}\left[,\left\{\begin{matrix}\text{имя файла 2}\\ (\text{r2})\end{matrix}\right\},\ldots,\left\{\begin{matrix}\text{имя файла n}\\ (\text{rn})\end{matrix}\right\}\right]$$

Макрокоманда READ осуществляет поиск записи и ее считывание. Формат макрокоманды:

$$[\text{имя}]\ \text{READ}\ \left\{\begin{matrix}\text{имя файла}\\(1)\end{matrix}\right\},\left\{\begin{matrix}\text{KEY}\\\text{ID}\end{matrix}\right\}$$

где KEY — осуществляет поиск записи по ключу и считывает только область данных;

ID — осуществляет поиск записи по идентификатору и считывает область ключа, если она имеется, и область данных.

Макрокоманда WRITE осуществляет поиск и запись или обновление записи. Формат макрокоманды:

$$[\text{имя}]\ \text{WRITE}\ \left\{\begin{matrix}\text{имя файла}\\(1)\end{matrix}\right\},\left\{\begin{matrix}\text{KEY}\\\text{ID}\\\text{AFTER, [EOF]}\\\text{RZERO}\end{matrix}\right\}$$

где KEY — осуществляет поиск записи по ключу и обновляет только область данных;

ID — осуществляет поиск записи по идентификатору и обновляет область ключа, если она имеется, и область данных;

AFTER — осуществляет поиск записи 0 указанной дорожки и проверяет, достаточно ли места на указанной дорожке для размещения новой записи. Если на дорожке достаточно места для размещения новой записи, то осуществляется поиск последней записанной записи на этой дорожке, а затем записываются область счетчика, область ключа и область данных сформированной записи. При этом корректируется свободная емкость дорожки в записи 0;

AFTER, EOF — осуществляет запись на указанную дорожку записи EOF за последней записью на этой дорожке;

RZERO — создает на указанной дорожке запись 0 и стирает все записи на дорожке, следующие за этой записью.

Макрокоманда WAITF ожидает завершения передачи данных, запрошенных предыдущей макрокомандой READ или WRITE. Формат макрокоманды:

$$[\text{имя}]\ \text{WAITF}\ \left\{\begin{matrix}\text{имя файла}\\(1)\end{matrix}\right\}$$

Макрокоманда CNTRL устанавливает механизм доступа к дорожке, указанной в поле SEEKADR. Формат макрокоманды:

$$[\text{имя}]\ \text{CNTRL}\ \left\{\begin{matrix}\text{имя файла}\\(1)\end{matrix}\right\},\ \text{SEEK}$$

Макрокоманда FREE освобождает дорожку, захваченную по макрокоманде READ. Адрес дорожки находится в поле SEEKADR. Дорожка, захваченная по макрокоманде WRITE (ID, KEY или AFTER), освобождается автоматически. Формат макрокоманды:

$$[\text{имя}]\ \text{FREE}\ \left\{\begin{matrix}\text{имя файла}\\(1)\end{matrix}\right\}$$

Макрокоманда CLOSE(R) закрывает файлы, запрещая их дальнейшую обработку. Для несамоперемещающихся программ используется

макрокоманда CLOSE, а для самоперемещающихся — CLOSER. Одной макрокомандой можно закрыть до шестнадцати файлов. Формат макрокоманды:

[имя] {CLOSE / CLOSER} {имя файла 1 (r1)} [, {имя файла 2 (r2)}, ..., {имя файла n (rn)}]

3.5. МЕТОД ИНДЕКСНО-ПОСЛЕДОВАТЕЛЬНОГО ДОСТУПА

Назначение и возможности. Метод индексно-последовательного доступа:

- Обрабатывает индексно-последовательные файлы с блокированными или неблокированными записями фиксированной длины (все записи должны быть с ключами), обеспечивая выполнение следующих функций:
  создание индексно-последовательного файла или расширение его за пределы существующей записи с наивысшим ключом (функция LOAD);
  добавление новых записей в уже созданный файл (функция ADD) без его полной реорганизации;
  извлечение записей для обработки в произвольном или последовательном порядке и/или обновление записей (функция RETRVE);
  добавление новых записей и извлечение для обработки и/или обновление (функция ADDRTR).
- Обрабатывает однотомные и многотомные файлы и многофайловые тома.
- Обрабатывает системные метки, для индексно-последовательных файлов метки пользователя не разрешаются.
- При выполнении функций загрузки или последовательного извлечения совмещает обработку с выполнением операций ввода-вывода, если определены две области ввода-вывода.
- Обрабатывает ошибки данных, возникающие при чтении и записи, а также любые другие ошибки ввода-вывода.
- Осуществляет контроль доступа к защищенным файлам и обеспечивает выполнение функции захвата дорожки.
- Определяет записи неверной длины и обеспечивает возможность их обработки с помощью подпрограммы пользователя.

Области переполнения и поля связи. Записи индексно-последовательного файла помещаются на диски в основную область данных. Записи, которые вытесняются из основной области при добавлении новых записей, располагаются в области переполнения. Могут использоваться области переполнения двух типов:

область переполнения цилиндров. Она резервируется на каждом цилиндре основной области данных. Для нее в конце каждого цилиндра отводится одинаковое количество дорожек. В эту область помещаются лишь записи переполнения этого цилиндра;

независимая область переполнения. Эта область одна для всего файла. Располагается она за пределами основной области данных или даже на отдельном томе.

Для размещения записей переполнения файла могут использоваться области переполнения обоих типов либо одна из них.

Для связи записей переполнения, относящихся к одной дорожке, используются поля связи. Они занимают 10 байт области данных каждой записи переполнения и содержат адрес следующей в возрастающем порядке по ключу записи переполнения.

Индексы содержат информацию о расположении записей файла на дорожках диска. При создании индексно-последовательного файла строятся индекс дорожек, индекс цилиндров и главный индекс (необязательно).

Индекс дорожек строится для каждого цилиндра основной области данных и содержит записи индекса только для этого цилиндра.

Каждая запись индекса дорожек содержит наивысший ключ записи данных индексируемой дорожки и адрес этой дорожки. Индекс дорожек располагается на индексируемом цилиндре перед записями данных и занимает часть дорожки, одну или более дорожек.

Индекс цилиндров строится для каждого цилиндра основной области данных и содержит по одной записи индекса для каждого цилиндра. Запись индекса цилиндров содержит наивысший ключ записи данных индексируемого цилиндра и адрес начала этого цилиндра, определяя таким образом адрес индекса дорожек индексируемого цилиндра. Индекс цилиндров должен располагаться за пределами основной области данных, на одном или нескольких цилиндрах и только на одном томе.

Главный индекс является индексом высшего уровня, он строится по желанию пользователя. Каждая запись главного индекса содержит наивысший ключ на дорожке индекса цилиндров и адрес этой дорожки. Главный индекс всегда предшествует индексу цилиндров на томе. Он может занимать один или несколько цилиндров и только на одном томе.

Индикация состояния. После выполнения любой императивной макрокоманды, кроме OPEN, CLOSE, в байте-указателе (30-й байт DTFIS) пользователю предоставляется информация о результате ее выполнения. Эта информация обрабатывается программой пользователя. Причины, вызывающие установку в единицу бит байта-указателя, приведены в табл. 3.14 и 3.15.

Используемые макрокоманды. Декларативные: DTFIS — описание индексно-последовательного файла, ISMOD — описание модуля для индексно-последовательных файлов; императивные: OPEN(R) — открытие файла, SETFL — подготовка файла к загрузке, WRITE — запись записи (загрузка, добавление или произвольное обновление), ENDFL — окончание загрузки файла, READ — чтение записи (произвольное извлечение), WAITF — ожидание завершения передачи данных, SETL — установка нижней границы для последовательного извлечения, GET — получение записи (последовательное) извлечение), PUT — обновление записи (последовательное, ESETL — окончание последовательного извлечения, ERET — возврат управления из подпрограмм пользователя для обработки ошибок ввода-вывода, FREE — освобождение захваченной дорожки, CLOSE(R) — закрытие файла.

Таблица 3.14

Байт-указатель макрокоманды DTFIS (режим LOAD)

Бит	Причина установки бита в 1	Пояснение	Макрокоманды, используемые в режиме LOAD
0	Ошибка на диске	Неисправимая ошибка (кроме записи неверной длины)	SETFL, WRITE, ENDFL
1	Запись неверной длины	При выполнении операции ввода-вывода найдена запись неверной длины	SETFL, WRITE, ENDFL
2	Основная область данных заполнена	При создании и расширении файла предпоследняя дорожка основной области данных оказалась заполненной	WRITE
3	Область индекса цилиндров мала	Обнаружено, что область индекса цилиндров не вмещает все необходимые записи индексов	SETFL
4	Область главного индекса мала	Обнаружено, что область главного индекса не вмещает все необходимые записи индексов	SETFL
5	Дублирование записей	Запись, которая должна загружаться, имеет такой же ключ, как и предыдущая	WRITE
6	Ошибка в последовательности ключей	Запись, которая должна загружаться, имеет ключ, меньший, чем предыдущая	WRITE
7	Переполнение основной области данных	При расширении файла обнаружено, что в основной области данных нет места для записи конца файла	ENDFL

Ниже описываются операнды для каждой декларативной макрокоманды.

Макрокоманда DTFIS:

[CYLOFL = n] — количество дорожек, отводимое в конце каждого цилиндра основной области данных под область переполнения цилиндра;

[DEVICE = шифр] — шифр используемого устройства ввода-вывода, на котором находится основная область данных и области переполнения файла (см. табл. 3.8). Если операнд отсутствует, предполагается устройство ЕС-5052;

DSKXTNT = n — максимальное количество участков для данного файла. В них включаются все участки основной области данных, участок независимой области переполнения и участок для индексов;

[HINDEX = шифр] — шифр устройства (см. табл. 3.8), на котором находится индекс цилиндров и главный индекс (если он используется);

ERREXT = YES — обработка модулем не только ошибок данных, возникающих при чтении и записи, но и других неустранимых ошибок ввода-вывода. При этом возврат в модуль из подпрограммы обработки ошибок ввода-вывода может осуществляться с помощью макрокоманды ERET;

[HOLD = YES] — использование функции захвата дорожки (не допускается для функции загрузки файла);

Таблица 3.15

Байт-указатель макрокоманды DTFIS (режим ADD, RETRVE или ADDRTR)

Бит	Причина установки бита в 1	Пояснение	Макрокоманды, используемые в режимах		
			ADD	RETRVE	ADDRTR
0	Ошибка на диске	Неисправимая ошибка (кроме записи неверной длины)	WRITE NEWKEY	SET, ESETL, WRITE, READ, PUT, GET	SETL, ESETL, WRITE, READ, PUT, GET
1	Запись неверной длины	При выполнении операции ввода-вывода найдена запись неверной длины	WRITE NEWKEY	SETL, ESETL, WRITE, READ, PUT, GET	SETL, ESETL, WRITE, READ, PUT, GET
2	Конец файла	Во время последовательной обработки (TYPEFLE=SEQNTL или RANSEQ) достигнут конец файла	—	GET	GET
3	Запись не найдена	Не найдена искомая запись. Имеет место при произвольной обработке (TYPEFLE=RANDOM или RANSEQ) и выполнении SETL с операндами KEY и GKEY	—	SETL, READ	SETL, READ
4	Недействительный адрес записи	Адрес записи, поставляемой макрокомандой SETL, лежит вне границ основной области данных	—	SETL	SETL
5	Дублирование записей	Ключ добавляемой записи такой же, как и у записи, имеющейся в файле	WRITE NEWKEY	—	WRITE NEWKEY
6	Область переполнения заполнена	Область переполнения цилиндра заполнена и не указана независимая область переполнения или независимая область переполнения уже заполнена и запись не может быть добавлена	WRITE NEWKEY	—	WRITE NEWKEY
7	Запись из переполнения	Указывает, что обрабатывалась запись переполнения (TYPEFLE= =RANDOM, RANSEQ, SEQNTL)	—	READ, GET	READ, GET

[INDAREA = имя] — символический адрес области основной памяти, в которую будет считываться индекс цилиндров (частично или полностью) при добавлении или произвольном извлечении записей;

[INDSKIP = YES] — пропуск записей индекса цилиндров с ключами, меньшими ключа обрабатываемой записи данных, при считывании записей индекса цилиндров в основную память;

[INDSIZE = n] — длина области INDAREA, отводимой для чтения записей индекса цилиндров в основную память. Минимальная длина: n = (N + 3)(KL + 6), где N — количество считываемых за один раз записей индекса цилиндров, KL — длина ключа;

[IOAREAL = имя] — символический адрес области вывода, используемой при создании или расширении либо при добавлении записей к уже созданному файлу. Область вывода должна вмещать область счетчика, ключа и данных, а при добавлении записей — и поле связи;

[IOAREAR = имя] — символический адрес области ввода-вывода, используемой при произвольной обработке (извлечение и обновление) записей. Область ввода-вывода должна вмещать область данных, а при наличии записей переполнения — и поле связи;

[IOAREAS = имя] — символический адрес области ввода-вывода, используемой при последовательной обработке (извлечение и обновление) записей. Эта область должна вмещать область ключа и данных для неблокированных записей или только область данных для блокированных записей, а при наличии записей переполнения — и поле связи;

[IOAREA2 = имя] — символическое имя второй области ввода-вывода, используемой при загрузке или последовательной обработке записей файла. Длина этой области должна совпадать с длиной, заданной операндом IOAREAS при последовательной обработке или операндом IOAREAL при загрузке;

[IOREG = (r)] — номер регистра, в который помещается адрес доступной для обработки записи или адрес области для построения очередной записи при последовательной или произвольной обработке записей непосредственно в области ввода-вывода;

IOROUT = функция — функция, которая должна быть выполнена: LOAD — создание или расширение файла; ADD — добавление записей в файл; RETRVE — извлечение записей из файла для произвольной и/или последовательной обработки; ADDRTR — добавление записей в файл и извлечение записей из файла для произвольной и/или последовательной обработки;

[IOSIZE = n] — длина области, определенной операндом IOAREAL: n = m(8 + KL + BL + 32) + 24, где m — максимальное количество физических записей, считываемых в основную память за одно обращение, KL — длина ключа, BL — длина блока, 8 — длина счетчика, 32 и 24 — длина цепочек CCW. Минимальная длина: n = (8 + KL + BL + 32 + 10) + 24;

[KEYARG = имя] — символический адрес области основной памяти, в которую пользователь должен помещать ключ извлекаемой записи при последовательном или произвольном извлечении по ключу;

KEYLEN = n — длина в байтах ключа записей;

[KEYLOC = n] — номер самого левого байта ключа в каждой логической записи, если записи блокированные;

[MSTIND = YES] — использование главного индекса;

[NRECDS = n] — количество логических записей в блоке. Если записи неблокированные, то предполагается, что операнд равен 1;

RECFORM = формат — формат записей описываемого файла: FIXUNB — неблокированные фиксированной длины, FIXBLK — блокированные фиксированной длины;

RECSIZE = n — длина логической записи. Все логические записи должны быть одинаковой длины и не должны превышать емкость дорожки;

$$\left[\text{TYPEFLE} = \begin{Bmatrix}\text{RANDOM}\\ \text{SEQNTL}\\ \text{RANSEQ}\end{Bmatrix}\right]$$ — тип обработки для функций RETRVE или ADDRTR. Если задан RANDOM, то выполняется произвольная обработка, SEQNTL — последовательная обработка, RANSEQ — произвольная и последовательная обработка;

[WORKL = имя] — символический адрес рабочей области при загрузке или добавлении записей в файл. Область должна вмещать логическую запись. Длина ее определяется значением операнда RECSIZE для блокированных записей, KEYLEN и RECSIZE — для неблокированных записей;

[WORKR = имя] — символический адрес рабочей области при произвольной обработке записей. Эта область должна вмещать логическую запись, длина которой определяется значением операнда RECSIZE;

[WORKS = YES] — использование рабочей области при последовательной обработке записей файла. Эта область должна вмещать логическую запись. Длина ее определяется значением операнда RECSIZE для блокированных записей и суммой значений операндов KEYLEN и RECSIZE — для неблокированных записей;

[MODNAME = имя], [RDONLY = YES], [SEPASMB = YES] — см. 3.3.1;

[VERIFY = YES] — см. 3.3.7.

Макрокоманда ISMOD;

[CORINDX = YES] — обработка индекса цилиндров в основной памяти;

[CORDATA = YES] — добавление записей и связанный с этим сдвиг записей дорожки в основной памяти;

[ERREXT = YES] — обработка не только ошибок данных, возникающих при чтении и записи, но и других неустранимых ошибок ввода-вывода;

[HOLD = YES] — выполнение функции захвата дорожки. Во время загрузки файла эта функция не поддерживается.

[IOAREA2 = YES] — загрузка и последовательная обработка файлов при наличии двух областей ввода-вывода;

IOROUT = функция — выполнение определенных функций: LOAD — создание файла; ADD — добавление записей к файлу;

RETRVE — извлечение (произвольное/последовательное) записей файла; ADDRTR — добавление записей и/или извлечение записей файла (комбинация функций ADD и RETRVE);

[RECFORM = формат] — формат записей файлов, в которые добавляются записи: FIXUNB — неблокированные фиксированной длины, FIXBLK — блокированные фиксированной длины, BOTH — блокированные и неблокированные фиксированной длины. Операнд обязателен при добавлении (IOROUT = ADD или IOROUT = ADDRTR);

$$\left[\text{TYPEFLE} = \begin{Bmatrix}\text{RANDOM}\\ \text{SEQNTL}\\ \text{RANSEQ}\end{Bmatrix}\right]$$ — произвольная (RANDOM), последовательная (SEQNTL) или произвольная и последовательная обработка (RANSEQ). Операнд используется при извлечении записей из файла;

[RDONLY = YES], [SEPASMB = YES] — см. 3.3.1.

Макрокоманда OPEN(R) открывает файлы, подготавливая их к обработке. Все используемые участки, даже на разных томах, подготавливаются к обработке одновременно. Для несамоперемещающихся программ используется OPEN, а для самоперемещающихся — OPENR. Одной макрокомандой осуществляется открытие до шестнадцати файлов. Формат макрокоманды:

$$[\text{имя}]\ \begin{Bmatrix}\text{OPEN}\\ \text{OPENR}\end{Bmatrix}\ \begin{Bmatrix}\text{имя файла 1}\\ (\text{r1})\end{Bmatrix}\left[,\begin{Bmatrix}\text{имя файла 2}\\ (\text{r2})\end{Bmatrix},\ldots,\begin{Bmatrix}\text{имя файла n}\\ (\text{rn})\end{Bmatrix}\right]$$

Макрокоманда SETFL подготавливает к созданию или расширению индексно-последовательный файл. Формат макрокоманды:

$$[\text{имя}]\ \text{SETFL}\ \begin{Bmatrix}\text{имя файла}\\ (0)\end{Bmatrix}$$

Макрокоманда WRITE NEWKEY записывает записи данных и записи индексов на диски при создании или расширении файла или просматривает записи индексов для определения места при добавлении записи в уже созданный файл. WRITE KEY записывает на диски записи данных, предварительно извлеченные макрокомандой READ KEY и обновленные пользователем. Формат макрокоманды:

$$[\text{имя}]\ \text{WRITE}\ \begin{Bmatrix}\text{имя файла}\\ (1)\end{Bmatrix},\begin{Bmatrix}\text{NEWKEY}\\ \text{KEY}\end{Bmatrix}$$

Макрокоманда ENDFL завершает режим загрузки индексно-последовательного файла. Формат макрокоманды:

$$[\text{имя}]\ \text{ENDFL}\ \begin{Bmatrix}\text{имя файла}\\ (0)\end{Bmatrix}$$

Макрокоманда READ считывает запись файла в область ввода-вывода или в рабочую область. Формат макрокоманды:

$$[\text{имя}]\ \text{READ}\ \begin{Bmatrix}\text{имя файла}\\ (1)\end{Bmatrix}$$

Макрокоманда WAITF ожидает завершения передачи данных, запрошенных предыдущей макрокомандой READ KEY, WRITE KEY (произвольное извлечение) или WRITE NEWKEY (добавление). Формат макрокоманды:

$$[\text{имя}]\ \text{WAITF}\ \left\{\begin{matrix}\text{имя файла}\\(1)\end{matrix}\right\}$$

Макрокоманда SETL начинает последовательную обработку записей с места, указанного вторым операндом. Формат макрокоманды:

$$[\text{имя}]\ \text{SETL}\ \left\{\begin{matrix}\text{имя файла}\\(г)\end{matrix}\right\},\left\{\begin{matrix}\left\{\begin{matrix}\text{адрес ID}\\(г)\end{matrix}\right\}\\\text{KEY}\\\text{GKEY}\\\text{BOF}\end{matrix}\right\}$$

где адрес ID — идентификатор обрабатываемой записи;
KEY — ключ обрабатываемой записи;
GKEY — обработка начинается с записи, принадлежащей группе записей, у которых старшие байты ключей совпадают;
BOF — обработка с начала файла.

Макрокоманда GET предоставляет пользователю в области ввода-вывода или в рабочей области следующую в возрастающем порядке по ключу запись файла. Формат макрокоманды:

$$[\text{имя}]\ \text{GET}\ \left\{\begin{matrix}\text{имя файла}\\(1)\end{matrix}\right\}\left[,\left\{\begin{matrix}\text{имя рабочей области}\\(0)\end{matrix}\right\}\right]$$

Макрокоманда PUT последовательно обновляет ранее считанную макрокомандой GET запись файла. Записи обрабатываются непосредственно в области ввода-вывода или в рабочей области. Формат макрокоманды:

$$[\text{имя}]\ \text{PUT}\ \left\{\begin{matrix}\text{имя файла}\\(1)\end{matrix}\right\}\left[,\left\{\begin{matrix}\text{имя рабочей области}\\(0)\end{matrix}\right\}\right]$$

Макрокоманда ESETL завершает последовательную обработку группы записей, начатой по макрокоманде SETL. Формат макрокоманды:

$$[\text{имя}]\ \text{ESETL}\ \left\{\begin{matrix}\text{имя файла}\\(1)\end{matrix}\right\}$$

Макрокоманда ERET осуществляет возврат управления в модуль из подпрограммы пользователя для обработки ошибок ввода-вывода. В модуле выполняются действия, зависящие от операндов. Формат макрокоманды:

$$[\text{имя}]\ \text{ERET}\ \left\{\begin{matrix}\underline{\text{SKIP}}\\\text{IGNORE}\\\text{RETRY}\end{matrix}\right\}$$

где SKIP, IGNORE — ошибка игнорируется и обработка продолжается;
RETRY — повторяется выполнение операции ввода-вывода.

Макрокоманда FREE вызывает освобождение захваченной по макрокоманде READ дорожки при произвольном извлечении записи, которая не подлежит обновлению. В случае обновления записи, извлекаемой произвольно (WRITE KEY), дорожка освобождается автоматически. Так же автоматически освобождается дорожка при последовательной обработке и добавлении записей. Формат макрокоманды:

$$[\text{имя}]\ \text{FREE}\ \left\{\begin{matrix}\text{имя файла}\\(\text{I})\end{matrix}\right\}$$

Макрокоманда CLOSE(R) закрывает файлы, прекращая работу с ними. Для несамоперемещающихся программ используется CLOSE, а для самоперемещающихся — CLOSER. Одной макрокомандой можно закрыть до шестнадцати файлов. Формат макрокоманды:

$$[\text{имя}]\ \left\{\begin{matrix}\text{CLOSE}\\\text{CLOSER}\end{matrix}\right\}\ \left\{\begin{matrix}\text{имя файла 1}\\(\text{r1})\end{matrix}\right\}\left[,\left\{\begin{matrix}\text{имя файла 2}\\(\text{r2})\end{matrix}\right\},\ldots,\left\{\begin{matrix}\text{имя файла n}\\(\text{rn})\end{matrix}\right\}\right]$$

ГЛАВА 4

ОБСЛУЖИВАНИЕ БИБЛИОТЕК

4.1. БИБЛИОТЕКИ

Библиотеки ДОС ЕС содержат компоненты системы в одном из трех состояний соответственно трем существующим этапам подготовки программ к выполнению. Программы, подготовленные на исходных языках программирования, хранятся в библиотеке исходных модулей (SL); результаты обработки исходных модулей трансляторами могут быть помещены в библиотеку объектных модулей (RL); отредактированная, готовая к выполнению программа является составной частью библиотеки абсолютных модулей (CL). В ДОС ЕС могут существовать системные и личные библиотеки SL, RL, CL. Системными называются библиотеки, принадлежащие резидентному файлу. Личная библиотека — библиотека, образующая отдельный файл. Ее структура аналогична соответствующей системной библиотеке.

4.1.1. БИБЛИОТЕКА АБСОЛЮТНЫХ МОДУЛЕЙ

Элементом библиотеки абсолютных модулей является фаза. Библиотека может содержать произвольное количество фаз. Фаза — результат работы Редактора. Она может быть отдельной программой или частью многофазовой программы. В библиотеку абсолютных модулей программы могут помещаться только Редактором. Таким образом, все программы в билиотеке абсолютных модулей имеют абсолютный формат и могут быть вызваны для выполнения в фиксированную для каждой программы область основной памяти. Исключение составляют самоперемещающиеся программы, которые после редактирования могут загружаться для выполнения в различные области основной памяти.

В оглавлении библиотеки абсолютных модулей для каждой фазы содержится запись со следующей информацией: имя фазы; адрес фазы в библиотеке; количество физических блоков, содержащих фазу; количество байт в последнем физическом блоке фазы; адрес основной памяти, начиная с которого фаза загружается для выполнения; адрес точки входа в фазу.

Фазы в библиотеке и соответствующие им записи в оглавлении располагаются в той последовательности, в какой фазы помещались Редактором в библиотеку.

Наименьшей единицей в библиотеке абсолютных модулей, которой может оперировать Библиотекарь, является фаза, более крупной единицей — программа. Совокупность фаз, у которых первые четыре символа имени совпадают, рассматривается Библиотекарем при выполнении определенных функций как программа. В библиотеке абсолютных модулей не могут содержаться фазы, имеющие одинаковые имена.

4.1.2. БИБЛИОТЕКА ОБЪЕКТНЫХ МОДУЛЕЙ

Элементом библиотеки объектных модулей является объектный модуль. Библиотека может содержать любое количество объектных модулей. Каждый объектный модуль — это результат трансляции. Объектные модули могут быть помещены в библиотеку только Библиотекарем. В оглавлении библиотеки объектных модулей для каждого объектного модуля содержится запись со следующей информацией: имя модуля; адрес модуля в библиотеке; количество физических блоков, содержащих модуль; номер версии и модификации (v m).

Модули в библиотеке и соответствующие им записи в оглавлении располагаются в той последовательности, в какой модули помещались в библиотеку.

Наименьшей единицей в библиотеке объектных модулей, которой может оперировать Библиотекарь, является модуль, а более крупной единицей — программа. Совокупность объектных модулей, у которых первые три символа имени совпадают, рассматривается Библиотекарем при выполнении определенных функций как программа. В библиотеке объектных модулей не могут содержаться модули с одинаковыми именами.

Модуль, помещаемый в библиотеку, кроме карт самого модуля ESD, RLD, TXT, REP, SYM, END, может также содержать управляющие операторы Редактора INCLUDE, PHASE, ENTRY и любые другие карты с пробелом в первой колонке.

4.1.3. БИБЛИОТЕКА ИСХОДНЫХ МОДУЛЕЙ

Элементом библиотеки исходных модулей является исходный модуль-книга (в дальнейшем будем называть «книга») — последовательность операторов на любом из языков программирования ДОС ЕС. В библиотеке может содержаться произвольное количество книг. Основное назначение библиотеки исходных модулей — накопление макроопределений, реализующих макрокоманды Ассемблера, и исходных модулей, написанных на языках программирования ДОС ЕС. В библиотеку исходных модулей книги могут помещаться только Библиотекарем.

Наименьшей единицей в библиотеке исходных модулей, которой может оперировать Библиотекарь при выполнении большинства функций, является книга, однако Библиотекарь может работать и с отдель-

ными операторами книги. Более крупной, чем книга, единицей в библиотеке исходных модулей является подбиблиотека. Подбиблиотека — это набор книг на исходном языке, предназначенных для использования определенным транслятором. Каждая книга библиотеки принадлежит некоторой подбиблиотеке. В системе определены подбиблиотеки для Ассемблера (имя подбиблиотеки A) и для Кобола (имя подбиблиотеки C). Благодаря классификации книг по подбиблиотекам книги, написанные на различных языках, могут иметь совпадающие имена.

Операторы исходной книги помещаются в библиотеку в сжатом формате: из них исключаются пробелы. При извлечении книги из библиотеки операторы расширяются до первоначального (80-символьного) формата. В оглавлении библиотеки исходных модулей для каждой книги содержится запись со следующей информацией: имя подбиблиотеки; имя книги; адрес книги в библиотеке; количество физических блоков, содержащих книгу; номер версии и модификации (v. m).

Книги в библиотеке и записи о книгах в оглавлении располагаются в той последовательности, в какой книги помещались в библиотеку. Все подбиблиотеки должны иметь различные имена. В одной подбиблиотеке не могут содержаться книги с одинаковыми именами.

4.1.4. СИСТЕМНЫЕ И ЛИЧНЫЕ БИБЛИОТЕКИ. РАЗМЕР УЧАСТКОВ ДЛЯ БИБЛИОТЕК

Каждая личная библиотека объектных модулей и личная библиотека исходных модулей образуют отдельный файл, содержащий только библиотеку и ее оглавление. Личная библиотека абсолютных модулей образует отдельный файл, содержащий кроме библиотеки и ее оглавления системные материалы. Структура личной библиотеки абсолютных модулей идентична структуре резидентного файла от дорожки 1 цилиндра 0 до конца системной библиотеки абсолютных модулей.

Дисковый пакет может содержать один или более файлов личных библиотек и любые другие файлы. Последовательность расположения файлов произвольна, но файл каждой личной библиотеки должен занимать один непрерывный участок на дисках. Нижняя граница файла личной библиотеки абсолютных модулей обязательно должна совпадать с началом цилиндра.

Личные библиотеки обслуживаются в том случае, если выполнены назначения для соответствующих логических устройств.

Резидентный файл и файлы личных библиотек могут быть записаны на пакеты дисков емкостью 7 (диски типа ЕС-5052) и 29 М байт (диски типа ЕС-5061).

Размер участка для резидентного файла в общем случае определяется следующей формулой:

$$K = X\ (CCL + CRL + CSL) + Z,$$

где K — размер участка для резидентного файла (в дорожках);
X — количество дорожек на цилиндре;
CCL — количество цилиндров, отведенных для библиотеки абсолютных модулей и ее оглавления;

CRL — количество цилиндров, отведенных для библиотеки объектных модулей и ее оглавления;

CSL — количество цилиндров, отведенных для библиотеки исходных модулей и ее оглавления;

Z — константа: для дисков типа ЕС-5052 она равна 19, для дисков типа ЕС-5061—39.

Размер участка для файлов личных библиотек абсолютных, объектных и исходных модулей равен X·CCL, X·CRL и X·CSL соответственно.

При вычислении величин CCL, CRL, CSL рекомендуется пользоваться информацией, приведенной в табл. 4.1. Для накопителей типа ЕС-5052 X = 10, для накопителей типа ЕС-5061 X = 20.

Таблица 4.1

Данные для расчета размеров библиотек и оглавлений

Элемент	Количество блоков на дорожке		Размер блока, байт		Количество записей на дорожке					
					первой	текущей	последней	первой	текущей	последней
	ЕС-5052	ЕС-5061	ЕС-5052	ЕС-5061	ЕС-5052			ЕС-5061		
Оглавление библиотеки CL	8	15	360	360	144	144	143	270	270	269
Библиотека CL	2	4	1728	1688						
Оглавление библиотеки RL	9	17	320	320	175	180	179	335	340	339
Библиотека RL	9	16	322	322						
Оглавление библиотеки SL	16	27	160	160	155	160	159	265	270	269
Библиотека SL	16	27	160	160						

4.2. БИБЛИОТЕКАРЬ

4.2.1. НАЗНАЧЕНИЕ И ФУНКЦИИ

Программы Библиотекаря предназначены для обслуживания библиотек ДОС ЕС. Они позволяют создавать, копировать и объединять библиотеки, корректировать их содержимое, изменять размеры библиотек и получать инфомацию о их состоянии, обеспечивают выполнение описанных ниже функций.

Функции корректировки обеспечивают:

- Каталогизацию элементов в библиотеки. При каталогизации добавляется фаза в библиотеку абсолютных модулей, объектный модуль — в библиотеку объектных модулей и книга — в библиотеку исходных модулей. Каталогизация объектных модулей и книг выполняется Библиотекарем, а каталогизация фаз осуществляется программой Редактор сразу после редактирования.
- Удаление элементов из библиотеки. Оно заключается в том, что в оглавлении соответствующей библиотеки стирается запись, описывающая удаляемую фазу, модуль, книгу, программу, подбиблиотеку.

Удаленные таким образом элементы библиотек перестают существовать для системы. Однако физические блоки на дисках, занимаемые элементами, — удаленные блоки — остаются в библиотеке, и это место не может быть использовано для других целей до выполнения сжатия.

- Переименование элементов библиотек. В результате объектам той или иной библиотеки присваиваются новые имена.
- Обновление книг в библиотеке исходных модулей. Оно выполняется только для библиотеки исходных модулей и позволяет изменять содержимое книг путем добавления, удаления или замены отдельных операторов книг.
- Сжатие (уплотнение) библиотек, которое позволяет исключить удаленные блоки из библиотеки и соответствующие записи из оглавления.
- Установление параметра автоматического сжатия. Оно производится для того, чтобы Библиотекарь следил за количеством блоков, доступных для записи в библиотеку, и осуществлял автоматическое сжатие в том случае, когда это количество становится меньше указанного параметра.
- Распределение библиотек. В результате размеры библиотек и их оглавлений перераспределяются: увеличиваются, уменьшаются, исключаются или добавляются оглавления и библиотеки резидентного файла.

Сервисные функции предназначены для вывода содержимого библиотек на печать и перфорацию с целью визуального контроля за их состоянием и получения копий отдельных элементов или всех библиотек на таких внешних носителях, как перфокарты, магнитная лента, диски. Эти функции обеспечивают:

- Вывод на печать содержимого целых библиотек или отдельных элементов этих библиотек.
- Вывод на печать содержимого транзитного оглавления и оглавлений конкретных или всех библиотек, рассортированных в афлавитном порядке или в порядке следования записей в оглавлении.
- Вывод на печать информации состояния системы и личных библиотек.
- Вывод на печать записей оглавления для любой фазы или группы фаз, рассортированных в алфавитном порядке или в порядке следования записей в оглавлении, а также значения v. m для этих фаз.
- Вывод на перфорацию содержимого целых библиотек или отдельных элементов этих библиотек.
- Одновременный вывод на печать и перфорацию содержимого целых библиотек или отдельных элементов этих библиотек.

Функции создания, слияния и копирования предназначены для получения страховочных копий резидентных пакетов и файлов личных библиотек, для создания новых резидентных пакетов и личных библиотек, для перенесения элементов из системных и личных библиотек в другие системные и личные библиотеки того же типа. Они обеспечивают:

- Создание новых резидентных пакетов.
- Создание личных библиотек.

- Частичное или полное копирование элементов одной библиотеки в другую, созданную ранее, пустую или не пустую (слияние библиотек).
- Получение копий резидентных пакетов или личных библиотек.

Новый резидентный пакет может быть создан только на основе исходного резидентного пакета и всегда содержит какую-то часть элементов исходного. Личные библиотеки могут быть созданы пустыми или на основе библиотек резидентного файла. В последнем случае созданная личная библиотека содержит часть или все элементы соответствующей системной библиотеки. Слияние библиотек позволяет копировать элементы из системных и личных библиотек в другие системные и личные библиотеки. Копии резидентных пакетов и личных библиотек могут быть получены как частные случаи функций создания и копирования.

4.2.2. СПОСОБ ИСПОЛЬЗОВАНИЯ

Имена программ. Функции Библиотекаря реализуются группой программ и распределяются между ними следующим образом:

Имя программы	Функция
MAINT	Корректировка всех библиотек
CSERV	Сервисные функции для библиотеки абсолютных модулей
RSERV	Сервисные функции для библиотеки объектных модулей
SSERV	Сервисные функции для библиотеки исходных модулей
DSERV	Печать оглавлений для всех библиотек
CORGZ	Создание, слияние и копирование библиотек.

Условия использования. Все программы Библиотекаря могут выполняться в фоновом разделе, некоторые из них — в разделах переднего плана в режиме пакетной обработки (табл. 4.2).

Логические устройства описаны в табл. 4.3.

Таблица 4.2

Режимы выполнения программ Библиотекаря

Имя программы	Режим выполнения
MAINT	Разделы фоновый и переднего плана (10 Кбайт). Программа самоперемещающаяся. При работе в разделах переднего плана требует, чтобы было выполнено назначение для SYSCLB
CSERV DSERV	Разделы фоновый и переднего плана (10 Кбайт). Программы самоперемещающиеся
RSERV SSERV	Разделы фоновый и переднего плана (10 Кбайт). Требуется редактирование для раздела
CORGZ	Фоновый раздел (10 Кбайт)

Таблица 4.3

Логические устройства Библиотекаря

Имя устройства	Имя файла	Назначение
SYSRES	IJSYSRS	Резидентный пакет. Используется при выполнении любых функций Библиотекаря для системных библиотек
SYSRDR	IJSYSIN	Ввод операторов Управления заданиями
SYSIPT	IJSYSIN	Ввод управляющих операторов Библиотекаря, объектных и исходных модулей для каталогизации и информации для обновления исходных модулей
SYSLST	IJSYSLS	Вывод результатов сервисной функции печати, состояния системы и личных библиотек и сообщений программисту
SYSLOG	—	Связь с оператором
SYSPCH	IJSYSPH	Вывод результатов сервисной функции перфорации
SYSCLB	IJSYSCL	Личная библиотека абсолютных модулей
SYSRLB	IJSYSRL	Личная библиотека объектных модулей
SYSSLB	IJSYSSL	Личная библиотека исходных модулей
SYS000	IJSYSPS	Исходная личная библиотека исходных модулей при выполнении функции слияния
SYS001	IJSYSPR	Исходная личная библиотека объектных модулей при выполнении функции слияния
SYS002	IJSYSRS	Результирующий файл при выполнении функции копирования или создания резидентного пакета. Исходный и(или) результирующий файл при выполнении функции слияния с участием системных библиотек
SYS003	IJSYSPC	Создаваемая личная библиотека абсолютных модулей. Исходная личная библиотека абсолютных модулей при выполнении функции слияния

Задание на выполнение. Ниже приводится общий вид задания на выполнение программ Библиотекаря. (Примеры заданий на выполнение конкретных функций приведены в 4.2.4.)

Оператор	Пояснение
// JOB	Начало задания
// DLBL // EXTENT	Информация о метках и участках файлов на дисках
// ASSGN ASSGN	Назначение периферийных устройств (для SYSCLB обязательно директива ASSGN без операнда TEMP; DLBL и EXTENT в этом случае должны предшествовать директиве ASSGN)
// EXEC	Выполнение любой программы Библиотекаря
Управляющая информация	Управляющие операторы Библиотекаря, объектные и исходные модули для каталогизации, информация для обновления исходных модулей
/*	Конец шага задания
/&	Конец задания

4.2.3. УПРАВЛЯЮЩИЕ ОПЕРАТОРЫ

Общие правила записи. Структура:

⟨признак⟩ ⟨код⟩ ⟨операнды⟩

где признак — один или несколько пробелов; код — код управляющего оператора.

Таблица 4.4

Коды управляющих операторов Библиотекаря

Код	Функция
CATALR	Каталогизация в библиотеку объектных модулей
CATALS	Каталогизация в библиотеку исходных модулей
DELETC	Удаление из библиотеки абсолютных модулей
DELETR	Удаление из библиотеки объектных модулей
DELETS	Удаление из библиотеки исходных модулей
RENAMC	Переименование в библиотеке абсолютных модулей
RENAMR	Переименование в библиотеке объектных модулей
RENAMS	Переименование в библиотеке исходных модулей
UPDATE	Обновление книг в библиотеке исходных модулей. За оператором UPDATE должны следовать операторы обновления и корректирующие операторы (табл. 4.5)
CONDS	Сжатие библиотек
CONDL	Установление параметра автоматического сжатия
ALLOC	Распределение системных библиотек при корректировке, создании
DSPLY	Печать содержимого библиотек и оглавлений
PUNCH	Перфорация содержимого библиотек
DSPCH	Печать и перфорация содержимого библиотек и оглавлений
NEWVOL	Создание личных библиотек
MERGE	Слияние библиотек
COPYC	Копирование элементов библиотеки абсолютных модулей при создании, слиянии и копировании
COPYR	Копирование элементов библиотеки объектных модулей при создании, слиянии и копировании
COPYS	Копирование элементов библиотеки исходных модулей при создании, слиянии и копировании
COPY	Копирование всех библиотек
COPYI	Копирование фазы начальной загрузки $$A$IPL2

Составные части оператора разделяются не менее чем одним пробелом. Для записи оператора используется одна перфокарта.

Таблица 4.5

Операторы обновления

Оператор обновления	Значение элемента оператора
) ADD n	ADD указывает, что исходные операторы, следующие за оператором ADD, будут добавляться в книгу; n — последовательный номер оператора книги (1—4 десятичные цифры), после которого будут добавляться новые операторы
) DEL n1[,n2]	DEL указывает, что будут удаляться операторы книги; n1, n2 — номера первого и последнего операторов удаляемой группы

Продолжение табл. 4.5

Оператор обновления	Значение элемента оператора
) REP n1[,n2]	REP указывает, что исходные операторы, следующие за оператором REP, представляют собой операторы замены; n1, n2 — номера первого и последнего операторов из той группы, которая должна быть заменена новыми операторами
) END [v.m[,c]]	END определяет конец корректировки данной книги; v. m — значение параметра v. m, которое должно быть присвоено книге после ее обновления; C указывает, что после обновления книге должен быть присвоен параметр C

Функции корректировки запрашиваются следующими управляющими операторами:

```
CATALR  имя[,v.m]
CATALS  подбиб.имя[,v.m[,c]]
{DELETC}
{DELETR} имя1,[,имя2...]
{DELETS}

{DELETC}
{DELETR} прог1.ALL[,прог2.ALL...]
DELETS  подбиб1.ALL[,подбиб2.ALL...]
{DELETR}
{DELETS} ALL

{RENAMC}
{RENAMR} старое имя,новое имя[,старое имя,новое имя...]

RENAMS подбиб.старое имя,подбиб.новое имя[,подбиб.старое имя, подбиб.новое
       имя ...]
UPDATE подбиб.имя[,подбиб.имя1],[v.m][,nn]
CONDS биб1[,биб2...]
CONDL биб1=nnnnn[,биб2=nnnnn...]
ALLOC биб1=цил(дор)[,биб2=цил(дор)]
```

Сервисные функции для библиотек запрашиваются следующими управляющими операторами:

```
DSPLY имя1[,имя2...]
DSPLY подбиб.имя1[,подбиб.имя2...]
DSPLY прог1.ALL[,прог.2.ALL...]
DSPLY подбиб1.ALL[,подбиб2.ALL...]
DSPLY ALL
{PUNCH}
{DSPCH} имя1,[имя2...]
{PUNCH}
{DSPCH} подбиб.имя1[,подбиб.имя2...][,CMPRSD]
{PUNCH}
{DSPCH} прог1.ALL[,прог2.ALL...]
```

```
{PUNCH}
{DSPCH}  подбиб1.ALL[,подбиб2.ALL...][,CMPRSD]

{PUNCH}
{DSPCH}  ALL[,CMPRSD]
```

Сервисные функции для оглавлений запрашиваются следующими управляющими операторами:

```
DSPLY[S]   огл1[,огл2...]
DSPLY[S]   ALL
DSPLY[S]   CD(имя[,nn])
```

Функции создания, слияния и копирования запрашиваются следующими управляющими операторами:

```
ALLOC   биб1=цил(дор)[,биб2=цил(дор)...]
NEWVOL  биб1=цил(дор)[,биб2=цил(дор)...]
MERGE   идент1,идент2

{COPYC}
{COPYR}  имя1[,имя2...]

COPYS  подбиб.имя1[,подбиб.имя2...]

{COPYC}
{COPYR}  прог1.ALL[,прог2.ALL]

COPYS   подбиб1.ALL[,подбиб2.ALL...]

{COPYC}
{COPYR}
{COPYS}  ALL

 COPY  ALL
 COPY1 $$A$IPL2
```

Операнды управляющих операторов. Ниже приводится описание операндов всех управляющих операторов Библиотекаря:

Операнд	Пояснение
имя	Имя фазы, модуля, книги (1 — 8 алфавитно-цифровых символов)
v. m	Параметр v.m для модуля или книги ($0 \leqslant v \leqslant 127$, $0 \leqslant m \leqslant 255$). Если опущен, полагается равным 0.0
подбиб	Имя подбиблиотеки: любой символ из диапазона 0 — 9, A — Z
C	Параметр, указывающий, что при всех дальнейших обновлениях книги требуется контроль ее параметра v. m
прог	Имя программы: 3 символа для библиотеки объектных модулей, 4 символа для библиотеки абсолютных модулей
.ALL	Признак обработки всего элемента, заданного первой частью операнда
ALL	Признак, запрашивающий выполнение функции для всех элементов библиотеки, всех библиотек или всех оглавлений
nn	Режим последовательной нумерации операторов книги после обновления: NO или десятичное число в диапазоне 1 — 10, Значение NO запрещает перенумерацию, значения 1 — 10 задают шаг перенумерации. Если опущен, полагается равным 1
биб	Задает библиотеку: CL — библиотека абсолютных модулей; RL — библиотека объектных модулей; SL — библиотека исходных модулей
nnnnn	Десятичное число (1 — 5 цифр), задающее значение параметра автоматического сжатия. Параметр автоматического сжатия — минимальное количество доступных блоков в библиотеке

цил	Количество цилиндров, которое отводится для библиотеки и ее оглавления
(дор)	Количество дорожек для оглавления
CMPRSD	Признак того, что запрашиваемые элементы библиотеки исходных модулей должны перфорироваться в сжатом формате
огл	Задает оглавления, которые требуется распечатать: CD — оглавление библиотеки абсолютных модулей; RD — оглавление библиотеки объектных модулей; SD — оглавление библиотеки исходных модулей; TD — транзитное оглавление
идент1	Задает исходный файл, с которого данные будут копироваться при слиянии: RES — исходный резидентный файл на SYSRES; NRS — исходный резидентный файл на SYS002; PRV — исходный файл личной библиотеки SL на SYS000 либо исходный файл личной библиотеки RL на SYS001, либо исходный файл личной библиотеки CL на SYS003 (табл. 4.6)
идент2	Задает результирующий файл, в который данные будут копироваться при слиянии: RES — результирующий резидентный файл на SYSRES; NRS — результирующий резидентный файл на SYS002; PRV — результирующий файл личной библиотеки SL—на SYSSLB, личной библиотеки RL—на SYSRLB, личной библиотеке CL—на SYSCLB (см. табл. 4.6)
$$A$IPL2	Имя фазы начальной загрузки, расположенной в резидентном файле вне библиотеки

Таблица 4.6

Сочетания операндов в операторе MERGE

Оператор	Имя файла и логического устройства							
	IJSYSRS SYSRES	IJSYSRS SYS002	IJSYSRL SYSRLB	IJSYSPR SYS001	IJSYSSL SYSSLB	IJSYSPS SYS000	IJSYSCL SYSCLB	IJSYSPC SYS003
MERGE RES, NRS	исх	рез	—	—	—	—	—	—
MERGE NRS, RES	рез	исх	—	—	—	—	—	—
MERGE RES, PRV	исх	—	рез	—	рез	—	рез	—
MERGE NRS, PRV	—	исх	рез	—	рез	—	рез	—
MERGE PRV, RES	рез	—	—	исх	—	исх	—	исх
MERGE PRV, NRS	—	рез	—	исх	—	исх	—	исх
MERGE PRV, PRV	—	—	рез	исх	рез	исх	рез	исх

Примечание. исх — исходный файл; рез — результирующий.

4.2.4. ЗАДАНИЯ НА ВЫПОЛНЕНИЕ КОНКРЕТНЫХ ФУНКЦИЙ

Общая схема задания на выполнение функций корректировки.

```
// JOB имя задания
[ // DLBL для IJSYSRS                    ]   требуются только
[ // EXTENT для IJSYSRS на SYSRES        ]   для ALLOC
// EXEC MAINT
  CATALR модуль [,v.m]
  (модуль для каталогизации)
  CATALS подбиб. книга [,v.m [,C] ]
  (книга для каталогизации)
  DELETC фаза1 [,фаза2...]
```

```
DELETC прог1.ALL[,прог2.ALL...]
DELETR модуль1[,модуль2...]
DELETR прог1.ALL[,прог2.ALL...]
DELETR ALL
DELETS подбиб.книга1[,подбиб.книга2...]
DELETS подбиб1.ALL[,подбиб2.ALL...]
DELETS ALL
RENAMC старое имя,новое имя [,старое имя,новое имя...]
RENAMR старое имя, новое имя, [,старое имя, новое имя...]
RENAMS подбиб. старое имя, подбиб. новое имя [,подбиб. старое имя,
       подбиб.новое имя...]
UPDATE подбиб.книга1,[подбиб.книга2],[v.m] [,nn]
) ADD, ) DEL, ) REP
(операторы для добавления или замены)
 ) END [v.m[,C]]
CONDS биб1[,биб2...]
CONDL биб1=nnnnn [биб2=nnnnn...]
ALLOC биб1=цил (дор) [, биб2=цил(дор)...]
/*
/&
```

Общая схема заданий на выполнение сервисных функций.

1. Сервисные функции для библиотек

```
// JOB имя задания
// EXEC CSERV или // EXEC RSERV, или // EXEC SSERV
(управляющие операторы сервисных функций для библиотек)
/*
/&
```

2. Сервисные функции для оглавлений

```
// JOB имя задания
// EXEC DSERV
(управляющие операторы сервисных функций для оглавлений)
/*
/&
```

Общая схема заданий на выполнение функций создания, копирования и слияния.

1. Создание и копирование резидентных пакетов

```
// JOB имя задания
// ASSGN для SYS002
// DLBL для IJSYSRS
// EXTENT для IJSYSRS на SYS002
// EXEC CORGZ
  ALLOC биб1=цил(дор)[,биб2=цил(дор)...]
(операторы COPYC, COPYR, COPYS или один оператор COPY ALL)
/*
/&
```

2. Создание личных библиотек

```
// JOB имя задания
// ASSGN для SYS003
// ASSGN для SYSRLB
// ASSGN для SYSSLB
// DLBL для IJSYSPC
// EXTENT для IJSYSPC на SYS003
// DLBL для IJSYSRL
// EXTENT для IJSYSRL на SYSRLB
// DLBL для IJSYSSL
```

```
// EXTENT для IJSYSSL на SYSSLB
// EXEC CORGZ
  NEWVOL биб1=цил(дор), биб2=цил(дор),биб3=цил(дор)
(операторы COPYC, COPYR, COPYS)
/*
/&
```

3. Слияние резидентных файлов

```
// JOB имя задания
// ASSGN для SYS002
// DLBL для IJSYSRS
// EXTENT для IJSYSRS на SYS002
// EXEC CORGZ
  MERGE RES,NRS или MERGE NRS, RES
(операторы COPYI, COPYC, COPYR, COPYS)
/*
/&
/
```

4. Слияние резидентного файла с личными библиотеками

```
// JOB имя задания
[// ASSGN для SYS002                    ] опускаются для оператора
[// DLBL для IJSYSRS                    ]   MERGE RES, PRV
[// EXTENT для IJSYSRS на SYS002]
// ASSGN для SYSRLB
// DLBL для IJSYSRL
// EXTENT для IJSYSRL на SYSRLB
// ASSGN для SYSSLB
// DLBL для  IJSYSSL
// EXTENT для IJSYSSL на SYSSLB
// DLBL для  IJSYSCL
// EXTENT для IJSYSCL  на  SYSCLB
  ASSGN для SYSCLB
// EXEC CORGZ
  MERGE NRS, PRV или MERGE RES, PRV
(операторы COPYC, COPYR, COPYS)
/*
/&
```

5. Слияние личных библиотек с резидентным файлом

```
// JOB имя задания
[// ASSGN для SYS002                    ]
[// DLBL для IJSYSRS                    ] опускаются для оператора MERGE
[// EXTENT для IJSYSRS на SYS002]          PRV, RES
// ASSGN для SYS001
// DLBL для IJSYSPR
// EXTENT для IJSYSPR на SYS001
// ASSGN для SYS000
// DLBL для IJSYSPS
// EXTENT для IJSYSPS на SYS000
// ASSGN для SYS003
// DLBL для IJSYSPC
// EXTENT для IJSYSPC на SYS003
// EXEC CORGZ
  MERGE PRV, NRS или MERGE PRV, RES
(операторы COPYC, COPYR, COPYS)
/*
/&
```

6. Слияние личных библиотек

```
// JOB имя задания
// ASSGN для SYSRLB
// DLBL для IJSYSRL
// EXTENT для IJSYSRL на SYSRLB
// ASSGN для SYS001
// DLBL для IJSYSPR
// EXTENT для IJSYSPR на SYS001
// ASSGN для SYSSLB
// DLBL для IJSYSSL
// EXTENT для IJSYSSL на SYSSLB
// ASSGN для SYS000
// DLBL для IJSYSPS
// EXTENT для IJSYSPS на SYS000
// DLBL для IJSYSCL
// EXTENT для IJSYSCL на SYSCLB
  ASSGN для SYSCLB
// ASSGN для SYS003
// DLBL для IJSYSPC
// EXTENT для IJSYSPC на SYS003
// EXEC CORGZ
  MERGE PRV, PRV
(операторы COPYC, COPYR, COPYS)
/*
/&
```

ГЛАВА 5

ТРАНСЛЯЦИЯ И РЕДАКТИРОВАНИЕ ПРОГРАММ

5.1. ТРАНСЛЯЦИЯ

5.1.1. ЯЗЫКИ ПРОГРАММИРОВАНИЯ

Пользователь, работающий с ДОС ЕС, может писать свои программы на следующих языках программирования:

Ассемблере — машинно-ориентированном языке;

РПГ — языке для составления программ, результатами выполнения которых являются разнообразные отчеты;

Коболе — языке программирования задач обработки данных;

Фортране — языке программирования научно-технических задач (Базисном Фортране и Фортране IV);

ПЛ/1 — универсальном языке программирования.

5.1.2. ТРАНСЛЯТОРЫ

Для каждого языка высокого уровня в ДОС ЕС имеется соответствующий транслятор; для программ на Ассемблере можно использовать два транслятора: Ассемблер E и Ассемблер F (Ассемблер F является более быстродействующим, но требует больше памяти).

Входной информацией для транслятора является исходный модуль — программа или часть программы на соответствующем языке программирования. Идентификация начала и конца исходного модуля индивидуальна для каждого языка программирования.

Результатом выполнения любого транслятора ДОС ЕС является один или несколько объектных модулей. Объектный модуль содержит непосредственно текст машинной программы и дополнительную информацию, которая позволяет объединить отдельные (возможно, полученные различными трансляторами) объектные модули в одну готовую для выполнения программу (см. 5.2). Объектные модули еще не являются готовыми к выполнению программами, но в них уже исчезает специфика тех языков программирования, с которых они получены. При трансляции объектные модули могут быть выведены либо на SYSLNK (режим LINK или CATAL в операторе OPTION), либо на

SYSPCH (режим DECK в операторе OPTION). Режим DECK может быть установлен стандартно при генерации системы, режимы LINK и CATAL должны задаваться каждый раз, когда требуется.

В табл. 5.1 приведена следующая информация по трансляторам ДОС ЕС: имя, которое должно использоваться в операторе EXEC при вызове транслятора для выполнения, требования к основной памяти и данные по использованию рабочих файлов. Для рабочих файлов трансляторов могут использоваться диски и магнитные ленты при соблюдении следующих требований:

при работе Кобола для SYS001 должны использоваться обязательно диски;

при работе транслятора ПЛ/1 для всех рабочих файлов должны использоваться однотипные носители (для всех диски или для всех магнитные ленты).

Таблица 5.1

Транслятор	Имя	Память		Рабочие файлы
		BG	F1, F2	
Ассемблер E	ASSEMBLY	14K	16K	SYS001
Ассемблер F	ASSEMBLF	44K	46K	SYS002 SYS003
РПГ	RPG	16K	—	
ПЛ/I	PL/I	14K	14K	
Кобол	FCOBOL	54K	54K	SYS001 SYS002 SYS003 SYS004
Базисный Фортран	FORTRAN	10K	10K	SYS001
Фортран IV	FFORTRAN	40K	40K	SYS001 SYS002

Как правило, рабочие файлы определяются стандартно при генерации системы.

Библиотека исходных модулей используется трансляторами Ассемблер, ПЛ/1 и Кобол для выборки и включения в транслируемую программу готовых исходных модулей.

Библиотека объектных модулей используется для выборки из нее и включения в оттранслированную программу объектных модулей логической системы управления вводом-выводом, объектных модулей стандартных подпрограмм и процедур, образующих библиотеки трансляторов.

Если выполнено назначение для личной библиотеки объектных или исходных модулей, то транслятор использует для поиска модулей сначала личную библиотеку, а затем системную; в противном случае используется только системная библиотека.

5.1.3. ЗАДАНИЕ НА ТРАНСЛЯЦИЮ

Общая схема задания на трансляцию

Оператор	Пояснения
/ JOB	Начало задания
// OPTION	Режимы работы системы и режимы трансляции
// DLBL	Информация о метках и участках файлов на
// EXTENT	дисках
// ASSGN	Назначение периферийных устройств
ASSGN	
// EXEC имя	Значение для операнда ⟨имя⟩ см. в табл. 5.1
Исходные данные	См. примечания
/*	Конец шага
/&	Конец задания

Примечания:

1. Для трансляторов Ассемблер E, Ассемблер F, РПГ, Базисный Фортран и Кобол в качестве исходных данных может быть один исходный модуль на соответствующем языке.

2. Для транслятора Фортран IV в качестве исходных данных может быть последовательность нескольких исходных модулей на Фортране IV.

3. Для транслятора ПЛ/1 в качестве исходных данных может быть последовательность нескольких процедур. Начало каждой процедуры отмечается оператором PROCESS, за оператором PROCESS следует исходный модуль на ПЛ/1.

5.2. РЕДАКТИРОВАНИЕ

Процесс подготовки к выполнению на ЭВМ программ, обработанных трансляторами ДОС ЕС, называется редактированием. Для редактирования программ в ДОС ЕС могут использоваться две программы: Редактор и Отладка.

Программа, отредактированная Редактором, может быть помещена на временное хранение в библиотеку абсолютных модулей для выполнения в этом же задании (режим LINK в операторе OPTION) либо может быть каталогизирована в эту библиотеку (режим CATAL в операторе OPTION). Для размещения программы используется личная библиотека абсолютных модулей, если для нее выполнено назначение; в противном случае используется системная библиотека. Аналогично используются личная и системная библиотеки объектных модулей при выборке модулей для редактирования. Программа, отредактированная Отладкой, может быть помещена только на временное хранение в системную библиотеку абсолютных модулей (режим LINK в операторе OPTION) и выполнена в этом же задании под контролем Отладки, обеспечивающей отладочные функции при ее выполнении.

5.3. РЕДАКТОР

5.3.1. НАЗНАЧЕНИЕ И ВОЗМОЖНОСТИ

Программа Редактор предназначена для подготовки к выполнению на ЭВМ программ, обработанных трансляторами ДОС ЕС, и обеспечивает следующие возможности:

собрать одну программу из объектных модулей или отдельных программных секций этих модулей;

преобразовать текст собранной программы так, чтобы она могла без дополнительной обработки выполняться в заданной при редактировании области основной памяти;

поместить отредактированную программу в библиотеку абсолютных модулей на постоянное или временное хранение.

5.3.2. СПОСОБ ИСПОЛЬЗОВАНИЯ

Имя программы — LNKEDT.

Условия использования. Разделы фоновый и переднего плана 10К. Если разделу отведено не менее 14К памяти, скорость редактирования повышается. Программа самоперемещающаяся.

Логические устройства: см. табл. 5.2.

Таблица 5.2

Устройство	Файл	Назначение
SYSRES	IJSYSRS	Резидентный пакет. Его библиотека абсолютных модулей используется для размещения результатов редактирования в том случае, если не выполнено назначение для личной библиотеки абсолютных модулей
SYSRDR	IJSYSIN	Ввод операторов Управления заданиями и управляющих операторов Редактора
SYSIPT	IJSYSIN	Ввод исходных данных для редактирования
SYSLST	IJSYSLS	Вывод обзора редактирования и сообщений программисту
SYSLOG	—	Связь с оператором
SYSCLB	—	Размещение результатов редактирования
SYSRLB	IJSYSRL	Считывание объектных модулей при редактировании
SYSLNK	IJSYSLN	Входной поток для редактирования
SYS001	IJSYS01	Рабочий файл Редактора. Для него могут использоваться диски или магнитная лента

Задание на выполнение

Оператор	Пояснения
// JOB	Начало задания
// OPTION	Режимы работы системы. В операторе обязательно должен присутствовать операнд LINK или CATAL, наличие других операндов определяется пользователем
// DLBL // EXTENT	Информация о метках и участках файлов на дисках
// ASSGN ASSGN	Назначение периферийных устройств (для SYSCLB обязательна директива ASSGN без операнда TEMP; DLBL и EXTENT в этом случае должны предшествовать директиве)
// LBLTYP	Резервирование области для обработки меток
ACTION PHASE INCLUDE ENTRY	Управляющие операторы, описывающие режимы редактирования и структуру редактируемой программы
// EXEC LNKEDT	Выполнение программы Редактор
/&	Конец задания

5.3.3. УПРАВЛЯЮЩИЕ ОПЕРАТОРЫ

Управляющие операторы считываются Редактором с SYSLNK, а также могут входить в состав тех смешанных и вызывающих модулей, которые во время редактирования считываются Редактором из системной и личной библиотек объектных модулей. Для записи на SYSLNK управляющие операторы Редактора поступают обычно с SYSRDR, могут также поступать с SYSIPT.

Общие правила записи. Структура:

⟨признак⟩ ⟨код⟩ ⟨операнды⟩

где признак — не менее одного пробела; код — ACTION, PHASE, INCLUDE, ENTRY.

Составные части оператора разделяются не менее чем одним пробелом. Для записи оператора используется одна перфокарта.

Оператор режимов редактирования — ACTION. Оператор предназначен для указания режимов редактирования: один оператор задает один режим, для установления нескольких режимов используется соответствующее количество управляющих операторов. Операторы ACTION не являются обязательными. Если они присутствуют, то должны быть первыми среди управляющих операторов. Если они отсутствуют, то предполагаются стандартные режимы.

Формат:

ACTION операнд

Операнд может принимать значения CLEAR, MAP, NOMAP, NOAUTO, CANCEL, BG, F1, F2.

CLEAR. В отредактированной программе должны быть заполнены нулями все области, предназначенные для промежуточных результатов (например, области, определяемые на Ассемблере операторами DS). Режим CLEAR значительно увеличивает время редактирования. Если CLEAR не задан, редактирование выполняется без режима CLEAR.

MAP. Во время редактирования на SYSLST должны быть отпечатаны информация обзора фаз и все диагностические сообщения Редактора. Для получения желаемого результата недостаточно задать операнд MAP в операторе ACTION, необходимо также выполнить назначение для SYSLST. При этом если назначение для SYSLST выполнено, то режим MAP предполагается Редактором и в том случае, когда операторы ACTION отсутствуют или не содержат никаких указаний о режиме MAP.

NOMAP. Устройство SYSLST недоступно для вывода информации. Если задан режим NOMAP, то (независимо от назначения для SYSLST) обзор фаз не печатается вообще, а диагностические сообщения Редактора печатаются на SYSLOG. Если назначение для SYSLST не выполнено, а операторы ACTION отсутствуют или не содержат никаких указаний о режиме MAP, то Редактором предполагается также режим NOMAP.

NOAUTO. Отменяет режим автоматического включения объектных модулей (AUTOLINK) во время редактирования всех фаз программы. Если NOAUTO не задан, предполагается режим AUTOLINK. Это значит, что в конце редактирования каждой фазы делается попытка

найти в личной и системной библиотеках объектных модулей модули, имена которых равны именам неразрешенных (не получивших конкретных значений) элементов ER. Найденные модули включаются в фазу.

CANCEL. Редактор должен выполнить снятие задания, если во время редактирования встретится любая из ошибок с номерами 2100I—2170I.

BG, F1, F2. Указывают, что адреса загрузки всех фаз редактируемой программы должны вычисляться относительно начала заданного раздела. В случае разделов F1 и F2 учитывается и длина области сохранения. Использование операндов BG, F1, F2 позволяет выполнять программу в разделах, отличных от тех, в которых выполнялось ее редактирование. Операнды BG, F1, F2 могут использоваться только в системе с мультипрограммированием, причем для заданного раздела в момент редактирования должна быть выделена память.

Оператор определения фазы — PHASE. Оператор предназначен для указания имени фазы, адреса ее загрузки и режима определения внешних ссылок (AUTOLINK). В одном шаге задания могут использоваться не более 119 операторов. Формат:

PHASE имя фазы, адрес загрузки[,NOAUTO]

где имя фазы — имя, под которым фаза будет каталогизирована в библиотеку абсолютных модулей; содержит от 1 до 8 алфавитно-цифровых символов. Первым символом имени фазы не должен быть символ «*». Для сокращения времени, затрачиваемого на вызов фаз многофазовой программы, необходимо соблюдать следующие правила:

первые 4 символа имен фаз многофазовой фоновой программы должны быть одинаковыми для всех фаз;

первые 3 символа имен фаз многофазовой программы переднего плана должны быть символами FGP.

NOAUTO — запрещается использование режима AUTOLINK во время редактирования данной фазы.

адрес загрузки — может задаваться одним из приведенных ниже шести выражений. Элемент «смещение» во всех выражениях задается самоопределенным термом, который может быть: шестнадцатеричным числом X'hhhhhh', десятичным числом ddddddd, величиной nK (K = = 1024).

Если редактируемая программа имеет общую область, то заданный данным операндом адрес загрузки увеличивается на длину общей области. Кроме того, при формировании окончательного значения адреса загрузки учитывается длина области меток (если был задан оператор LBLTYP) и длина области сохранения (если редактируется программа переднего плана).

Форматы адреса загрузки:

симв.имя[(имя фазы)] [±смещение]	(а)
$ ±смещение	(б)
ROOT	(в)
*[±смещение]	(г)
±смещение	(д)
F+адрес раздела	(е)

Формула (а):

симв.имя — может быть ранее определенным именем фазы, программной секции или входным именем. Указывает, что адрес загрузки должен вычисляться относительно действительного значения указанного имени;

(имя фазы) — указывает ранее определенное имя фазы, к которой принадлежит имя программной секции или входное имя, заданное первым элементом операнда. Должно задаваться в тех случаях, когда имя секции или входное имя может встретиться более чем в одной фазе программы;

±**смещение** — может быть как положительным, так и отрицательным числом.

Формула (б):

S — способ вычисления адреса загрузки, заданного элементом S, зависит от режима редактирования. Если в ACTION был задан режим F1(F2), то S означает адрес начала соответствующего раздела, увеличенный на длину области сохранения. В противном случае S означает адрес первого двойного слова после конца ядра Супервизора;

+ **смещение** — может быть только положительным числом.

Формула (в):

ROOT — формируемая фаза является корневой фазой. Значение адреса загрузки определяется так же, как и по элементу S предыдущего выражения. Кроме того, ROOT влияет на режим редактирования: если программная секция включена в корневую фазу, то игнорируется запрос на ее включение в другие фазы этой программы. Операнд ROOT может быть задан только в первом для данной программы операторе PHASE, в противном случае ROOT рассматривается как символическое имя.

Формула (г):

***** — адрес загрузки должен вычисляться относительно адреса первого двойного слова после конца предыдущей фазы. Если «*» используется в первом для данной программы операторе PHASE, то она трактуется, как элемент S;

± **смещение** — для первого оператора PHASE терм смещения может быть только положительным.

Формула (д):

+ **смещение** — позволяет задавать в качестве адреса загрузки абсолютный конкретный адрес основной памяти. Значение + 0 используется для задания адреса загрузки самоперемещающейся программы.

Формула (е):

F — редактируемая фаза является фазой программы переднего плана;

+ **адрес раздела** — указывает значение, которое Редактор должен использовать в качестве начала раздела переднего плана. Распределение памяти между разделами в момент редактирования в этом случае не имеет значения (в частности, оно может быть и нулевым). Адрес загрузки фазы вычисляется как сумма адреса раздела и длины области сохранения (учитываются также длина общей области и области меток). Адрес раздела задается таким же самоопределенным термом, как и смещение, и может быть только положительным.

Оператор описания структуры фазы — INCLUDE предназначен для указания модулей и отдельных программных секций, которые должны включаться в фазу. По крайней мере один оператор описания структуры фазы должен следовать за оператором описания фазы. Для построения одной фазы может использоваться произвольное количество операторов. Формат:

INCLUDE [имя модуля], (список секций)]

где имя модуля — указывает имя модуля (1 — 8 алфавитно-цифровых символов), который должен быть найден в личной или системной

библиотеке объектных модулей и включен в фазу. Найденный модуль не всегда полностью включается в фазу: способ его включения зависит от второго операнда;

список секций — задается в виде (секция 1, секция 2, ...). Задает имена секций, которые должны быть выбраны из указанного модуля и включены в фазу. Общее количество имен не должно превышать пяти. Если операнд опущен, модуль целиком включается в фазу. Если первый операнд опущен (на его месте стоит запятая), а второй присутствует, то из текущего (очередного) модуля на SYSLNK выбираются заданные вторым операндом секции и включаются в фазу. Если опущены оба операнда, то это означает, что программа Управление заданиями должна переписать с SYSIPT на SYSLNK всю информацию до оператора/*.

Оператор точки входа — ENTRY предназначен для описания точки входа в редактируемую программу и индикации конца управляющих операторов. Оператор не является обязательным. Он присутствует, если требуется указать точку входа; должен быть последним среди управляющих операторов. Формат:

ENTRY [точка входа]

где точка входа — символическое имя точки входа в первую фазу редактируемой программы, которое должно быть именем программной секции или входным именем, определенным в первой фазе (от 1 до 8 алфавитно-цифровых символов). Если ENTRY опущен или не содержит операнда, то в качестве адреса точки входа используется адрес с первой карты END, на которой задан адрес точки входа. Если ни одна карта END не содержит адреса, то в качестве адреса точки входа в фазу используется адрес загрузки фазы. Для всех фаз программы, кроме первой фазы, правила выбора точки входа аналогичны, за исключением того, что оператор ENTRY не принимается во внимание. Если в карте END или в операторе ENTRY в качестве точки входа указано внешнее имя, то оно должно быть определено к моменту поступления END или ENTRY.

5.4. ОТЛАДКА

5.4.1. НАЗНАЧЕНИЕ И ФУНКЦИИ

Программа Отладка служит для редактирования и выполнения в режиме отладки программ пользователя. Она выполняет следующие функции:

- Редактирование отлаживаемой программы, поступающей на отладку в виде объектных модулей.
- Корректировку текста отлаживаемой программы во время редактирования. При корректировке обеспечиваются замена отдельных команд и констант отлаживаемой программы и добавление новых команд.
- Контроль за выполнением отредактированной отлаживаемой программы для обеспечения следующих отладочных функций:

отображение памяти и регистров во время выполнения отлаживаемой программы. При отображении памяти на печать выводится содержимое заданной области основной памяти в любом из допустимых форматов. При отображении регистров на печать выводятся содержимое общих регистров, регистров плавающей запятой и байты 24 — 127 основной памяти;

разгрузка памяти при завершении отлаживаемой программы;

печать списка фаз отлаживаемой программы, вызываемых в память во время ее выполнения. Эта функция не может быть отменена программистом;

частичная защита от зацикливания отлаживаемой программы, которая обеспечивает своевременное снятие отлаживаемой программы в случае ее зацикливания.

Возможность получить на печати все результаты отладки, накопленные до точки аварии, если при выполнении отлаживаемой программы возникла аварийная ситуация, не позволяющая завершить отладочное задание (функция аварийного продолжения).

5.4.2. СПОСОБ ИСПОЛЬЗОВАНИЯ

Имя программы. ATLEDT — имя программы Отладка, ATLECONT — имя программы, выполняющей функцию аварийного продолжения.

Условия использования. Фоновый раздел, для разделов переднего плана должно быть установлено нулевое распределение. Требуется редактирование для фонового раздела.

Логические устройства: см. табл. 5.3.

Таблица 5.3

Устройство	Файл	Назначение
SYSRES	IJSYSRS	Резидентный пакет. Его библиотека абсолютных модулей используется для временного хранения отредактированной отлаживаемой программы
SYSRDR	IJSYSIN	Ввод операторов Управления заданиями и управляющих операторов для редактирования
SYSIPT	IJSYSIN	Ввод управляющих операторов Отладки и исходных данных для редактирования
SYSLST	IJSYSLS	Вывод обзора редактирования, результатов выполнения отладочных функций и сообщений программисту
SYSLOG	—	Связь с оператором
SYSRLB	IJSYSRL	Считывание объектных модулей при редактировании
SYSLNK	IJSYSLN	Входной поток для редактирования
SYSLNK	IJSYSAT	Накопление результатов выполнения отладочных функций для последующего вывода на печать
SYS001	IJSYS01	Рабочий файл

Задание на выполнение Отладки

Оператор	Пояснения
// JOB	Начало задания
ALLOC	Вся память, доступная пользователю, отводится для фонового раздела (F1 = 0K, F2 = 0K)

// OPTION	Режимы работы системы. В операторе должен обязательно присутствовать операнд LINK, наличие других операндов определяется пользователем
// ASSGN ASSGN	Назначение периферийных устройств
// DLBL // EXTENT	Информация о метках и участках файлов на дисках
// LBLTYP	Резервирование области для обработки меток
ACTION PHASE INCLUDE ENTRY	Управляющие операторы Редактора, описывающие режим редактирования и структуру редактируемой программы. В операторе ACTION не допускается использование операндов F1 и F2. В операторе PHASE в качестве адреса загрузки не допускается использование формата (F + абсолютный адрес)
// EXEC ATLEDT	Выполнение программы Отладка
./ PHQ ./ CSQ ./ TR ./ EXC ./ ADD ./ CON ./ PCC	Управляющие операторы Отладки, запрашивающие выполнение конкретных отладочных функций. Оператор./ PCC должен быть последним среди управляющих операторов Отладки
Операторы Управления заданиями	Любые операторы Управления заданиями, необходимые для подготовки отлаживаемой программы к выполнению, кроме операторов: // OPTION LINK, // OPTION CATAL, // EXEC LNKEDT, // EXEC ATLEDT
// EXEC	Выполнение отлаживаемой программы
Данные	Данные для отлаживаемой программы, если нужно
./ ATEOF	Конец задания для Отладки
ALLOC	Восстанавливается распределение памяти между разделами
/&	Конец задания

Задание на выполнение программы аварийного продолжения

Оператор	Пояснения
// JOB	Начало задания
// ASSGN	Восстановление назначений периферийных устройств
// DLBL // EXTENT	Информация о метках и участках файлов на дисках
// EXEC ATLECONT	Выполнение программы аварийного продолжения
Перфокарты отладочного и последующих заданий	Перфокарты отладочного и последующих заданий, оставшиеся не введенными к моменту аварии

5.4.3. УПРАВЛЯЮЩИЕ ОПЕРАТОРЫ

Управляющие операторы делятся на две самостоятельные группы: управляющие операторы для редактирования отлаживаемой программы, полностью идентичные управляющим операторам Редактора (см. 5.3.3), и отладочные управляющие операторы: операторы определения фаз и программных секций; операторы отображения; операторы корректировки; оператор конца управляющих операторов Отладки; оператор конца отладочного задания.

Примечание. За управляющими операторами корректировки обязательно должны следовать карты с корректирующей информацией, которая записывается в шестнадцатеричном формате, начиная с любой колонки карты. Непосредственно перед корректирующей информацией записывается символ «*», указывающий начало корректирующей информации на карте.

Общие правила записи. Структура:

	⟨признак⟩ ⟨код⟩ ⟨операнды⟩
Признак	./
Код	PHQ, CSQ, TR, EXC, ADD, CON, PCC, ATEOF

Составные части оператора разделяются пробелом. Для записи оператора отображения могут использоваться перфокарты продолжения. Для записи прочих операторов используется одна перфокарта.

Операторы описания фаз и программных секций. Эти операторы предназначены для указания имен фаз и программных секций, относительно начала которых должны вычисляться адреса в последующих операторах корректировки и отображения; не являются обязательными операторами. Если они отсутствуют, то адреса в операторах корректировки и отображения вычисляются относительно начала первой программной секции в первой фазе отлаживаемой программы. Если присутствуют, то должны предшествовать относящимся к определяемой фазе операторам корректировки и отображения. В одном шаге задания может использоваться произвольное количество операторов. Формат:

```
./ fn name
```

где ./ — признак управляющего оператора Отладки;

fn — код оператора: PHQ — для определения фазы; CSQ — для определения программной секции;

name — имя фазы или программной секции (1 — 8 алфавитно-цифровых символов), относительно которых должны вычисляться адреса всех последующих операторов отображения и корректировки (до нового оператора PHQ или CSQ).

Оператор отображения TR. Оператор предназначен для запроса функций отображения. Является необязательным. В одном шаге задания может использоваться произвольное количество операторов. Формат:

```
./ TR tpa,(anf,end,h),fn,unt,ob,(anf,end,h),hsanz,reganz
```

где tpa — адрес точки выполнения функции отображения. Возможные значения:

шестнадцатеричный (1 — 6 знаков);

символический (1 — 8 символов); используется, если объектные модули получены Ассемблером в режиме // OPTION SYM;

выражение из алгебраической суммы символического адреса и десятичной константы (длина константы 1 — 4 цифры);

, (anf, end, h) — общий параметр активизации для всех функций отображения, запрошенных данным оператором TR. Имеет практический смысл только тогда, когда точка находится внутри программного цикла. Компонент anf задает число выполнений цикла, при котором

Отладка должна начать выполнение функций отображения в данной точке. Компонент end задает число выполнений цикла, при достижении которого Отладка должна прекратить выполнение функций отображения в данной точке. Компонент h задает шаг (относительно числа циклов), с которым должны выполняться функции отображения в данной точке внутри интервала (anf, end). Если конкретная функция отображения имеет собственный параметр активизации, то общий параметр активизации игнорируется. Каждый компонент представляет собой десятичное число (1 — 4 цифры). Общий параметр активизации может быть опущен, но если он задается, то должны присутствовать все компоненты;

, fn — задает вид функции отображения:

DSP — отображение памяти; PNL — отображение регистров; DPL — отображение памяти и регистров;

, unt, ob — адреса верхней и нижней границ области памяти, подлежащей отображению. Возможные значения:

десятичный (1 — 6 цифр);

шестнадцатеричный (1 — 6 знаков);

символический (1 — 8 символов). Используется, если объектные модули получены Ассемблером в режиме // OPTION SYM;

выражение из алгебраической суммы символического адреса и десятичной константы (длина константы 1 — 4 цифры);

составной адрес: P. C. adr

P — имя фазы; C — имя секции; adr — относительный адрес (шестнадцатеричный, символический или выражение).

Параметр необязателен для функций DSP и DPL; недействителен для функции PNL;

, (anf, end. h) — функциональный параметр активизации, который распространяется на конкретную функцию отображения. Компоненты anf, end, h имеют тот же смысл, что и соответствующие компоненты общего параметра активизации. Параметр может быть опущен, но если задается, то должны присутствовать все компоненты;

, hsanz — задает формат отображения памяти: C — символьный; B — шестнадцатеричный с символьным (в две строки); E — десятичные числа с плавающей запятой (длиной в слово); D — десятичные числа с плавающей запятой (длиной в двойное слово); F — десятичные числа с фиксированной запятой (длиной в слово); H — десятичные числа с фиксированной запятой (длиной в полуслово); M — шестнадцатеричный с мнемоникой (в две строки). При указании мнемоники каждые два байта интерпретируются как мнемонический код команды на Ассемблере. Параметр необязателен для функций DSP и DPL, недействителен для функции PNL. Если параметр опущен, предполагается шестнадцатеричный формат;

, reganz — режим отображения регистров; в качестве значения может использоваться любая комбинация записанных последовательно нижеприведенных значений: G — общие регистры; F — регистры плавающей запятой; S — байты 24 — 127 основной памяти. Параметр необязателен для функций PNL и DPL; недействителен для функции DSP. Если параметр опущен, предполагается G.

Операторы корректировки. Операторы предназначены для запроса функций корректировки; являются необязательными. За операторами корректировки должны следовать карты с корректирующей информацией. В одном шаге задания может использоваться произвольное количество операторов. Формат:

./ fn,tpa,nn

где fn — код оператора, задает конкретную функцию корректировки:
ADD — добавление команд; EXC — замена команды; CON — замена константы;

tpa — адрес точки корректировки. Возможные значения:
шестнадцатеричный (1 — 6 знаков);
символический (1 — 8 символов); используется, если объектные модули получены Ассемблером в режиме // OPTION SYM;
выражение из алгебраической суммы символического адреса и десятичной константы (длина константы 1 — 4 цифры);

,nn — задает номер колонки на корректирующих картах, в которой записан символ «*», указывающий начало корректирующей информации. Десятичное число (1 — 2 цифры).

Карты с корректирующей информацией. Корректирующая информация записывается в шестнадцатеричном представлении. Признаком начала служит символ «*», признаком конца — символ «пробел». Карты продолжения не используются.

Оператор конца управляющих операторов Отладки — PCC указывает, что завершается ввод управляющих операторов Отладки, и задает режимы разгрузки памяти. После его ввода могут обрабатываться операторы Управления заданиями для подготовки отлаживаемой программы к выполнению. Является обязательным. Должен быть последним среди отладочных операторов. В одном шаге задания может использоваться один оператор. Формат:

./ PCC norm,anorm,n

где PCC — код оператора;

norm — задает формат разгрузки памяти при нормальном завершении отладочного задания. В качестве значения может использоваться любая комбинация записанных последовательно ниже приведенных значений X, A, M и S : N — разгрузка не выполняется; X — в разгрузку включается область Супервизора; A — шестнадцатеричный с символьным (в две строки); M — шестнадцатеричный с мнемоникой (в две строки); S — шестнадцатеричный с указанием символических имен: над строкой шестнадцатеричного представления печатаются символические имена (первый символ имени располагается над адресуемым им байтом). Формат S действителен только для объектных модулей, полученных Ассемблером в режиме // OPTION SYM. Если операнд опущен, предполагается шестнадцатеричный формат;

, anorm — задает формат разгрузки при ненормальном завершении отлаживаемой программы. В качестве значения может использоваться любая комбинация записанных последовательно нижеприведенных значений: A — шестнадцатеричный с символьным (в две строки); M —

шестнадцатеричный с мнемоникой (в две строки); S — шестнадцатеричный с указанием символических имен;

, n — параметр защиты от зацикливания. Задает общее число проходов программы через точки выполнения отображений, при превышении которого требуется прекратить выполнение отладочного задания. Если опущен, предполагается 50.

Оператор конца отладочного задания — ATEOF предназначен для указания на конец отладочного задания и используется для определения конца прогона перфокарт задания после выполнения программы аварийного продолжения. Является обязательным. Должен следовать после перфокарт шага выполнения отлаживаемой программы. В одном отладочном задании может использоваться один оператор. Формат:

```
./ ATEOF
```

5.5. ПРИМЕРЫ ЗАДАНИЙ НА РЕДАКТИРОВАНИЕ

Общие условия. SYSIPT=SYSRDR. Для рабочих файлов используются стандартные назначения. Личные библиотеки объектных, исходных и абсолютных модулей находятся на пакете с регистрационным номером PRIVAT. В пояснениях объектный модуль, полученный после трансляции, называется именем исходного модуля.

5.5.1. ТРАНСЛЯЦИЯ, РЕДАКТИРОВАНИЕ С КАТАЛОГИЗАЦИЕЙ И ВЫПОЛНЕНИЕ

Условия. Объектные модули MOD1 и MOD3 находятся в личной RL, объектный модуль MOD2 хранится на перфокартах.

```
   // JOB CATALOG
   // OPTION CATAL
 3 // DLBL IJSYSRL,'PRIVATE RL',99/365,SD
 4 // EXTENT SYSRLB,PRIVAT,1,1,10,190
 5 // ASSGN SYSRLB,X'192'
 6 // DLBL IJSYSCL,'PRIVATE CL',99/365,SD
 7 // EXTENT SYSCLB,PRIVAT,1,1,200,300
 8 // ASSGN SYSCLB,X'192'
 9 // DLBL IJSYSSL,'PRIVATE SL',99/365,SD
10 // ASSGN SYSSLB,PRIVAT,1,1,500,400
11 // ASSGN SYSSLB,X'192'
     ACTION NOMAP,CANCEL
     PHASE PROG, ROOT
     INCLUDE MOD1
   // EXEC ASSEMBLF
    <исходный модуль A>
   /*
     PHASE PROG1,*,NOAUTO
     INCLUDE
    <объектный модуль MOD2>
   /*
   // EXEC FFORTRAN
    <исходный модуль F1>
    <исходный модуль F2>
```

```
   PHASE PROG2, ENTR1 (PROG1)
   INCLUDE MOD3, (CSECT1, CSECT4)
   ENTRY
// EXEC LNKEDT
/&
// JOB PROGEXEC
   . . . . . .
// EXEC PROG
   <исходные данные>
/&
```

Результат редактирования. Каталогизированная в личную библиотеку абсолютных модулей программа следующей структуры:

корневая фаза PROG построена из объектных модулей MOD1 и A, загружается в начало BG;

фаза PROG1 построена из объектных модулей MOD2, F1, F2; загружается следом за PROG;

фаза PROG2 построена из секций CSECT1 и CSECT4 модуля MOD3 и загружается, перекрывая фазу PROG1 с адреса ENTR1.

Примечание. Если не используется личная RL, опускаются операторы 3, 4, 5 задания; если не используется личная CL или SL, опускаются операторы 6, 7, 8 или 9, 10, 11.

5.5.2. ТРАНСЛЯЦИЯ, РЕДАКТИРОВАНИЕ ДЛЯ ОТЛАДКИ И ВЫПОЛНЕНИЕ

Условия. Объектные модули MOD1 и MOD2 хранятся на перфокартах. Исходный модуль A транслируется Ассемблером в режиме SYM.

```
 // JOB  TEMPTEST
    ALLOC F1=0K,F2=0K
 // OPTION LINK,SYM
4 // DLBL  IJSYSRL,'PRIVATE  RL',99/365,SD
5 // EXTENT SYSRLB,PRIVAT,1,1,10,190
6 // ASSGN  SYSRLB,X'192'
7 // DLBL  IJSYSSL,'PRIVATE SL',99/365,SD
8 // EXTENT SYSSLB, PRIVAT,1,1,500,400
9 // ASSGN  SYSSLB,X'192'
    ACTION MAP
    PHASE  PHAS1,X'2800'
    INCLUDE
    <объектный модуль MOD1>
    <объектный модуль MOD2>
 /*
    PHASE PHAS2,*
 // EXEC RPG
    <исходный модуль R>
 /*
    PHASE PHAS3,*
 // EXEC ASSEMBLF
    <исходный модуль A>
 /*
   ENTRY
 // EXEC ATLEDT
./ PHQ PHAS2
```

```
./ TR X'2830',(1,50,2),DSP,X'285A',X'288E',,B
./ TR X'2840',(1,50,2),PNL,,GFS
./ PHQ PHAS3
./ TR CHECK,(1,80,10),PNL,,GFS
./ PCC N,A,2000
// EXEC
   <исходные данные для отлаживаемой программы>
./ ATEOF
   ALLOC F1=28K, F2=14K
/&
```

Результат редактирования. Помещенная на временное (в течение отладочного задания) хранение в системную билиотеку абсолютных модулей программа следующей структуры:

первая фаза PHAS1 построена из объектных модулей MOD1 и MOD2; загружается для выполнения с адреса X'2800';

вторая фаза PHAS2 построена из объектного модуля R; загружается следом за PHAS1;

третья фаза PHAS3 построена из объектного модуля A; загружается следом за PHAS2.

Для выполнения отладочных функций в PHAS2 отмечены точки X'2830' и X'2840'; в PHAS3 отмечена точка с символическим адресом CHECK.

Результаты отладочных функций. При выполнении отлаживаемой программы выводится на печать в шестнадцатеричном и символьном форматах область X'285A' — X'288E' при проходах программы через точку X'2830' (с шагом 2, конец при проходе 50). При проходе (с шагом 2, конец при проходе 50) через точку X'2840' выводится на печать содержимое общих регистров, регистров плавающей запятой и байты 24 — 127 основной памяти. Аналогичный вывод происходит при проходе (с шагом 10, конец при проходе 80) через точку CHECK.

Если число проходов через точки выполнения отладочных функций превысит 2000, отладочное задание будет снято. При любом снятии отладочного задания будет выполнена разгрузка памяти в шестнадцатеричном и символьном форматах.

Примечание. Если не используется личная RL или SL, опускаются операторы 4, 5, 6 или 7, 8, 9 задания.

ГЛАВА 6

ОБСЛУЖИВАНИЕ НОСИТЕЛЕЙ ДАННЫХ

Для обслуживания носителей данных—пакетов дисков и магнитных лент—предназначены специальные программы ДОС ЕС, описанные в этой главе. Некоторые функции по обслуживанию носителей данных выполняет программа Сервис (см. гл. 7).

Подготовка магнитных лент в зависимости от их дальнейшего использования выполняется одним из двух способов: если лента будет использоваться для файла (или файлов) со стандартными метками, выполняется ее инициализация с помощью программы Инициализировать ленту (см. разд. 6.6); в противном случае на ленту достаточно записать ленточную марку с помощью управляющего оператора или директивы МТС (чистую ленту использовать нельзя).

6.1. ИНИЦИАЛИЗИРОВАТЬ ДИСК

6.1.1. НАЗНАЧЕНИЕ И ФУНКЦИИ

Программа предназначена для подготовки пакетов дисков перед использованием их на машинах ЕС ЭВМ. Она выполняет следующие функции:

- Проверяет имеющееся на пакете оглавление тома для обнаружения файлов с неистекшим сроком хранения, чтобы избежать их разрушения.
- Выполняет анализ поверхности пакета дисков, если этот режим задан пользователем. При первой инициализации пакета анализ обязателен.
- В процессе анализа поверхности генерирует на каждой дорожке собственный адрес и запись описания дорожки (запись 0), очищая оставшуюся часть каждой дорожки.
- Отмечает дефектные дорожки (обнаруженные в результате анализа или указанные пользователем), при этом дефектным основным дорожкам назначает запасные.
- Форматизирует две записи программы Начальная загрузка (записи 1 и 2 дорожки 0 цилиндра 0), с тем чтобы в них можно было записать

часть Начальной загрузки, если пакет будет использоваться в качестве резидентного.

- Записывает обязательную системную метку тома (запись 3 дорожки 0 цилиндра 0), которая используется для идентификации пакета, и вслед за ней может записать до 7 дополнительных меток тома.
- Форматизирует оглавление тома путем заполнения дорожек пустыми записями, формат которых соответствует формату системных меток файла (системные метки записываются в оглавление тома во время открытия выводных файлов).

6.1.2. СПОСОБ ИСПОЛЬЗОВАНИЯ

Имя программы — INTD.

Условия использования. Разделы фоновый и переднего плана (10K), требуется редактирование для разделов переднего плана.

Логические устройства: SYSRDR, SYSIPT, SYSLST, SYSLOG; SYS000 — инициализируемый пакет дисков.

Задание на выполнение

Оператор	Пояснения
// JOB	Начало задания
// ASSGN	Назначение периферийных устройств
// EXEC INTD	Выполнение программы
[Оператор модификации]	
Оператор оглавления тома	
Оператор(ы) метки тома	
// END	Оператор конца
/&	Конец задания

6.1.3. УПРАВЛЯЮЩИЕ ОПЕРАТОРЫ

Оператор модификации. Оператор предназначен для указания режима работы программы. Формат:

// UID Ix,Cn,R = (ccccchhh),...,R = (ccccchhh)

где // UID — признак и код оператора модификации (можно // U);

Ix — режим ввода (режим анализа поверхности: IA — выполняется анализ всех дорожек пакета дисков; IR — дорожки, отмеченные как дефектные, не анализируются; IS — анализ дорожек не выполняется;

Cn — счетчик анализа поверхности, указывающий, сколько раз нужно выполнять анализ поверхности каждой проверяемой дорожки: десятичное число 1 — 255;

R = (ccccchhh) — адрес дорожки, которую нужно отметить как дефектную без анализа поверхности: семизначное десятичное число (cccc — номер цилиндра, hhh — номер дорожки). Количество параметров не должно превышать общего количества запасных дорожек на пакете.

Значение по умолчанию: // U IR,C1

Оператор оглавления тома. Оператор предназначен для указания адреса и размера оглавления тома. Формат:

```
// VTOC STANDARD
// VTOC STRTADR = (ccccchhh),EXTENT = (n)
```

где // VTOC — признак и код оператора оглавления тома;

STANDARD — указание стандартного расположения оглавления тома, используемого обычно для рабочих пакетов (вслед за последней меткой тома до конца нулевого цилиндра);

STRTADR = (ccccchhh) — адрес дорожки, начиная с которой располагается оглавление тома;

EXTENT = (n) — количество дорожек, занимаемых оглавлением тома: десятичное число, не превышающее количества дорожек от заданного начального адреса оглавления тома до конца цилиндра.

Оператор(ы) метки тома. Каждый оператор предназначен для описания одной системной метки тома. Оператор VOL1 является обязательным, операторы VOL2 — VOL8 могут отсутствовать. Операторы должны следовать в порядке возрастания номеров меток. Формат:

Позиции	Содержимое
1—4	Идентификатор и номер метки (VOL1—VOL8)
5—10	Для обязательной метки — регистрационный номер тома, для дополнительных — любая информация
11—80	Любая информация

6.2. НАЗНАЧИТЬ ЗАПАСНУЮ ДОРОЖКУ

6.2.1. НАЗНАЧЕНИЕ И ВОЗМОЖНОСТИ

Программа позволяет без повторной инициализации использовать пакеты дисков, на которых в процессе эксплуатации появились дефектные дорожки. Программа обеспечивает следующие возможности:

- Выполняет для указанной пользователем основной дорожки диска либо функцию назначения запасной дорожки (с анализом или без анализа поверхности), либо функцию отмены назначения запасной дорожки.
- Выполняет функцию назначения и в том случае, когда основной дорожке уже была назначена запасная дорожка.
- При выполнении функции назначения копирует имеющиеся записи с исходной дорожки (основной или старой запасной) на вновь назначаемую запасную дорожку. Анализ поверхности (если он задан) выполняется только после перемещения записей. Если в результате анализа обнаружен дефект дорожки или анализ поверхности не задан, завершает назначение. В противном случае записи возвращаются обратно на исходную дорожку и назначение фактически не выполняется.
- При выполнении функции назначения корректирует ошибочные записи, если задан режим корректировки и поставляется пользователем корректирующая информация для каждой корректируемой записи.

- Во время копирования записей на запасную дорожку печатает в шестнадцатеричном виде содержимое либо всех перемещаемых записей, либо только тех, при чтении которых появились ошибки.
- При выполнении функции отмены назначения возвращает имеющиеся записи с освободившейся запасной дорожки на основную, не выполняя анализа поверхности основной дорожки. Освободившаяся запасная дорожка доступна для нового назначения только в том случае, если она была последней назначенной запасной дорожкой.

6.2.2. СПОСОБ ИСПОЛЬЗОВАНИЯ

Имя программы — ATAD.

Условия использования. Разделы фоновый и переднего плана (10K), требуется редактирование для разделов переднего плана.

Логические устройства: SYSRDR, SYSIPT, SYSLST, SYSLOG; SYS000 — обрабатываемый пакет дисков.

Задание на выполнение

Оператор	Пояснения
/// JOB	Начало задания
// ASSGN	Назначение периферийных устройств
// EXEC ATAD	Выполнение программы
Оператор модификации	
Оператор дорожки Корректирующая запись . . . . . . Оператор дорожки Корректирующая запись /*	Присутствуют только в том случае, если в операторе модификации задан режим корректировки (параметр UY)
&	Конец задания

6.2.3. УПРАВЛЯЮЩАЯ ИНФОРМАЦИЯ

Оператор модификации. Оператор предназначен для указания адреса дорожки, которой манипулирует программа, и режима работы программы. Формат

// UAT R = (ccccchhh),Ix,Ox,Cn,Ux

где // UAT — признак и код оператора модификации (можно // U);

R = (ccccchhh) — адрес основной дорожки, для которой нужно назначить запасную дорожку или отменить назначение запасной дорожки: семизначное десятичное число (cccc — номер цилиндра, hhh — номер дорожки). Параметр обязательный и должен быть первым;

Ix — режим ввода (режим манипуляции дорожкой): IA — назначается запасная дорожка с анализом поверхности исходной дорожки; IF — назначается запасная дорожка без анализа поверхности исходной дорожки; IU — отменяется назначение запасной дорожки;

Cn — счетчик анализа поверхности, указывающий, сколько раз нужно выполнять анализ поверхности дорожки: десятичное число 1 — 255;

Ux — режим корректировки записей: UY — корректировка выполняется; UN — корректировка не выполняется;

Ox — режим вывода (режим печати записей, перемещаемых на запасную дорожку) : OY — печатаются все записи; ON — печатаются только ошибочные записи.

Значение по умолчанию (необязательные параметры) : IF, OY, C1, UN.

Оператор дорожки. Каждый оператор предназначен для указания адреса дорожки, которой манипулирует программа (той же, что и в операторе модификации), и описания одной корректируемой записи. Вся числовая информация представляется в шестнадцатеричной системе счисления. Формат:

TRACK=ccccchhhhrrkkddddS

где TRACK= — идентификация оператора дорожки: позиции 1 — 6;
cccc — номер цилиндра: позиции 7 — 10;
hhhh — номер дорожки: позиции 11 — 14;
rr — номер корректируемой записи (00 — FF) : позиции 15 — 16;
kk — длина области ключа записи (00 — FF) : позиции 17 — 18;
dddd — длина области данных записи (0000 — FFFF) : позиции 19 — 22;
S — продолжение записи на следующую дорожку: позиция 23.

Корректирующая запись. Запись предназначена для указания информации, корректирующей заданную в операторе дорожки запись. Каждый байт записи представляется в виде двух шестнадцатеричных цифр и занимает 2 колонки перфокарты. Для представления каждой корректирующей записи используется необходимое количество перфокарт в зависимости от длины записи. Карты, за исключением последней, заполняются полностью.

6.3. ВЫВЕСТИ ОГЛАВЛЕНИЕ ТОМА

6.3.1. НАЗНАЧЕНИЕ И ФУНКЦИИ

Программа предназначена для вывода информации о содержимом пакета дисков и выполняет следующие функции:

- Распечатывает основные поля обязательной системной метки тома (метки VOL1) и меток файлов из оглавления тома
- Позволяет выводить оглавление тома для временного хранения на диски или на магнитную ленту в таком виде, что распечатка этой информации возможна с помощью программ перезаписи.

6.3.2. СПОСОБ ИСПОЛЬЗОВАНИЯ

Имя программы — LISTVTOC.

Условия использования. Разделы фоновый и переднего плана (10K), программа может редактироваться самоперемещающейся или несамоперемещающейся.

Логические устройства: SYSRDR, SYSLOG;
SYS004 — обрабатываемый пакет дисков;
SYS005 — вывод оглавления тома и сообщений программисту.

Задание на выполнение

Оператор	Пояснения
// JOB	Начало задания
[// DLBL // EXTENT]	Если оглавление тома выводится на диски, информация о метках и участках файла, предназначенного для размещения выводимого оглавления: UOUT — имя файла, SD — тип файла
[// TLBL]	Если оглавление тома выводится на магнитную ленту, информация для проверки меток файла: UOUT — имя файла
// ASSGN	Назначение периферийных устройств
[// LBLTYP TAPE]	Если программа самоперемещающаяся и задан оператор TLBL, резервирует область для метки
// EXEC LISTVTOC	Выполнение программы
/&	Конец задания

6.4. ОЧИСТИТЬ ДИСК

6.4.1. НАЗНАЧЕНИЕ И ФУНКЦИИ

Программа предназначена для предварительной форматизации однотомного или многотомного файла на дисках, обеспечивая возможность последующей записи информации в этот файл в произвольном порядке. Она выполняет следующие функции:

- Заполняет каждую дорожку форматизируемого файла записями фиксированной длины (с ключами либо без ключей), формат и содержимое которых указываются пользователем.
- Записывает в оглавление тома метку форматизированного файла.

6.4.2. СПОСОБ ИСПОЛЬЗОВАНИЯ

Имя программы — CLRDSK.

Условия использования. Разделы фоновый и переднего плана (10K), требуется редактирование для разделов переднего плана.

Логические устройства: SYSRDR, SYSIPT, SYSLST, SYSLOG; SYSnnn — форматизируемый файл (используемые устройства указываются в операторах EXTENT; по умолчанию — SYS005).

Задание на выполнение

Оператор	Пояснения
// JOB	Начало задания
// DLBL // EXTENT	Информация о метках и участках форматизируемого файла: UOUT — имя файла, SD — тип файла
// ASSGN	Назначение периферийных устройств
// EXEC CLRDSK	Выполнение программы
[Оператор модификации]	
// END	Оператор конца
/&	Конец задания

6.4.3. ОПЕРАТОР МОДИФИКАЦИИ

Оператор предназначен для указания формата и содержимого (символа заполнения) физических записей, записываемых на дорожки файла, и режима выполнения программы. Формат:

$$// \text{ UCL } B=(K=l,D=l), \left\{\begin{matrix} C'c' \\ X'xx' \end{matrix}\right\}, Ox, \ E=(e)$$

где // UCL — признак и код оператора модификации (можно // U);

B = (K = *l*,D = *l*) — формат физических записей, которые нужно записывать на дорожки форматизируемого файла: K = *l* — длина области ключа (не более 3 десятичных цифр); D = *l* — длина области данных (не более 4 десятичных цифр). Если ключ не нужен, K = 0;

C'c' — символ заполнения в символьном виде;

X'xx' — символ заполнения в шестнадцатеричном виде;

Ox — режим вывода (режим проверки записанной на диск информации): OY — проверка выполняется; ON — проверка не выполняется;

E = (e) — шифр накопителя на магнитных дисках. Для устройств ЕС-5052, ЕС-5055 и ЕС-5056 указывается обобщенный шифр 5052, для устройства ЕС-5061 — шифр 5061.

Значение по умолчанию:

```
// U  B=(K=0,D = 100),X'00',OY,E=(5052)
```

6.5. ПРОГРАММЫ КОПИРОВАНИЯ И ВОССТАНОВЛЕНИЯ

6.5.1. НАЗНАЧЕНИЕ И ВОЗМОЖНОСТИ

Программы предназначены для получения страховочных копий последовательных файлов, индексно-последовательных файлов, файлов прямого доступа и полных пакетов дисков (томов) на дисках, магнитных лентах и перфокартах; при этом информацию, скопированную на ленту и перфокарты, необходимо перед использованием восстановить на дисках. Программы обеспечивают следующие возможности:

- Обрабатывают многотомные файлы, расположенные на разделенных или неразделенных цилиндрах.
- Получают на дисках (при непосредственном копировании или в результате восстановления) информацию, идентичную исходной не только по содержанию, но и по расположению.
- Прежде чем записывать информацию на диски, проверяют, чтобы на заданных пользователем участках не было файлов с неистекшим сроком хранения—проверяемые участки выводного пакета необходимо задавать и при копировании-восстановлении тома.
- В процессе работы могут создавать контрольные точки на дисках или на магнитной ленте, что позволяет в случае ошибки повторить выполнение программы не с начала, а с контрольной точки (с помощью оператора Управления заданиями RESTART).

- При копировании на магнитную ленту выполняют перемотку ленты до начала копирования и перемотку с разгрузкой после копирования.
- Позволяют идентифицировать скопированную на магнитную ленту информацию и файл контрольных точек на магнитной ленте с помощью системных меток.
- Позволяют копировать на диски и магнитную ленту метки пользователя и восстанавливать их с магнитной ленты.
- Позволяют копировать вместе с файлом две записи программы Начальная загрузка, что необходимо при копировании резидентных файлов.

6.5.2. СПОСОБ ИСПОЛЬЗОВАНИЯ

Имена программ. См. табл. 6.1.

Условия использования. Разделы фоновый и переднего плана (10K), требуется редактирование для разделов переднего плана.

Логические устройства: SYSRDR, SYSIPT (при копировании), SYSLST, SYSLOG;

SYS003 — создание контрольных точек, если назначены диски или магнитная лента (в противном случае контрольные точки не создаются);

SYS004 — ввод копируемого тома с дисков, а также ввод восстанавливаемого тома и файла с магнитной ленты и перфокарт;

SYS005 — вывод копируемой информации на магнитную ленту;

SYS006 — вывод копируемой информации на перфокарты;

SYSnnn — ввод копируемого файла с дисков, а также вывод копируемой и восстанавливаемой информации на диски (используемые устройства указываются в операторах EXTENT, по умолчанию — SYS004 для ввода, SYS005 для вывода).

Задание на выполнение

Оператор	Пояснения
// JOB	Начало задания
// DLBL // EXTENT	Информация о метках и участках файлов на дисках или о проверяемых участках выводного пакета при копировании-восстановлении тома: UIN — имя вводного файла, UOUT — имя выводного файла или пакета, UCHKPT — имя файла контрольных точек (см. табл. 6.1)
// TLBL	Информация о метках файлов на магнитных лентах: UIN — имя вводного файла, UOUT — имя выводного файла, UCHKPT — имя файла контрольных точек (см. табл. 6.1)
// ASSGN	Назначение периферийных устройств. Отмена назначения для SYS003, если не создаются контрольные точки
// MTC	Если предполагается копирование информации на чистую ленту, запись ленточной марки. Если контрольные точки на магнитной ленте, установка ленты в нужное положение
// UPSI	Режим обработки меток для файлов на магнитных лентах и дисках (см. 6.5.4)
// EXEC имя	Выполнение программы (имя в табл. 6.1)
Оператор модификации (только для программ копирования)	
/&	Конец задания

Таблица 6.1

Использование операторов DLBL, EXTENT и TLBL

Программа	Имя	DLBL, EXTENT*	TLBL*
Копировать диск на диск	CRDD	Файл — UIN, UOUT Том — UOUT	
Копировать диск на ленту	CRDT	Файл — UIN	UOUT **
Копировать диск на карты	CRDC	Файл — UIN	
Восстановить диск с ленты	CRTD	Файл, том — UOUT	UIN **
Восстановить диск с карт	CRCD	Файл, том — UOUT	

* В случае создания контрольных точек для любой программы дополнительно указываются: DLBL (UCHKPT), EXTENT, если точки на дисках; TLBL (UCHKPT), если точки на магнитной ленте с системными метками.

** Указываются, если лента с системными метками.

6.5.3. ОПЕРАТОР МОДИФИКАЦИИ

Оператор предназначен для указания режима работы программы копирования. Формат:

// UCR Tt,A=(a),IPL,Mx,Ox,E = (e)

где // UCR — признак и код оператора модификации;

Tt — функция (обязательный параметр): TF — копирование файла, TV — копирование тома;

A = (a) — максимальная длина блока копируемого файла или тома: десятичное число, не содержащее незначащих нулей и не превышающее емкости дорожки диска (параметр обязательный);

IPL — копирование записей Начальной загрузки при копировании файла (при копировании тома недействителен);

Mx — организация копируемого файла: MS — последовательный файл, MD — файл прямого доступа, MI — индексно-последовательный файл;

Ox — количество областей ввода-вывода для программы Копировать диск на ленту: O1 — одна область ввода-вывода, O2 — две области ввода-вывода;

E = (e) — шифр накопителя на магнитных дисках; для устройств EC-5052, EC-5055 и EC-5056 указывается обобщенный шифр 5052, для устройства EC-5061 — шифр 5061.

Значение по умолчанию (необязательные параметры): MS, O2, E = (5052).

6.5.4. ОБРАБОТКА МЕТОК

Обработка меток магнитной ленты. Скопированный на магнитную ленту файл или том, а также файл контрольных точек представляют собой файлы с системными метками либо файлы без меток. Информация для обработки меток указывается с помощью оператора TLBL, режим обработки меток — с помощью оператора UPSI.

Бит 0 — обработка меток вводного файла (программа Восстановить диск с ленты): 1 — выполняется проверка меток; 0 — отсутствует проверка меток.

Бит 2 — обработка меток выводного файла (программа Копировать диск на ленту) и файла контрольных точек: 1 — метки создаются, 0 — метки не создаются.

Бит 4 — запись ленточной марки в начале выводного файла и файла контрольных точек, не имеющих меток: 0 — марка записывается; 1 — марка не записывается.

Обработка меток пользователя. При копировании файла на диски и магнитную ленту можно копировать и метки пользователя, при восстановлении файла с ленты скопированные метки пользователя должны восстанавливаться. Для указания режима обработки меток используется оператор UPSI.

Бит 7 — обработка меток пользователя: 0 — метки не копируются и не восстанавливаются; 1 — метки копируются и восстанавливаются.

6.6. ИНИЦИАЛИЗИРОВАТЬ ЛЕНТУ

6.6.1. НАЗНАЧЕНИЕ И ФУНКЦИИ

Программа предназначена для подготовки катушек магнитных лент, которые планируется использовать для записи файлов с системными метками. Она выполняет следующие функции:

- Не проверяя содержимого инициализируемой ленты, записывает на ленту обязательную системную метку тома, которая используется для идентификации ленты, до 7 дополнительных меток тома, если пользователь поставляет для них информацию, фиктивную начальную метку файла и ленточную марку.
- За одно выполнение программы может инициализировать любое количество магнитных лент, причем позволяет устанавливать одновременно до 16 лент, используя для них отдельные логические устройства (SYS000 — SYS015).

6.6.2. СПОСОБ ИСПОЛЬЗОВАНИЯ

Имя программы — INTT.

Условия использования. Разделы фоновый и переднего плана (10K), требуется редактирование для разделов переднего плана.

Логические устройства: SYSRDR, SYSIPT, SYSLOG;
SYS000 — инициализируемая лента;
SYS001 — SYS015 — дополнительные инициализируемые ленты

Задание на выполнение

Оператор	Пояснения
// JOB	Начало задания
// ASSGN	Назначение периферийных устройств: выполняются в возрастающей последовательности начиная с SYS000. Отмена назначения для первого неиспользуемого устройства SYS001—SYS015
// EXEC INTT	Выполнение программы

```
Оператор модификации
┌Оператор(ы) метки тома      ┐
[// END]                     │  Присутствуют только в том случае, если
. . . . . .                  │  в операторе модификации не указывается
Оператор(ы) метки тома       │  информация для создания метки VOL1 (задан
└[// END]                    ┘  параметр CARD)
/&                  Конец задания
```

6.6.3. УПРАВЛЯЮЩИЕ ОПЕРАТОРЫ

Оператор модификации. Оператор предназначен для указания режима работы программы и — необязательно — для поставки информации, необходимой для создания обязательной системной метки тома (метки VOL1). Формат:

```
// INTT  CARD,REWIND,SERIAL = (xxxxxx),P = (x),
                CODE = (xxxxxxxxxx)
```

где // INTT — признак и код оператора модификации;

CARD — указывает на наличие в задании операторов метки тома для создания меток тома. При отсутствии параметра информация для создания обязательной метки тома содержится в операторе модификации, а дополнительные метки не создаются;

REWIND — перемотка ленты после инициализации. При отсутствии параметра лента перематывается и разгружается;

SERIAL = (xxxxxx) — регистрационный номер тома первой инициализируемой ленты (поле метки тома) — шестизначное десятичное число. Для дополнительных лент это число увеличивается на 1. Параметр обязательный, если отсутствует параметр CARD;

P = (x) — код защиты тома (поле метки тома). Параметр может указываться, если отсутствует параметр CARD. По умолчанию P = (0);

CODE = (xxxxxxxxxx) — идентификация владельца тома (поле метки тома) — 10 алфавитно-цифровых символов. Параметр обязательный, если отсутствует параметр CARD.

Оператор(ы) метки тома. Каждый оператор предназначен для описания одной системной метки тома. Количество операторов метки тома для каждой инициализируемой ленты определяется количеством создаваемых на ленте меток тома и располагаться они должны в порядке возрастания номеров меток: VOL1 — VOL8. Если для ленты указывается меньше 8 операторов метки тома, вслед за ними должен следовать оператор // END. Формат операторов метки тома должен совпадать с форматом системных меток тома (см. 6.1.3).

6.7. СРАВНИТЬ ЛЕНТЫ

6.7.1. НАЗНАЧЕНИЕ И ВОЗМОЖНОСТИ

Программа предназначена для сравнения любых двух файлов, расположенных на катушках магнитной ленты, чтобы установить их идентичность. Обеспечивает следующие возможности:

- Сравнивает файлы любой организации без меток, позволяя рассматривать метки как данные: как отдельные файлы, если метки отделены от файла ленточной маркой, или как часть файла в противном случае.
- Сравнивает файлы с системными метками, проверяя идентификаторы обрабатываемых файлов, указанные в начальных и конечных метках.
- Сравнивает многотомные файлы, разрешая при этом использовать сменные устройства.
- Перед началом сравнения выполняет подвод магнитной ленты к началу файла с системными метками. Для файлов без меток начинает сравнение с той зоны ленты, перед которой установлены головки чтения-записи.
- Пропускает имеющиеся записи контрольных точек.
- Позволяет за одно выполнение программы сравнивать любое количество пар файлов.
- Позволяет оператору исправлять ошибочные операторы модификации.
- Предусматривает возможность оперативной связи оператора с программой.
- Позволяет включить в состав программы подпрограмму пользователя для обработки несовпадающих блоков.

6.7.2. СПОСОБ ИСПОЛЬЗОВАНИЯ

Имя программы — TPCP.

Условия использования. Разделы фоновый и переднего плана (10K), требуется редактирование для разделов переднего плана.

Логические устройства: SYSRDR, SYSIPT, SYSLST, SYSLOG; SYS004 — ввод одного из сравниваемых файлов (файла A); SYS005 — ввод другого из сравниваемых файлов (файла B).

Задание на выполнение

Оператор	Пояснения
// JOB	Начало задания
// ASSGN	Назначение периферийных устройств
// MTC	Если требуется, подвод магнитных лент к началу сравниваемых файлов
// EXEC TPCP	Выполнение программы
Оператор(ы) модификации	
/*	
/&	Конец задания

6.7.3. ОПЕРАТОР МОДИФИКАЦИИ

Оператор предназначен для описания сравниваемых файлов и указания режима работы программы. Формат:

```
// TPCP RECSIZ = (m),REELS = (n),LABELS,ALTA,ALTB,EXIT
```

где // TPCP — признак и код оператора модификации;

RECSIZ = (m) — максимальная длина блока сравниваемых файлов — десятичное число не более 5 цифр без незначащих нулей. Параметр обязательный и должен быть первым;

REELS = (n) — максимальное количество катушек, занимаемых одним файлом, — десятичное число 1 — 255. При отсутствии параметра n = 1;

LABELS — необходимость проверки идентификаторов файлов из системных меток сравниваемых файлов. При отсутствии параметра проверка не выполняется;

ALTA — наличие сменного устройства для файла A. При отсутствии параметра сменное устройство не используется;

ALTB — наличие сменного устройства для файла B. При отсутствии параметра сменное устройство не используется;

EXIT — наличие подпрограммы пользователя для обработки несовпадающих блоков. При отсутствии параметра выполняется только печать несовпадающих блоков с указанием причины несравнения (несовпадающие символы, фактическая длина блока превышает заданную, неодинаковая длина сравниваемых блоков).

6.7.4. ПОДПРОГРАММА ПОЛЬЗОВАТЕЛЯ ДЛЯ ОБРАБОТКИ НЕСОВПАДАЮЩИХ БЛОКОВ

Подпрограмма предназначена для любой обработки несовпадающих блоков. Точка входа — IJWXIT1. Подпрограмма не должна использовать регистр 4. Получает информацию: регистр 0 — адрес поля, содержащего для файла A адрес области ввода (4 байта) и длину блока (4 байта); регистр 1 — адрес поля, содержащего для файла B адрес области ввода (4 байта) и длину блока (4 байта); регистр 10 — номер несовпадающих блоков. Возвращает управление по адресу в регистре 14.

Включается в программу Сравнить ленты на этапе редактирования. Задание на редактирование и каталогизацию в системную библиотеку абсолютных модулей (используются стандартные назначения устройств, объектный модуль с подпрограммой пользователя на перфокартах):

```
// JOB TPCPUSR
// OPTION CATAL
   [ACTION F1 (или F2)]   Для разделов переднего плана
   PHASE TPCP,*,NOAUTO
   INCLUDE IJWTCP
   INCLUDE IJJCPD0
   INCLUDE
Объектный модуль
/*
   INCLUDE IJWTPCP
   ENTRY
// EXEC LNKEDT
/&
```

ГЛАВА 7

ПЕРЕЗАПИСЬ И СОРТИРОВКА ИНФОРМАЦИИ

Для осуществления перезаписи информации между различными носителями данных можно воспользоваться программами перезаписи (см. разд. 7.1 и 7.2) и программой Сервис (см. разд. 7.3); дублирование информации на дисках можно выполнить также с помощью программы Копировать диск на диск (см. гл. 6). Можно составить свои программы, используя при программировании на языке Ассемблера макрокоманды перезаписи (см. разд. 7.4) или макрокоманды логической системы управления вводом-выводом (см. гл. 3).

Сортировку файлов можно выполнить одной из двух программ (см. разд. 7.5 и 7.6): программой сортировки на магнитных лентах (TSRT) или программой сортировки на магнитных лентах и дисках (SORT).

Эти программы отличаются набором обеспечиваемых ими возможностей и временны́ми характеристиками. Возможности программы TSRT несколько у́же, чем программы SORT. Так, программа TSRT позволяет сортировать данные, расположенные на магнитных лентах, в то время как программа SORT — и на магнитных лентах, и на дисках. Программу TSRT нельзя использовать как подпрограмму некоторой другой программы, в то время как программа SORT обеспечивает и такой вариант. Имеются и другие менее значительные отличия. Однако в некоторых случаях программа TSRT по сравнению с программой SORT обеспечивает меньшее время выполнения при равных условиях использования [5].

7.1. ПРОГРАММЫ ПЕРЕЗАПИСИ

7.1.1. НАЗНАЧЕНИЕ И ВОЗМОЖНОСТИ

Программы предназначены для перезаписи с одних носителей данных на другие последовательно организованных файлов, состоящих из записей фиксированной, переменной или неопределенной длины. Они обеспечивают следующие возможности:

- За одно выполнение каждой программы перезаписывают один файл.
- Обрабатывают файлы с системными метками, с системными метками и метками пользователя (диски и магнитная лента), а также файлы с нестандартными метками и файлы без меток (магнитная лента).

- Обрабатывают многотомные файлы на дисках и магнитных лентах. Файлы на дисках могут быть на разделенных или неразделенных цилиндрах, для файлов на магнитных лентах разрешается использовать сменные устройства.
- Дают возможность пользователю задавать режим установки магнитной ленты, что позволяет обрабатывать любые файлы на многофайловых томах.
- Пропускают имеющиеся записи контрольных точек на магнитных лентах.
- При выводе на диски могут выполнять проверку записанной информации с помощью контрольного чтения.
- Осуществляют ввод и вывод карт в коде ДКОИ — карты в коде КПК-12, автоматическое преобразование в код или из кода ДКОИ, а также двоичный ввод и вывод карт, что позволяет вводить и выводить карты в любом коде.
- Могут проверять последовательность ввода карт по возрастанию заданного поля нумерации и генерировать поле нумерации на каждой карте выводного файла.
- Выводят карты в указанный карман или обеспечивают управление выбором кармана по первому символу (тип D). Перфорируют управляющий символ выбора кармана, если он не исключается в результате перекомпоновки.
- Печатают информацию в каждой строке, через одну строку, через две строки или обеспечивают управление форматом печати по первому символу (типы A, C, D). Выводят управляющий символ печати, если он не исключается в результате перекомпоновки.
- Печатают информацию в символьном или шестнадцатеричном виде.
- Печатают файлы в формате отображения, обеспечивающем визуальную картину файла, или в формате списка, позволяющем получить список логических записей файла.
- Могут печатать строку заглавия на каждой странице файла.
- В процессе перезаписи могут выполнять некоторую обработку файла, которая определяется типами носителей и форматом записей: переблокирование записей, перекомпоновку, упаковку или распаковку заданных полей логических записей.
- Позволяют пропустить в начале файла вплоть до 99 999 логических записей.
- Позволяют включить в состав программы подпрограмму пользователя для обработки меток пользователя (стандартных или нестандартных).

7.1.2. СПОСОБ ИСПОЛЬЗОВАНИЯ

Имена программ. См. табл. 7.1.

Условия использования. Разделы фоновый и переднего плана (10К), требуется редактирование для разделов переднего плана.

Логические устройства: SYSRDR, SYSIPT, SYSLST, SYSLOG;
SYS004 — ввод файла с перфокарт и магнитной ленты;
SYS005 — вывод файла на печать и магнитную ленту;

Таблица 7.1

Имена, инициалы, форматы записей и функции программ перезаписи

Программа			Функции			
Название	Имя	Инициалы	Записи неопределенной длины	Записи фиксированной длины без ключей	Записи фиксированной длины с ключами	Записи переменной длины
КАРТЫ-ПЕЧАТЬ/ КАРТЫ	CDPP	CP	—	TC, TF, TD, TL, TLF, TB, TBF	—	—
КАРТЫ-ДИСК	CDDK	CD	—	TC, TF, TR, TRF	TF	—
КАРТЫ-ЛЕНТА	CDTP	CT	—	TC, TF, TR, TRF	—	—
ДИСК-КАРТЫ	DKCD	DC	—	TC, TF, TR, TRF	TF	—
ДИСК-ДИСК	DKDK	DD	TC	TC, TF, TR, TRF	TC, TF — оба файла TF, TRF — один файл	TC, TF, TR, TRF
ДИСК-ПЕЧАТЬ	DKPR	DP	TD	TD *, TL, TLF	TD *, TL, TLF	TD *, TL, TLF
ДИСК-ЛЕНТА	DKTP	DT	TC	TC, TF, TR, TRF	TF, TRF	TC, TF, TR, TRF
ЛЕНТА-КАРТЫ	TPCD	TC	—	TC, TF, TR, TRF	—	—
ЛЕНТА-ДИСК	TPDK	TD	TC	TC, TF, TR, TRF	TF, TRF	TC, TF, TR, TRF
ЛЕНТА-ПЕЧАТЬ	TPPR	TP	TD	TD *, TL, TLF	—	TD *, TL, TLF
ЛЕНТА-ЛЕНТА	TPTP	TT	TC	TC, TF, TR, TRF	—	TC, TF, TR, TRF

SYS006 — вывод файла на перфокарты;
SYSnnn — ввод файла с дисков и вывод файла на диски (используемые устройства указываются в операторах EXTENT; по умолчанию — SYS004 для ввода, SYS005 для вывода).

Задание на выполнение

Оператор	Пояснения
// JOB	Начало задания
// DLBL // EXTENT	Если вводной и (или) выводной файл на дисках, информация о метках и участках файлов: UIN — имя вводного файла, UOUT — имя выводного файла, SD — тип файла
// TLBL	Если вводной и (или) выводной файл на магнитной ленте со стандартными метками, информация о метках файлов: UIN — имя вводного файла, UOUT — имя выводного файла
// ASSGN	Назначение периферийных устройств
// MTC	Если вводной и (или) выводной файл на магнитной ленте, при необходимости установка ленты в нужное положение, а также запись ленточной марки на чистую ленту
// UPSI	Для файлов на магнитной ленте и дисках режим обработки меток (см. 7.1.4)
// EXEC имя [Оператор модификации] [Оператор(ы) перекомпоновки] [Оператор(ы) заглавия]	Выполнение программы (имя в табл. 7.1)
// END	Оператор конца
Вводной файл /*	При вводе файлов с перфокарт (SYS004 = SYSRDR = = SYSIPT)
/&	Конец задания

7.1.3. УПРАВЛЯЮЩИЕ ОПЕРАТОРЫ

Оператор модификации. Оператор предназначен для описания вводного и выводного файлов и указания режима работы программы. Формат:

// Uxx Tt,Ff,A=(ввод),B=(вывод),Ix,Ox,Px,Rx,Sx,Q=(x,y),E=(шифр)

где // Uxx — признак и код оператора модификации. Инициалы программы xx (табл. 7.1) могут отсутствовать;

Tt — функция, выполняемая программой: TC — копирование, TF — перекомпоновка, TR — переблокирование, TRF — переблокирование с перекомпоновкой, TD—отображение, TL — список, TLF—список с перекомпоновкой, TB — печать и перфорация, TBF — печать и перфорация с перекомпоновкой. Функции, выполняемые каждой программой перезаписи в зависимости от формата записей файлов, перечислены в табл. 7.1;

Ff — формат записей вводного и выводного файлов: FF — записи фиксированной длины, FV — записи переменной длины, FU — записи неопределенной длины. Наличие ключей для файлов на дисках указывается в параметрах A и B. Форматы записей, допустимые для каждой программы перезаписи, приведены в табл. 7.1;

* Если задано управление форматом печати, функция TD недействительна.

A = (ввод) — описание вводного файла:

A = (n,m) — для записей фиксированной длины без ключей (n — длина записи, m — длина блока) и записей переменной длины с перекомпоновкой (n — длина фиксированной части записи, m — максимальная длина блока). Для неблокированных записей фиксированной длины значения n и m должны быть одинаковыми, для блокированных записей m должно быть кратно n.

A = (K = *l*,D = *l*) — для записей фиксированной длины с ключами (K = *l* — длина области ключа, D = *l* — длина области данных).

A = (g) — для записей неопределенной длины и записей переменной длины без перекомпоновки (g — максимальная длина блока). Для записей неопределенной длины с ключами указывается суммарная длина областей ключа и данных;

B = (вывод) — описание выводного файла:

B = (n,m) — аналогично параметру A = (n,m);

B = (K = *l*), — аналогично параметру A = (K = *l*,D = *l*);

B = (g) — аналогично параметру A = (g);

B = (p) — при выводе на печать записей фиксированной длины, переменной длины без перекомпоновки и неопределенной длины (p — длина печатной строки). Для формата отображения устанавливается p = 120.

B = (n,p) — при выводе на печать записей переменной длины с перекомпоновкой (n — длина фиксированной части выводной записи, p — длина печатной строки);

Ix — режим ввода:

Для программ CDPP, CDDK, CDTP: I1 — ввод перфокарт в коде ДКОИ; I2 — двоичный ввод перфокарт.

Для программ TPCD, TPDK, TPPR, TPTP: IR — перемотка вводной ленты как до, так и после перезаписи файла, IN — нет перемотки вводной ленты ни до, ни после перезаписи файла, IU — перемотка вводной ленты до перезаписи и перемотка с разгрузкой после перезаписи файла;

Ox — режим вывода:

Для программ CDPP, DKCD, TPCD: O1 — вывод перфокарт в коде ДКОИ, O2 — двоичный вывод перфокарт.

Для программ CDDK, DKDK, TPDK: OY — выполняется проверка записанной на диск информации, ON — отсутствует проверка записанной на диск информации.

Для программ CDPP, DKPR, TPPR (формат отображения): OC — символьный вывод на печать, Ox — шестнадцатеричный вывод на печать. В формате списка обычно символьный вывод, а шестнадцатеричный вывод полей записи можно задать в операторе перекомпоновки.

Для программ CDTP, DKTP, TPTP: OR — перемотка выводной ленты как до, так и после перезаписи файла, ON — нет перемотки выводной ленты ни до, ни после перезаписи файла, OU — перемотка выводной ленты до перезаписи и перемотка с разгрузкой после перезаписи файла;

Px — нумерация страниц в формате списка (для программ CDPP, DKPR, TPPR): PY — страницы нумеруются, PN — страницы не нумеруются. В формате отображения страницы нумеруются всегда;

Rx — номер логической записи вводного файла, начиная с которой выполняется перезапись (будет пропущено x — 1 записей). Для функций TC и TB всегда устанавливается R1;

Sx — управление выводом. При выводе на перфокарты (для программ CDPP, DKCD, TPCD): S1 — карман 1, S2 — карман 2, S3 — управление выбором кармана по первому символу.

При выводе на печать в формате списка (для программ CDPP, DKPR, TPPR): S1 — пропуск 1 строки после печати (печать в каждой строке), S2 — пропуск 2 строк после печати (печать через 1 строку), S3 — пропуск 3 строк после печати (печать через 2 строки), SA — управление форматом печати по первому символу (тип A), SC — управление форматом печати по первому символу (тип C), SD — управление форматом печати по первому символу (тип D). В формате отображения информация печатается через 1 строку;

Q = (x,y) — генерация или проверка поля нумерации карт (для программ CDPP, CDDK, CDTP, DKCD, TPCD): x — номер первой колонки поля нумерации карты, y—длина поля нумерации карты (не более 10 колонок);

E = (шифр) — шифр накопителя на магнитных дисках (для программ CDDK, DKCD, DKDK, DKPR, DKTP, TPDK): E = (e) — один файл на дисках или оба файла на устройствах одного типа (e — шифр устройства) E=(c,d) — оба файла на дисках (c — шифр устройства для вводного файла, d — шифр устройства для выводного файла). Для устройств EC-5052, EC-5055 и EC-5056 указывается обобщенный шифр 5052, для устройства EC-5061 — шифр 5061.

Значение по умолчанию: см. табл. 7.2, с. 172.

Взаимосвязь параметров оператора модификации. Форма параметров описания файлов определяется форматом записей. Возможны следующие комбинации параметров F, A, B:

FF, A = (n,m), B = (n, m)	FV, A = (n, m), B = (n, m)
FF, A = (n, m), B = (K = *l*, D = *l*)	FV, A = (n, m), B = (n, p)
FF, A = (n, m), B = (p)	FV, A = (g), B = (g)
FF, A = (K = *l*, D = *l*), B = (n, m)	FV, A = (g), B = (p)
FF, A = (K = *l*, D = *l*), B = (K = *l*, D = *l*)	FU, A = (g), B = (g)
FF, A = (K = *l*, D = *l*), B = (p)	FU, A = (g), B = (p)

Оператор перекомпоновки. Если в операторе модификации задан режим перекомпоновки (функция TF, TRF, TLF или TBF), вслед за ним следуют операторы перекомпоновки. Каждый оператор предназначен для указания информации, необходимой для пересылки полей логических записей и их преобразования. Формат:

// FS параметр/параметр/ . . ./параметр

Параметр перекомпоновки зависит от способа пересылки поля — с преобразованием или без преобразования — и его местоположения — ключ или данные.

Таблица 7.2

Стандартные значения параметров оператора модификации

Программа перезаписи	Обязательные параметры				Необязательные параметры						
	Tf	Ff	A=(ввод)	B=(вывод)	Ix	Ox	Px	Rx	Sx	Q=(x, y)	E=(шифр)
CDPP { карты / печать	TB	FF	A=(80, 80)	B=(80, 80) B=(80)	II	OI OC	— PY	R1	S2	Поле нумерации не проверяется, не генерируется	—
CDDK	TC	FF	A=(80, 80)	B=(80, 80)	II	OY	—	R1	—	Поле нумерации не проверяется	E=(5052)
CDTP	TC	FF	A=(80, 80)	B=(80, 80)	II	OU	—	R1	—	Поле нумерации не проверяется	—
DKCD	TC	FF	A=(80, 80)	B=(80, 80)	—	OI	—	R1	S2	Поле нумерации не генерируется	E=(5052)
DKDK	TC	FU	A=(1000)	B=(1000)	—	OY	—	R1	—	—	E=(5052)
DKPR	TD	FU	A=(1000)	B=(120)	—	OX	PY	R1	S1	—	E=(5052)
DKTP	TC	FU	A=(1000)	B=(1000)	—	OU	—	R1	—	—	E=(5052)
TPCD	TC	FF	A=(80, 80)	B=(80, 80)	IU	OI	—	R1	S2	Поле нумерации не генерируется	—
TPDK	TC	FU	A=(1000)	B=(1000)	IU	OY	—	R1	—	—	E=(5052)
TPPR	TD	FU	A=(1000)	B=(120)	IU	OX	PY	R1	S1	—	—
TPTP	TC	FU	A=(1000)	B=(1000)	IU	OU	—	R1	—	—	—

Пересылка поля без изменения:

r,s,t	r, s, (K,t)
(K,r),s,t	(K,r),s,(K,t)

Упаковка поля (если файлы не печатаются):

r,(P,n,m),t	r,(P,n,m),(K,t)
(K,r),(P,n,m),t	(K,r),(P,n,m),(K,t)

Распаковка поля:

r,(U,n,m),t	r,(U,n,m),(K,t)
(K,r),(U,n,m),t	(K,r),(U,n,m),(K,t)

Преобразование поля в шестнадцатеричный вид (при печати файлов):

r,(X,n),t
(K,r),(X,n),t

Используемые обозначения:

r — начальная позиция входного поля относительно первой позиции вводной записи;
t — начальная позиция выходного поля относительно первой позиции выводной записи;
s — длина поля (входного и выходного);
n — длина входного поля;
m — длина выходного поля;
K — признак того, что начальная позиция поля отсчитывается относительно первой позиции области ключа;
P — признак упаковки;
U — признак распаковки;
X — признак преобразования в шестнадцатеричный вид.

Если для записей переменной длины нужно копировать переменную часть записи, в качестве одного из параметров оператора перекомпоновки нужно задать буквы CV.

Операторы заглавия. Для программ CDPP, DKPR, TPPR операторы предназначены для указания заглавия, которое будет печататься на каждой странице файла. Существуют два оператора заглавия, причем второй указывается тогда, когда весь текст заглавия не умещается в первом. Формат:

```
// H1 текст заглавия
// H2 продолжение текста заглавия
```

В первом операторе заглавия текст можно указывать в позициях 7 — 80, во втором — в позициях 7 — 76. Максимальная длина строки заглавия — 144 символа, но если длина печатной строки конкретного устройства меньше 144 символов, остаток теряется без сообщения об ошибке.

7.1.4. ОБРАБОТКА МЕТОК

Обработка меток. Режим обработки меток указывается с помощью оператора Управления заданиями UPSI.

Биты 0, 1 — обработка меток вводного файла.

Биты 2, 3 — обработка меток выводного файла.

Для обоих файлов: 00 — обрабатываются системные метки; 01 — обрабатываются системные метки и стандартные метки пользователя; 10 — отсутствует обработка меток; 11 — обрабатываются нестандартные метки.

Бит 4 — запись ленточной марки в начале выводного файла без меток: 0 — марка записывается; 1 — марка не записывается.

Подпрограмма пользователя для обработки меток. Подпрограмма предназначена для обработки меток пользователя (стандартных или нестандартных). Она должна содержать две или три подпрограммы:

подпрограмму инициализации, которая получает управление при загрузке подпрограммы пользователя в основную память и может выполнять любые подготовительные действия для подпрограмм(ы) обработки меток (должна содержать по крайней мере команду возврата в программу перезаписи — BR14); точка входа — в карте END объектного модуля;

подпрограмму обработки меток вводного файла, которая получает управление во время открытия и закрытия вводного файла (см. 3.1.5); точка входа — IJWLABIN;

подпрограмму обработки меток выводного файла, которая получает управление во время открытия и закрытия выводного файла (см. 3.1.5); точка входа — IJWLABOU.

Должно быть определено имя IJWLABND — адрес последнего байта подпрограммы пользователя, увеличенный на 1. Если отсутствует подпрограмма обработки меток вводного или выводного файла, то нужно задавать соответственно

```
IJWLABIN EQU IJWLABND
```

или

```
IJWLABOU EQU IJWLABND
```

Включается в программу перезаписи на этапе редактирования. Задание на редактирование и каталогизацию в системную библиотеку абсолютных модулей (используются стандартные назначения устройств; объектный модуль с подпрограммой пользователя, например IJWUSER — в системной библиотеке объектных модулей):

```
// JOB UTILITY
// OPTION CATAL
   [ACTION F1 (или F2)] Для разделов переднего плана
   INCLUDE IJWxx
   PHASE xxxx5,IJWxxCS2,NOAUTO
   INCLUDE IJWUSER
   ENTRY
[// LBLTYP TAPE]    Для программ, обрабатывающих файлы на
                    магнитных лентах
// EXEC LNKEDT
/&
```

Примечание. xx — инициалы программы, xxxx — имя программы (имена и инициалы приведены в табл. 7.1).

7.2. ПРОГРАММЫ ПЕРЕЗАПИСИ ДЛЯ ПЕРФОЛЕНТЫ

7.2.1. НАЗНАЧЕНИЕ И ВОЗМОЖНОСТИ

Программы предназначены для перезаписи файлов с перфоленты на магнитную ленту и диски. Они обеспечивают следующие возможности:

- За одно выполнение каждой программы перезаписывают один файл.
- Обрабатывают файлы с системными метками и метками пользователя (диски и магнитная лента), а также файлы с нестандартными метками и файлы без меток (магнитная лента). Могут проверять метки на перфоленте.
- Обрабатывают многотомные файлы на дисках и магнитных лентах. Файлы на дисках могут быть на разделенных или неразделенных цилиндрах; для файлов на магнитных лентах разрешается использовать сменные устройства.
- Дают возможность пользователю задавать режим установки магнитной ленты, что позволяет обрабатывать любые файлы на многофайловых томах.
- При выводе на диски могут выполнять проверку записанной информации с помощью контрольного чтения.
- Позволяют обрабатывать перфоленты в любом одно- или двухрегистровом коде. Для перфолент в коде ISO (нулевой набор кода КОИ-7) выполняют перекодировку в код ДКОИ. Для перекодировки информации из других кодов позволяют включать на этапе редактирования в состав программы соответствующие таблицы перекодировки.
- Позволяют пропустить в начале файла вплоть до 99 999 логических записей.
- В процессе перезаписи выполняют перекомпоновку логических записей, в результате которой можно переставлять поля, упаковывать их, удалять, расширять, добавлять новые поля.
- Позволяют включить в состав программы подпрограмму пользователя для обработки меток пользователя (стандартных или нестандартных).

7.2.2. СПОСОБ ИСПОЛЬЗОВАНИЯ

Имена программ: Перфолента-лента — PTTP, Перфолента-диск — PTDK.

Условия использования. Разделы фоновый и переднего плана (10K), требуется редактирование для разделов переднего плана.

Логические устройства: SYSRDR, SYSIPT, SYSLST, SYSLOG;
SYS004 — ввод файла с перфоленты;
SYS005 — вывод файла на магнитную ленту;
SYSnnn — вывод файла на диски (используемые устройства указываются в операторах EXTENT, по умолчанию— SYS005).

Задание на выполнение

Оператор	Пояснения
// JOB	Начало задания
// DLBL	Если выводной файл на дисках, информация о метках
// EXTENT	и участках файла: UOUT — имя файла, SD — тип файла
// TLBL	Если выводной файл на магнитной ленте со стандартными метками, информация о метках файла: UOUT — имя файла
// ASSGN	Назначение периферийных устройств
// MTC	Если выводной файл на магнитной ленте, при необходимости установка ленты в нужное положение, а также запись ленточной марки на чистую ленту
// UPSI	Для файлов на магнитной ленте и дисках режим обработки меток (см. 7.2.4)
// EXEC имя	Выполнение программы (имя — PTTP или PTDK)
Оператор модификации	
Оператор(ы) перекомпоновки	
// END	Оператор конца
/&	Конец задания

7.2.3. УПРАВЛЯЮЩИЕ ОПЕРАТОРЫ

Оператор модификации. Оператор предназначен для описания вводного и выводного файлов и указания режима работы программы. Формат:

// Uxx TF,Ff,A = (n,m), B = (вывод),I = x,Ox,Rx,C = (c_1,...,c_7),
Q = q,E = (e)

где // Uxx — признак и код оператора модификации. Инициалы программы xx (PT для программы Перфолента-лента и PD для программы Перфолента-диск) могут отсутствовать;

TF — функция: перекомпоновка;

Ff — формат логических записей вводного файла: FF — записи фиксированной длины, FU — записи неопределенной длины.

Примечания: 1. Физические записи вводного файла должны быть неопределенного формата.

2. Записи выводного файла всегда имеют фиксированную длину.

3. Наличие ключей для файла на дисках указывается в параметре B;

A = (n,m) — описание вводного файла (n — количество полей в логической записи: не более 255; m — максимальная длина физической записи с учетом всех служебных символов);

B = (вывод) — описание выводного файла: B = (n,m) — для записей на магнитной ленте и дисках без ключей (n — длина записи, m — длина блока). Для неблокированных записей n и m должны быть одинаковыми, для блокированных — m должно быть кратно n. B = (K = *l*,D = *l*) — для записей на дисках с ключами (K = *l* — длина области ключа, D = *l* — длина области данных);

I = x — режим ввода: I = 2 — ввод в коде ISO, I = 3 — ввод в коде ДКОИ, I = имя — ввод в любом коде пользователя (указывается имя фазы, являющейся таблицей перекодировки);

Ox — режим вывода:

для программы PTDK: OY — выполняется проверка записанной на диск информации, ON — отсутствует проверка записанной на диск информации;

для программы PTTP: OR — перемотка ленты как до перезаписи файла, так и после перезаписи файла, ON — нет перемотки ленты ни до перезаписи файла, ни после перезаписи, OU — перемотка ленты до перезаписи и перемотка с разгрузкой после перезаписи файла;

Rx — номер логической записи вводного файла, начиная с которой выполняется перезапись (будет пропущено x — 1 записей);

C=(c_1, ..., c_7) — служебные символы на перфоленте (каждый в виде двух шестнадцатеричных цифр): c_1 — признак одного регистра (в коде перфоленты); c_2 — признак другого регистра (в коде перфоленты); c_3 — «Конец поля» (в коде ДКОИ); c_4 — «Конец логической записи» (в коде ДКОИ); c_5 — символ удаления (в коде перфоленты); c_6 — «Ошибочная логическая запись» (в коде ДКОИ); c_7 — «Ошибочная запись» (в коде ДКОИ). Параметр должен отсутствовать для кода ISO, для остальных кодов параметр обязателен;

Q = q — идентификатор файла на перфоленте (не более 15 символов). Используется для проверки меток (см. 7.2.4);

E = (e) — шифр накопителя на магнитных дисках. Для устройств ЕС-5052, ЕС-5055 и ЕС-5056 указывается обобщенный шифр 5052, для устройства ЕС-5061 — шифр 5061.

Значение по умолчанию (необязательные параметры): I = 1,OY (для PTDK), OU (для PTTP), R1, E = (5052), C = (00, 00, 09, 1E, FF, 18, 18).

Оператор перекомпоновки. Оператор предназначен для указания информации, необходимой для пересылки полей логических записей и их преобразования. Формат:

// FS параметр/параметр/ . . ./параметр

Параметр перекомпоновки зависит от формата логических записей вводного файла, способа пересылки поля (с преобразованием или без преобразования) и его местоположения (ключ или данные).

Для записей фиксированной и неопределенной длины:

r,(P,n,m),t
r,(P,n,m),(K,t) } — упаковка поля;

r,(D,n) — удаление поля вводной записи;

(I,m,xx),t
(I,m,xx),(K,t) } — добавление поля в выводную запись.

Для записей фиксированной длины:

r,s,t
r,s,(K,t) } — пересылка поля без изменения.

Для записей неопределенной длины:

r,(L,m,xx),t
r,(L,m,xx),(K,t) } — расширение поля (поле слева);

r,(R,m,xx),t
r,(R,m,xx),(K,t) } — расширение поля (поле справа).

Используемые обозначения:
r — порядковый номер поля во вводной записи;
t — порядковый номер поля в выводной записи;
s — длина поля (входного и выходного);
n — длина входного поля;
m — длина выходного поля;
xx — символ-заполнитель (в шестнадцатеричном виде);
K — признак того, что поле находится в области ключа;
P — признак упаковки;
D — признак удаления;
I — признак добавления;
L — признак смещения поля влево (при расширении);
R — признак смещения поля вправо (при расширении).

7.2.4. ОБРАБОТКА МЕТОК

Обработка меток на перфоленте. Выполняется проверка идентификатора, содержащегося в начальной метке, если задан параметр Q оператора модификации.

Обработка меток на магнитной ленте и дисках. Режим обработки меток указывается через оператор Управления заданиями UPSI.

Биты 2, 3 — обработка меток выводного файла (см. 7.1.4).

Бит 4 — запись ленточной марки в начале выводного файла без меток (см. 7.1.4).

Подпрограмма пользователя для обработки меток. Подпрограмма аналогична соответствующей подпрограмме для общих программ перезаписи (см. 7. 1. 4). Отличия: не может содержать подпрограмму обработки меток вводного файла; оператор IJWLABIN EQU IJWLABND не требуется; имена и инициалы программ: Перфолента-лента — PTTP и PT, Перфолента-диск — TPDK и PD.

7.3. СЕРВИС

7.3.1. НАЗНАЧЕНИЕ И ФУНКЦИИ

Программа осуществляет перезапись информации произвольной структуры между различными носителями данных и выполняет некоторое обслуживание носителей данных. Может вводить управляющие операторы с перфокарт или, работая в режиме диалога, — с пишущей машинки. Выполняет следующие функции:

- Карточные функции, обеспечивающие дублирование перфокарт, печать содержимого перфокарт, перезапись данных с перфокарт на магнитную ленту.
- Ленточные функции, обеспечивающие перезапись данных с магнитной ленты на перфокарты или другую ленту, печать данных с магнитной ленты, изменение физических записей на магнитной ленте, поиск на магнитной ленте физической записи по заданному аргументу поиска, инициализацию магнитной ленты, управление магнитной лентой.

- Дисковые функции, обеспечивающие печать данных с дисков, изменение физической записи на дисках, поиск на дисках физических записей по заданному аргументу поиска, изменение идентификации пакета дисков (регистрационного номера), запись признака «Конец файла» по указанному адресу.
- Вспомогательные функции, позволяющие получить справочные сведения о возможностях и способе использования программы и завершить выполнение программы.

7.3.2. СПОСОБ ИСПОЛЬЗОВАНИЯ

Имя программы — DITTO.

Условия использования. Разделы фоновый и переднего плана (18К без функции XXX или 26 Кбайт с функцией XXX), программа самоперемещающаяся. При выполнении программы в режиме диалога оператор должен отменить временные назначения для логических устройств: по завершении работы программы (после функции EOJ) ввести оператор /&.

Логические устройства: SYSRDR;

SYSIPT — ввод управляющих операторов с перфокарт и ввод данных при выполнении карточных функций;

SYSPCH — перфорация данных;

SYSLST — вывод сообщений программисту и печать данных;

SYSLOG — связь с оператором и ввод управляющей информации в режиме диалога;

SYSnnn — ввод данных при выполнении карточных функций в режиме диалога, а также ввод данных с магнитной ленты и дисков и вывод данных на магнитную ленту и диски при выполнении ленточных и дисковых функций.

Примечание. Логические устройства указываются программистом, когда управляющие операторы вводятся с перфокарт. В режиме диалога оператор указывает абсолютные адреса периферийных устройств, а программа сама выбирает логические устройства SYSnnn. Системные логические устройства предполагаются по умолчанию.

Задание на выполнение

Оператор	Пояснения
// JOB	Начало задания
[// UPSI 1]	Если управляющие операторы на перфокартах
[// ASSGN]	Назначение периферийных устройств, если указываются логические устройства
// EXEC DITTO	Выполнение программы
[Управляющие операторы программы Сервис на перфокартах]	
/&	Конец задания

7.3.3. ФУНКЦИИ ПРОГРАММЫ СЕРВИС

Карточные функции

CC — Копирование перфокарт

CCS — Дублирование перфокарт с перфорацией на каждой карте последовательного номера и идентификатора по пра-

вилам идентификации исходных программ Кобола, РПГ, Ассемблера (Фортрана)

CD — Печать перфокарт в символьно-шестнадцатеричном формате

CP — Печать перфокарт в символьном формате

CT — Перезапись перфокарт на магнитную ленту с блокированием записей

CTS — Перезапись перфокарт на магнитную ленту с добавлением последовательного номера и идентификатора и блокированием записей

Ленточные функции

TC — Перфорация неблокированных или блокированных записей с магнитной ленты

TD, TDD, TDV — Печать данных с магнитной ленты в символьно-шестнадцатеричном формате: TD — неблокированные записи, TDD — блокированные записи фиксированной длины, TDV — неблокированные или блокированные записи переменной длины

TP, TPD, TPV — Печать данных с магнитной ленты в символьном формате: TP — неблокированные записи, TPD — блокированные записи фиксированной длины, TPV — неблокированные или блокированные записи переменной длины

TFA, TFD — Печать данных с магнитной ленты, содержащих управляющие символы печати: TFA — управляющие символы типа A, TFD — управляющие символы типа D

TT — Копирование от 1 до 99 файлов одной магнитной ленты на другую ленту (под файлом подразумеваются данные между двумя ленточными марками)

TTR — Перезапись файла с записями фиксированной длины с одной магнитной ленты на другую с переблокированием записей

TRL* — Корректировка на магнитной ленте физических записей (усечение или расширение записей, изменение их содержимого)

TRS* — Поиск на магнитной ленте физической записи по заданному аргументу поиска и установка ленты перед заданной записью

INT* — Инициализация магнитной ленты, т. е. подготовка ленты для последующей записи на нее файлов со стандартными метками

BSF — Шаг на группу зон назад на магнитной ленте

BSR — Шаг на зону назад на магнитной ленте

FSF — Шаг на группу зон вперед на магнитной ленте

FSR — Шаг на зону вперед на магнитной ленте

REW — Перемотка магнитной ленты

RUN — Перемотка и разгрузка магнитной ленты

WTM — Запись ленточной марки на магнитную ленту

Примечание. Функции, отмеченные звездочкой, выполняются только в режиме диалога.

Дисковые функции

DD(SD) DDD(SDD)	Печать данных с дисков в символьно-шестнадцатеричном формате: DD и SD — неблокированные записи, DDD и SDD — блокированные записи фиксированной длины
DP(SP) DPD(SPD)	Печать данных с дисков в символьном формате: DP и SP — неблокированные записи, DPD и SPD — блокированные записи фиксированной длины
DRL*	Корректировка физической записи на дисках (неразделенные и разделенные цилиндры)
DRS(SRS)*	Поиск на дисках в пределах указанного участка всех физических записей по заданному аргументу поиска и печать найденных записей. Может осуществляться поиск по ключу, поиск по данным или поиск записей «Конец файла»
DID*	Изменение регистрационного номера тома на пакете дисков
EOF*	Запись признака «Конец файла»

Примечания:
1. Функции, отмеченные звездочкой, выполняются только в режиме диалога.
2. Функции, указанные в скобках, используются для разделенных цилиндров.

Вспомогательные функции

XXX	Справочные сведения о программе Сервис
EOJ	Завершение выполнения программы

7.3.4. УПРАВЛЯЮЩИЕ ОПЕРАТОРЫ НА ПЕРФОКАРТАХ

Список управляющих операторов

```
$$DITTO CC
$$DITTO CCS DECKTYPE = xxx, DECKNAME = xxx...x
$$DITTO CD
$$DITTO CP
$$DITTO CT OUTPUT = SYSnnn, BLKFACTOR = nnn
$$DITTO CTS OUTPUT = SYSnnn, BLKFACTOR = nnn, DECTYPE =
          xxx, DECKNAME = xx...x
$$DITTO TC INPUT = SYSnnn
$$DITTO TD INPUT = SYSnnn [, NBLKS = nnnn]
$$DITTO TDD INPUT = SYSnnn, RECSIZE = nnnnn
          [,NBLKS = nnnn]
$$DITTO TDV INPUT = SYSnnn [,NBLKS = nnnn]
$$DITTO TP INPUT = SYSnnn [,NBLKS = nnnn]
$$DITTO TPD INPUT = SYSnnn, RECSIZE = nnnnn [,NBLKS=nnnn]
$$DITTO TPV INPUT = SYSnnn [,NBLKS = nnnn]
$$DITTO TFA INPUT = SYSnnn
$$DITTO TFD INPUT = SYSnnn,
$$DITTO TT INPUT = SYSnnn, OUTPUT = SYSnnn
```

```
$$DITTO  TTR   INPUT=SYSnnn,OUTPUT=SYSnnn,RECSIZE=
                    nnnnn, BLKFACTOR=nnn
$$DITTO BSF OUTPUT=SYSnnn
$$DITTO BSR OUTPUT=SYSnnn,NBLKS=nnnn
$$DITTO FSF OUTPUT=SYSnnn
$$DITTO FSR OUTPUT=SYSnnn,NBLKS=nnnn
$$DITTO REW OUTPUT=SYSnnn
$$DITTO RUN OUTPUT=SYSnnn
$$DITTO WTM OUTPUT=SYSnnn
$$DITTO DD  INPUT=SYSnnn,BEGIN=ccchh,END=ccchh
$$DITTO DDD INPUT=SYSnnn,BEGIN=ccchh,END=ccchh,
                    RECSIZE=nnnnn
$$DITTO DP  INPUT=SYSnnn,BEGIN=ccchh,END=ccchh
$$DITTO DPD INPUT=SYSnnn,BEGIN=ccchh,END=ccchh,
                    RECSIZE=nnnnn
$$DITTO XXX
$$DITTO EOJ
```

Примечание. Операторы для функций SD, SDD, SP, SPD (разделенные цилиндры) аналогичны соответствующим операторам для неразделенных цилиндров.

Операнды управляющих операторов

BEGIN=ccchh — нижняя граница участка дисков (ссс — номер цилиндра, hh — номер дорожки). Для разделенных цилиндров указывается первый цилиндр и первая дорожка участка;

BLKFACTOR=nnn — коэффициент блокирования;

DECKNAME = xx...x — идентификатор для дублируемых перфокарт (или их образов);

DECKTYPE=xxx — тип колоды перфокарт: COB — исходная программа на Коболе (номер в колонках 1— 6, идентификатор в колонках 73 — 80), RPG — исходная программа на РПГ (номер в колонках 1— 5, идентификатор в колонках 75 — 80), BAL — исходная программа на языке Ассемблера, Фортране, управляющие операторы (номер в колонках 76 — 80, идентификатор в колонках 73 — 75);

END = ccchh — верхняя граница участка дисков (ссс — номер цилиндра, hh — номер дорожки). Для разделенных цилиндров указывается последний цилиндр участка и дорожка разделения;

INPUT=SYSnnn — логическое устройство для ввода данных;

NBLKS=nnnn — количество пропускаемых блоков магнитной ленты: 1 — 9999;

OUTPUT=SYSnnn — логическое устройство для вывода данных;

RECSIZE=nnnnn — длина логических записей: 1—99999.

7.3.5. УПРАВЛЯЮЩАЯ ИНФОРМАЦИЯ (РЕЖИМ ДИАЛОГА)

В режиме диалога можно сократить число запросов программы и ответов оператора, если сразу ввести необходимые параметры.

Список параметров

CCS	DECKTYPE,DECKNAME
CT	CUUMM,BLKFACTOR
CTS	CUUMM,BLKFACTOR,DECKTYPE,DECKNAME
TC	CUUMM
TD	CUUMM,NRECS
TDD	CUUMM,RECSIZE,NRECS
TDV	CUUMM,NRECS
TP	CUUMM,NRECS
TPD	CUUMM,RECSIZE,NRECS
TPV	CUUMM,NRECS
TFA	CUUMM
TFD	CUUMM
TT	CUUMM,CUUMM,NFILES
TTR	CUUMM,CUUMM,RECSIZE,BLKFACTOR
TRL	CUUMM,CUUMM,NBLKS,COPYD
TRS	CUUMM,RECSIZE,SCANLNTH
INT	CUUMM,VOLSER
BSF	CUUMM,NFILES
BSR	CUUMM,NBLKS
FSF	CUUMM,NFILES
FSR	CUUMM,NBLKS
REW	CUUMM
RUN	CUUMM
WTM	CUUMM
DD(SD)	CUU,CCCHH,CCCHH
DDD(SDD)	CUU,CCCHH,CCCHH,RECSIZE
DP(SP)	CUU,CCCHH,CCCHH
DPD (SPD)	CUU,CCCHH,CCCHH,RECSIZE
DRL	CUU,CCC — HH — RRR
DRS (SRS)	CUU,CCCHH,CCCHH,SCANTYPE
DID	CUU,VOLSER
EOF	CUU,CCC — HH — RRR

Примечание. Для функций CC, CD, CP, XXX, EOJ параметры не требуются.

Назначение параметров

BLKFACTOR	Коэффициент блокирования (1 — 999)
CCCHH	Номера цилиндра и дорожки
CCC — HH — RRR	Номера цилиндра, дорожки и записи
COPYD	Направление движения магнитной ленты (F— вперед, B — назад)
CUU	Номер канала и номер устройства
CUUMM	Номер канала, номер устройства и плотность записи
DECKNAME	Идентификатор колоды
DECKTYPE	Тип колоды (COB, RPG, BAL)
NBLKS	Количество блоков (1 — 9999)
NFILES	Количество файлов (1 — 99)
NRECS	Количество записей

RECSIZE	Длина логических записей (1 — 99999) либо для функции TRS признак записей переменной длины—буква V
SCANLNTH	Длина аргумента поиска (1 — 35)
SCANTYPE	Тип поиска (K — по ключу, D — по данным, E — поиск записей «Конец файла»)
VOLSER	Регистрационный номер тома

7.4. МАКРОКОМАНДЫ ПЕРЕЗАПИСИ

7.4.1. НАЗНАЧЕНИЕ И ВОЗМОЖНОСТИ

Макрокоманды перезаписи предназначены для ввода с носителей данных в основную память и вывода из основной памяти на носители данных последовательно организованных файлов, состоящих из записей фиксированной или неопределенной длины. Используются, когда программа составляется на языке Ассемблера. Обеспечивают следующие возможности:

- Обрабатывают файлы с системными метками, с системными метками и метками пользователя (диски и магнитная лента), а также файлы без меток (магнитная лента).
- Обрабатывают многотомные файлы на дисках и магнитных лентах. Файлы на дисках могут быть на разделенных или неразделенных цилиндрах, для файлов на магнитных лентах разрешается использовать сменные устройства.
- Пропускают имеющиеся записи контрольных точек на магнитных лентах или обрабатывают их как данные.
- При выводе на диски могут полностью заполнять дорожки либо записывать на них указанное пользователем количество физических записей.
- Осуществляют ввод и вывод карт в коде ДКОИ — карты в коде КПК-12, автоматическое преобразование в код или из кода ДКОИ.
- Выводят карты в карман 2 или обеспечивают управление выбором кармана по первому символу (тип D). Управляющий символ выбора кармана не перфорируют.
- Печатают информацию через одну строку или обеспечивают управление форматом печати по первому символу (типы A, C, D). Управляющий символ печати не выводят.
- Печатают информацию в символьном виде.
- Позволяют обрабатывать перфоленты в любом коде. Для перфолент в коде КОИ-7 выполняют перекодировку в код ДКОИ и наоборот.
- Позволяют включать в составляемую программу подпрограммы пользователя, выполняющие обработку записей, обработку ошибок и меток пользователя, перекодировку информации из любых кодов перфоленты и т. д.
- При каждом обращении к макрокоманде ввода выбирают из вводного файла очередную логическую запись, при необходимости выполняя деблокирование, помещают ее в основную память и сообщают ее адрес и длину в регистрах.

- При каждом обращении к макрокоманде вывода помещают в выводной файл из основной памяти очередную логическую запись, адрес и длина которой указаны в регистрах. При необходимости выполняют блокирование записей.
- С помощью одной макрокоманды могут осуществлять ввод или вывод нескольких файлов.
- Являются самоперемещающимися и могут выполняться в любом разделе основной памяти без повторного редактирования.
- Ориентированы на выполнение в разделах переднего плана (в том числе и в режиме одиночных программ), хотя могут выполняться и в фоновом разделе.

7.4.2. СВЯЗЬ МЕЖДУ МАКРОКОМАНДАМИ ВВОДА И ВЫВОДА

Связь осуществляется с помощью регистров. Использование регистров показано в табл. 7.3. Регистры 0 и 1 используются также для управления макрокомандами вывода (табл. 7.4).

Таблица 7.3

Использование регистров макрокомандами перезаписи

Регистр	Подпрограмма ввода	Подпрограмма вывода
0	Помещает адрес записи	Получает длину записи (исключение — OUTLOG)
1	Помещает длину записи	Получает длину записи (исключение — OUTLOG)
10—12	Не изменяет	Не изменяет. Исключение — OUTDISK: если длина области вывода равна 1450 — 3500 байт, в качестве регистра базы дополнительно используется регистр 10; если длина области вывода равна 3501 — 5500 байт, в качестве регистров базы дополнительно используются регистры 10 и 11; если длина области вывода больше 5500 байт, в качестве регистров базы дополнительно используются регистры 10, 11 и 12
13	Использует в качестве регистра связи	Использует в качестве регистра связи
14	Помещает адрес точки повторного входа	Получает адрес выхода из подпрограммы, если разрешен возврат в подпрограмму ввода и выполнение программы не закончено
15	Использует в качестве регистра базы	Использует в качестве регистра базы

Таблица 7.4

Управление макрокомандами вывода

Регистр 0	Регистр 1	Действие макрокоманды вывода	Передача управления макрокомандой вывода	
			RETURN=YES	RETURN=NO
≠0	≠0	Обеспечивает вывод очередной записи, адрес которой указан в регистре 0, длина — в регистре 1	По адресу в регистре 14	Следующей команде
0	0	Выводит последний блок, если нужно, и закрывает выводной файл	Завершает выполнение программы (EOJ)	Следующей команде
≠0	0	Игнорирует очередную запись, адрес которой указан в регистре 0	По адресу в регистре 14	Следующей команде
0	≠0	Выводит последний блок, если нужно, закрывает выводной файл и настраивается на вывод следующего файла	По адресу в регистре 14	Следующей команде

Примечание. Макрокоманда ввода устанавливает оба регистра или в ненулевое значение (адрес и длина записи), или в нулевое (конец файла); различные значения регистров можно установить подпрограммой пользователя.

7.4.3. ФОРМАТ МАКРОКОМАНД

Макрокоманды ввода

INCARD — ввод с перфокарт,
INTAPE — ввод с магнитной ленты,
INDISK — ввод с дисков,
INPATA — ввод с перфоленты,
INLOG — ввод с пишущей машинки.

Перечень операндов макрокоманд ввода приведен в табл. 7.5.

Макрокоманды вывода

OUTCARD — вывод на перфокарты,
OUTAPE — вывод на магнитную ленту,
OUTDISK — вывод на диски,
OUTPATA — вывод на перфоленту,
OUTPRT — вывод на печать,
OUTLOG — вывод на пишущую машинку.

Перечень операндов макрокоманд вывода приведен в табл. 7.6.

Таблица 7.5

Макрокоманды ввода

Операнд	INCARD	INTAPE	INDISK	INPATA	INLOG
BUFFER= {имя / (r)}					+
BUFSIZ=n		+	+	+	
CHARACT= {(e, s, b) / (1E, 18, 38)}				*	
CHKPT= {NO / имя}		*			
CODE= {E / I / имя}				*	
COUNT= {n / (r)}					*
DEVICE=шифр			*		
ERROR= {SKIP / IGNORE / имя}		*	*	*	
FILE= {имя файла / (r)}		*	+		
LBL=имя		*	*		
RECFORM= {UNDEF / FIX}				*	
RECSIZ= {m / имя / RECEND}		*	*	*	
UNIT= {SYS001 / SYSnnn}	*	*	*	*	

Примечания: 1. Символ «+» означает обязательный операнд, символ «*» означает необязательный операнд.

2. Значение RECSIZ=RECEND может использоваться только для макрокоманды INPATA.

Таблица 7.6

Макрокоманды вывода

Операнд	OUTCARD	OUTAPE	OUTDISK	OUTPATA	OUTPRT	OUTLOG
BLK= {m / имя}		*	*	*		
BUFFER= {имя / (r)}						+
BUFSIZ=n		+	+	+	*	
CHARACT= {e / <u>1E</u>}				*		
CODE= {<u>E</u> / I / имя}				*		
COUNT= {n / (r)}						*
DEVICE=шифр	+		*			
EOR=r				*		
ERROR= {<u>IGNORE</u> / имя}			*			
FILE= {имя файла / (r)}		*	+			
FORMAT= {<u>FULL</u> / n / имя}			*			
FORMS= {A / C / D}					*	
LBL=имя		*	*			
RECEND=YES				*		
RETURN= {<u>YES</u> / NO}	*	*	*	*	*	*
STCTL= {<u>NO</u> / YES / имя}	*					
UNIT= {<u>SYS002</u> / SYSnnn}	*	*	*	*	*	

Примечание. Символ «+» означает обязательный операнд, символ «*» означает необязательный операнд.

Операнды макрокоманд. Ниже в алфавитном порядке приводится описание операндов всех макрокоманд перезаписи.

BLK = $\left\{\begin{matrix} m \\ \text{имя} \end{matrix}\right\}$ Блокирование записей: m — коэффициент блокирования, если $m > 1$; имя — символический адрес слова с коэффициентом блокирования, если содержимое больше 1. Если операнд отсутствует, $m \leqslant 1$, содержимое слова меньше или равно 1, записи не блокируются.

BUFFER= $\left\{\begin{matrix} \text{имя} \\ (r) \end{matrix}\right\}$ Адрес области ввода или вывода: имя — символический адрес области; (r) — номер регистра с адресом области.

BUFSIZ = n n — длина области ввода или вывода. В случае отсутствия операнда для OUTPRT n = 144.

CHARACT= $\left\{\begin{matrix} (e, s, b), \\ (1E, 18, 38) \end{matrix}\right\}$ Служебные символы при вводе с перфоленты в коде ДКОИ (каждый в виде двух шестнадцатеричных цифр): e — символ «Конец логической записи»; s — символ «Ошибочная логическая запись»; b — символ «Ошибочная запись». В случае отсутствия операнда для кода КОИ-7 используются указанные значения. Для других кодов операнд обязателен, если RECSIZ = = RECEND.

CHARACT = $\left\{\begin{matrix} e \\ 1E \end{matrix}\right\}$ Служебный символ при выводе на перфоленту в коде ДКОИ (в виде двух шестнадцатеричных цифр): e — символ «Конец логической записи». В случае отсутствия операнда для кода КОИ-7 используется указанное значение. Для других кодов операнд обязателен, если RECEND = YES.

CHKPT = $\left\{\begin{matrix} NO \\ \text{имя} \end{matrix}\right\}$ Обработка контрольных точек: NO — обрабатываются как данные; имя — символический адрес слова, содержимое которого указывает способ обработки: ненулевое содержимое— обрабатываются как данные, нулевое содержимое—пропускаются. Если операнд отсутствует, контрольные точки пропускаются.

CODE = $\left\{\begin{matrix} E \\ I \\ \text{имя} \end{matrix}\right\}$ Код перфоленты: E — код E (нет перекодировки); I — код I (перекодировка из кода или в код КОИ-7);

имя — символический адрес подпрограммы пользователя для перекодировки каждой физической записи при вводе или каждой логической записи при выводе. Регистр 0 содержит адрес записи, регистр 1— ее длину. Возврат из подпрограммы через регистр 14. Если операнд отсутствует, предполагается код E.

COUNT = $\left\{\begin{matrix} n \\ (r) \end{matrix}\right\}$ Количество вводимых или выводимых символов: n — количество символов; (r) — номер регистра с количеством сим-

волов. Если операнд отсутствует и BUFFER = имя, количество символов равно длине области ввода или вывода.

DEVICE = шифр. Шифр устройства ввода-вывода. Для INDISK и OUTDISK указывается обобщенный шифр 5052 для устройств EC-5052, EC-5055, EC-5056 и шифр 5061 — для устройства EC-5061.

EOR = r r — символ «Конец записи» (в виде двух шестнадцатеричных цифр в коде перфоленты). Если операнд отсутствует, символ «Конец записи» не перфорируется.

$$\text{ERROR} = \left\{ \begin{array}{c} \text{SKIP} \\ \text{IGNORE} \\ \hline \text{имя} \end{array} \right\}$$ Обработка ошибок: SKIP — любой ошибочный блок пропускается (только для макрокоманд ввода); IGNORE — ошибка игнорируется и блок обрабатывается; имя — символический адрес подпрограммы пользователя для обработки ошибок. Регистр 0 содержит адрес блока, регистр 1 — длину. Возврат из подпрограммы через регистр 14. Подпрограмма пользователя указывает содержимым регистра 1 дальнейшую обработку ошибочного блока: нулевое содержимое — блок пропускается, ненулевое содержимое—блок обрабатывается. Если операнд отсутствует, ошибка игнорируется.

$$\text{FILE} = \left\{ \begin{array}{c} \text{имя файла} \\ (\text{r}) \end{array} \right\}$$ Наличие системных меток и имя файла, указываемое в операторе TLBL или DLBL: имя файла — константа (1 — 7 символов); (r) — номер регистра с адресом имени файла (7 символов). Если операнд отсутствует, предполагается, что файл не имеет меток. (Отсутствие меток можно указать и в операнде LBL при наличии операнда FILE.)

$$\text{FORMAT} = \left\{ \begin{array}{c} \text{FULL} \\ \hline \text{n} \\ \text{имя} \end{array} \right\}$$ Формат дорожки диска: FULL — дорожка заполняется полностью; n — количество записей на дорожке (если n слишком велико, дорожка заполняется полностью); имя — символический адрес слова, содержимое которого указывает количество записей на дорожке: если содержимое больше 0, оно является количеством записей; если содержимое меньше или равно 0, дорожка заполняется полностью. Если операнд отсутствует, дорожка заполняется полностью.

$$\text{FORMS} = \left\{ \begin{array}{c} \text{A} \\ \text{C} \\ \text{D} \end{array} \right\}$$ Управление форматом печати по первому символу и тип управления форматом печати. Управляющий символ не печатается. Если операнд отсутствует, управление форматом печати по первому символу не осуществляется.

LBL = имя имя — символический адрес области, содержимое которой указывает режим обработки меток (см. также операнд FILE): если содержимое равно 0 для файлов на магнитных лентах и меньше

или равно 0 для файлов на дисках, обрабатываются только системные метки, а имеющиеся метки пользователя пропускаются; если содержимое меньше 0 для файлов на магнитных лентах, метки файла не обрабатываются; если содержимое больше 0, обрабатываются системные метки и метки пользователя, а область должна содержать адрес подпрограммы пользователя для обработки меток.

RECEND=YES Признак наличия символа «Конец логической записи», который задается операндом CHARACT.

RECFORM = {UNDEF / FIX} Формат физических записей на перфоленте: UNDEF — неопределенный формат (наличие символа «Конец записи»); FIX — фиксированный формат (отсутствие символа «Конец записи»). Если операнд отсутствует, предполагается неопределенный формат.

RECSIZ = {m / имя / RECEND} Деблокирование записей: m — длина логической записи, если m > 0; имя — символический адрес слова с длиной логической записи, если содержимое больше 0; RECEND — при вводе с перфоленты признак наличия символа «Конец логической записи», который указывается операндом CHARACT. Если операнд отсутствует, m = 0 или содержимое слова равно 0, предполагается, что записи неблокированные.

RETURN = {YES / NO} Выход из макрокоманды вывода: YES — выход по адресу в регистре 14; NO — выход к команде, следующей за макрокомандой вывода.

STCTL = {NO / YES / имя} Выбор кармана: NO — нет управления выбором кармана, все карты попадают в карман 2, перфорируются все символы записи; YES — карман определяется первым символом каждой логической записи, управляющий символ не перфорируется; имя — символический адрес слова, содержимое которого указывает способ выбора кармана: нулевое содержимое — управление выбором кармана не обеспечивается; ненулевое содержимое — управление выбором кармана обеспечивается. Если операнд отсутствует, предполагается STCTL = NO.

UNIT = {SYS001 / SYSnnn} Логическое устройство для ввода файлов. Если операнд отсутствует, используется SYS001.

UNIT = {SYS002 / SYSnnn} Логическое устройство для вывода файлов. Если операнд отсутствует, используется SYS002.

7.5. ПРОГРАММА СОРТИРОВКИ НА МАГНИТНЫХ ЛЕНТАХ

7.5.1. НАЗНАЧЕНИЕ И ВОЗМОЖНОСТИ

Программа предназначена для сортировки-слияния последовательно организованных файлов, расположенных на магнитных лентах, и обеспечивает следующие возможности:

- Сортирует до 9 файлов в один упорядоченный файл.
- Объединяет до 7 упорядоченных файлов в один упорядоченный файл.
- Выполняет сортировку-слияние файлов на магнитных лентах, содержащих записи фиксированной или переменной длины, блокированные и неблокированные. Файлы могут быть многотомными, иметь системные или нестандартные метки или быть без меток.
- Выполняют сортировку-слияние записей файлов максимум по 12 управляющим полям, содержащим символьные данные или десятичные числа в упакованном или распакованном формате, или числа с фиксированной или плавающей точкой. Сортировка-слияние может быть выполнена как по убыванию, так и по возрастанию управляющих полей. Формат данных должен быть одинаков для всех управляющих полей, а направление упорядочивания может быть различным.
- Позволяет использование для вводных, выводного и рабочих файлов только накопителей на магнитной ленте
- Позволяет использование для рабочих файлов от 3 до 6 накопителей на магнитной ленте.
- Может создавать записи контрольных точек при выполнении процесса сортировки, что позволяет прекратить процесс сортировки при необходимости и осуществить повторный запуск с последней контрольной точки.
- Позволяет подключать подпрограммы пользователя для обработки меток пользователя и нестандартных меток, корректировки вводных и выводного файлов, преобразования управляющих полей, обработки ошибок ввода-вывода и других действий.

7.5.2. СПОСОБ ИСПОЛЬЗОВАНИЯ

Имя программы — TSRT.

Условия использования. Фоновый раздел (10K), требуется редактирование для раздела.

Логические устройства: SYSRDR, SYSIPT, SYSLST, SYSLOG; SYS001 — SYS009 — вводные, выводной и рабочие файлы (см. табл. 7.7).

Таблица 7.7

Имена логических устройств и файлов для программы TSRT

Назначение устройства	Сортировка		Слияние	
	Имя файла	Имя логического устройства	Имя файла	Имя логического устройства
Выводной файл	FILEO	SYS001	FILEO	SYS001
Вводные файлы	FILEA ... FILEI	SYS002	FILEA ... FILEG	SYS002 ... SYS008
Рабочие файлы	—	SYS003 ... SYS008	—	—
Сменное устройство для рабочих файлов	—	SYS009	—	—

Задание на выполнение

Оператор	Пояснения
// JOB	Начало задания
// TLBL	Если файлы на магнитных лентах имеют системные метки, информация о метках
// ASSGN	Назначение периферийных устройств
// EXEC TSRT	Выполнение программы
Управляющие операторы программы (SORT или MERGE, RECORD, MODS, INPFIL, OUTFIL, OPTION, END)	
/&	Конец задания

Операторы SORT (MERGE), RECORD, INPFIL, END являются обязательными, а MODS, OUTFIL, OPTION могут отсутствовать.

7.5.3. УПРАВЛЯЮЩИЕ ОПЕРАТОРЫ

Общие правила записи. Структура:

⟨признак⟩ ⟨код⟩ ⟨операнды⟩

В качестве признака используется пробел. Составные части оператора разделяются не менее чем одним пробелом. Для записи оператора могут использоваться перфокарты продолжения. При использовании перфокарт продолжения нужно следить за тем, чтобы общее количество перфокарт с управляющими операторами сортировки не превышало 25.

Оператор SORT предназначен для описания управляющих полей при операции сортировки, задания числа рабочих файлов и режима для создания контрольных точек. Формат:

SORT FIELDS = $(p_1,m_1,s_1,\ldots,p_n,m_n,s_n)$,FORMAT = xx,WORK = n [,CHKPT]

где FIELDS — описание управляющих полей для операции сортировки: p_i — относительный адрес i-го управляющего поля ($1 \leqslant i \leqslant 12$); m_i — длина i-го управляющего поля в байтах; s_i — направление упорядочения по i-му управляющему полю: A—в порядке возрастания значений управляющих полей, D—в порядке убывания значений управляющих полей;

FORMAT — формат управляющих полей: BI (CH) — двоичные данные без знака или символьные данные; ZD — десятичные распакованные числа; PD — десятичные упакованные числа; FI — числа с фиксированной точкой; FL — числа с плавающей точкой; SC — русский текст;

WORK — количество устройств для рабочих файлов при выполнении операции сортировки (n = 3, ..., 6);

CHKPT — режим работы программы, при котором создаются контрольные точки. Организацию повторного запуска с контрольной точки см. в 2.2.5.

Оператор MERGE предназначен для описания управляющих полей при операции слияния и задания порядка слияния. Формат:

MERGE FIELDS = $(p_1,m_1,s_1,\ldots,p_n,m_n,s_n)$,FORMAT = xx,ORDER = n

где FIELDS — описание управляющих полей для операции слияния (см. оператор SORT);

FORMAT — формат управляющих полей (см. оператор SORT)

ORDER — количество объединенных файлов ($1 \leqslant n \leqslant 7$)

Оператор RECORD предназначен для описания формата и задания длины обрабатываемых записей. Формат:

$$\text{RECORD}\left[\text{TYPE}=\begin{Bmatrix}\text{F}\\ \text{V}\end{Bmatrix},\right]\text{LENGTH}=\begin{Bmatrix}l_1,l_2,l_3\\ l_1,l_2,l_3,l_4\end{Bmatrix}$$

где TYPE — формат обрабатываемых записей:

F — записи фиксированной длины;

V — записи переменной длины;

LENGTH — описание длин обрабатываемых записей:

l_1 — длина записей вводного файла;

l_2 — длина записей рабочего файла;

l_3 — длина записей выводного файла;

l_4 — минимальная длина записей вводного файла для записей переменной длины.

Для записей переменной длины параметры l_1, l_2, l_3 означают максимальную длину записей.

Оператор INPFIL предназначен для описания структуры вводных файлов и действий по установке магнитных лент при открытии и закрытии вводных файлов. Формат:

$$\text{INPFIL BLKSIZE}=(n,X)[,\text{BYPASS}]\left[,\text{VOLUME}=\left\{\begin{matrix}n\\(n_1,n_2,\ldots,)\end{matrix}\right\}\right]$$

$$\left[,\text{OPEN}=\left\{\begin{matrix}\text{RWD}\\\text{NORWD}\end{matrix}\right\}\right]\left[,\text{CLOSE}=\left\{\begin{matrix}\text{RWD}\\\text{UNLD}\\\text{NORWD}\end{matrix}\right\}\right]$$

где BLKSIZE — описание длины и формата блоков вводного файла:

n — длина блоков для записей фиксированной длины или максимальная длина блоков для записей переменной длины;

X (X'E7') — признак блоков фиксированной длины;

BYPASS — пропуск блоков вводного файла, при считывании которых возникают ошибки чтения или ошибки длины;

VOLUME — количество томов, занимаемых вводными файлами:

n — общее количество томов, занимаемых вводными файлами для операции сортировки; n = 1, если VOLUME не задан;

n_i — количество томов, занимаемых i-м вводным файлом для операции слияния ($1 \leqslant i \leqslant 7$), $n_i = 1$, если VOLUME не задан;

OPEN — действия по установке магнитных лент перед открытием первого тома каждого вводного файла:

RWD — перемотка ленты к точке загрузки;

NORWD — перемотка ленты не выполняется;

CLOSE — действия по установке магнитных лент после закрытия последнего тома каждого вводного файла:

RWD — перемотка ленты к точке загрузки;

UNLD — перемотка и разгрузка ленты;

NORWD — перемотка ленты не выполняется.

Оператор OUTFIL предназначен для задания длины блоков выводного файла и действий по установке магнитной ленты при открытии и закрытии выводного файла. Формат:

$$\text{OUTFIL }[\text{BLKSIZE}=n]\left[,\text{OPEN}=\left\{\begin{matrix}\text{RWD}\\\text{NORWD}\end{matrix}\right\}\right]\left[,\text{CLOSE}=\left\{\begin{matrix}\text{RWD}\\\text{NORWD}\\\text{UNLD}\end{matrix}\right\}\right]$$

где BLKSIZE — длина блоков выводного файла для записей фиксированной длины или максимальная длина блоков выводного файла для записей переменной длины;

OPEN — аналогичен операнду OPEN оператора INPFIL;

CLOSE — аналогичен операнду CLOSE оператора INPFIL.

Оператор OPTION предназначен для указания типа меток обрабатываемых файлов и режимов работы программы. Формат:

OPTION [LABEL = (⟨ввод⟩, ⟨рабочий⟩, ⟨вывод⟩)]

[,NOTPMK] [,ALTWK] [,FILES = n] [,PRINT]

где LABEL — тип меток обрабатываемых файлов (вводных, рабочих, выводного):

S — системные метки и метки пользователя;

N — нестандартные метки;

U — файлы без меток;

NOTPMK — режим работы программы, при котором в начале каждого тома выводного файла без меток не записывается ленточная марка;

ALTWK — использование для рабочих файлов сменной магнитной ленты;

FILES — число вводных файлов для операции сортировки ($1 \leqslant n \leqslant 9$), n = 1, если FILES не задан;

PRINT — вывод на SYSLST сообщений программы.

Оператор MODS предназначен для описания выходов к подпрограммам пользователя, используемым для каждого этапа программы. Формат:

MODS PHn = (‹имя фазы›, ‹адрес загрузки›, Enm)

где PHn — описание выходов, используемых для n-го этапа программы; n = 1, 2, 3 соответственно для этапов внутренней сортировки, внешней сортировки и слияния;

имя фазы — имя фазы в библиотеке абсолютных модулей, содержащей программы пользователя, обслуживающие выходы данного этапа;

адрес загрузки — абсолютный адрес загрузки фазы пользователя в основную память (в десятичной системе счисления);

Enm — номер выхода данного этапа программы, используемый для подключения подпрограмм пользователя: E11, E12, E13, E14, E15 — для внутренней сортировки; E21, E22 — для внешней сортировки; E31, E32, E33, E34, E35 — для слияния.

Оператор END предназначен для указания конца управляющих операторов. Операндов не имеет.

7.5.4. ВЫХОДЫ К ПОДПРОГРАММАМ ПОЛЬЗОВАТЕЛЯ

Назначение и функции. Выходы программы сортировки-слияния обеспечивают подключение в определенных точках программы подпрограмм, написанных пользователем и предназначенных для выполнения некоторых дополнительных функций. Выходы предусмотрены для этапов внутренней и внешней сортировки и слияния. Каждый выход идентифицируется трехзначным обозначением Ei_j (i — номер этапа, j — обозначение выполняемой функции) и предназначен для выполнения одной или нескольких конкретных функций. Функции выходов приведены в табл. 7.8.

Связь с подпрограммами пользователя осуществляется посредством регистров, в которые перед передачей управления к подпрограммам пользователя программа сортировки заносит следующую информацию:

Таблица 7.8

Функции выходов программы сортировки на магнитных лентах

Функции	Внутренняя сортировка					Внешняя сортировка		Слияние				
	E11	E12	E13	E14	E15	E21	E22	E31	E32	E33	E34	E35
Обработка нестандартных меток или меток пользователя вводных файлов	×							×				
Обработка нестандартных меток рабочих файлов	×											
Создание нестандартных меток или меток пользователя выводного файла						×				×		
Преобразование управляющих полей записей вводных файлов			×						×			
Преобразование управляющих полей записей выводного файла							×				×	
Изменение и (или) удаление записей вводных файлов				×								
Изменение, вставка и (или) удаление записей выводного файла							×				×	
Суммирование записей выводного файла							×					
Обработка блоков вводных файлов, имеющих неверную длину или при считывании которых обнаружена ошибка чтения					×							×

R0—R2, R4—R12 — параметры, передаваемые подпрограммами пользователя (табл. 7.9);

R14 — адрес возврата в программу сортировки;

R15 — адрес таблицы переходов для подпрограмм выходов соответствующего этапа сортировки.

Таблица 7.9

Параметры, передаваемые подпрограммам пользователя для выходов TSRT

Выход	Регистр	Параметр
E11, E12 E21, E31	R0	Код действия (X'D6' — открывается файл, X'C6' — закрывается файл; X'ES' — открывается том)
E33	R1	Шифр логического устройства, используемого для файла (нестандартные метки) или адрес метки в основной памяти (метки пользователя)
E13	R8	Адрес очередной записи вводного файла
E14	R4	Адрес записи вводного файла, которая подлежит пересылке в область вывода
	R8	Адрес, по которому запись вводного файла пересылается в область вывода
E15	R1	Адрес CCB для файла, при считывании которого обнаружена ошибка
	R8	Адрес области ввода, куда считывался блок вводного файла
E22	R6	Адрес последней записи в области вывода
	R7	Уменьшенный на 12 адрес поля, в котором хранится адрес записи, подлежащей пересылке в область вывода
	R10	Признак обработки последней записи вводного файла (последняя запись — <R10> меньше 0)
E32	R4	Адрес обрабатываемой записи
	R5	Адрес записи, подлежащей обработке
	R9	Адрес таблицы, содержащей информацию о файле (табл. 7.10)
E34	R9	Адрес таблицы, содержащей информацию о файле (см. табл. 7.10)
	R10	Признак окончания процесса слияния (слияние окончено — <R10> меньше 0)
	R11	Адрес записи вводного файла, которая подлежит пересылке в область вывода
	R12	Адрес последней записи в области вывода
E35	R2	Уменьшенный на 24 адрес CCB для файла, при считывании которого обнаружена ошибка
	R9	Уменьшенный на 12 адрес поля, в котором хранится адрес информации считанного блока

Таблица 7.10

Таблица информации о файле

Байты	Содержимое
0—1	Количество записей в области ввода
2—3	Количество томов, занимаемых файлом
4—5	Шифр логического устройства, используемого для файла
6—9	Адрес области ввода
10—11	Количество обработанных блоков файла

7.6. ПРОГРАММА СОРТИРОВКИ НА МАГНИТНЫХ ЛЕНТАХ И ДИСКАХ

7.6.1. НАЗНАЧЕНИЕ И ВОЗМОЖНОСТИ

Программа предназначена для сортировки-слияния последовательно организованных файлов, расположенных на магнитных лентах или на дисках, и обеспечивает следующие возможности:

- Сортирует до 9 файлов в один упорядоченный файл.
- Объединяет до 8 упорядоченных файлов в один упорядоченный файл.
- Допускает использование для вводных, выводного и рабочих файлов накопителей на магнитной ленте или накопителей на магнитных дисках.
- Позволяет использование для рабочих файлов от 3 до 9 накопителей на магнитной ленте или от 1 до 8 накопителей на магнитных дисках.
- Выполняет сортировку-слияние файлов, содержащих записи фиксированной или переменной длины, блокированные или неблокированные. Файлы могут быть многотомными, иметь стандартные метки, а в случае магнитных лент — и нестандартные метки или быть без меток.
- Выполняет сортировку-слияние записей файлов максимум по 12 управляющим полям, содержащим символьные данные или десятичные числа в упакованном или распакованном формате, или числа с фиксированной или плавающей точкой. Сортировка-слияние может быть выполнена как по убыванию, так и по возрастанию управляющих полей. Формат данных и направление упорядочивания для каждого управляющего поля могут быть различными.
- Допускает выполнение сортировки-слияния либо в отдельном шаге задания, либо в процессе выполнения другой программы; в последнем случае программа сортировки-слияния вызывается для выполнения этой программой и используется как ее подпрограмма.
- Позволяет выполнять сортировку адресов файла на дисках, получая в результате файл адресов, в котором адреса записей исходного файла упорядочены в соответствии со значениями управляющих полей.
- Может создавать записи контрольных точек при выполнении процесса сортировки, что позволяет прекратить процесс сортировки при необходимости и осуществить повторный запуск с последней контрольной точки.
- Позволяет подключение подпрограмм пользователя для выполнения ввода-вывода и (или) корректировки вводных и выводных файлов, открытия и закрытия файлов с метками пользователя и нестандартными метками, преобразования данных управляющих полей, обработки ошибок ввода-вывода и других действий.

7.6.2. СПОСОБ ИСПОЛЬЗОВАНИЯ

Имя программы — SORT.

Условия использования: фоновый раздел, разделы переднего плана. Минимальная память — 10K при использовании для рабочих файлов магнитных лент или дисков типа EC-5052 и 22K при использовании для рабочих файлов дисков типа EC-5061. Программа самоперемещающаяся, редактирование для раздела не требуется.

Логические устройства: SYSRDR, SYSIPT, SYSLST, SYSLOG; SYS001 — SYS009 — вводные, выводной и рабочие файлы сортировки-слияния (табл. 7.11).

Таблица 7.11

Имена логических устройств и файлов для программы SORT

Назначение устройства	Имя файла	Имена логических устройств			
		Чтение вводного и запись выводного файла сортировкой-слиянием	Чтение вводного файла подпрограммой для выхода E15	Запись выводного файла подпрограммой для выхода E35	Чтение вводного и запись выводного файла подпрограммой для выходов E15, E35
Выводной файл	SORTOUT	SYS001	SYS001	—	—
Вводные файлы	SORTIN1 ... SORTIN9	SYS002 ... SYS(n+1)	—	SYS001 ... SYS(n)	—
Рабочие файлы	SORTWK1 ... SORTWK9	SYS(n+2) ... SYS(n++m+1)	SYS002 ... SYS(m++1)	SYS(n+1) ... SYS(n+m)	SYS001 ... SYS(m)
Сменное устройство для рабочих файлов	SORTALT	SYS(n++m+2)	SYS(m++2)	SYS(n++m+1)	SYS(m++1)
Файл контрольных точек	SORTCKP	SYS000	SYS000	SYS000	SYS000

Обозначения: n — число вводных файлов, заданное в операнде FILES оператора SORT или MERGE; m — число рабочих устройств, заданное в операнде WORK оператора SORT.

Задание на выполнение

Оператор	Пояснения
// JOB	Начало задания
// TLBL	Информация о метках для файлов на магнитных лентах с системными метками
// DLBL	Информация о метках для файлов на дисках
// EXTENT	
// ASSGN	Назначение периферийных устройств

```
// LBLTYP TAPE        Если файлы на магнитных лентах с системными метками, резервирование области для обработки меток
// EXEC SORT          Выполнение программы
Управляющие операторы программы (SORT или MERGE, RECORD, MODS, INPFIL, OUTFIL, OPTION, END)
/&                    Конец задания
```

Операторы SORT (MERGE), RECORD, END являются обязательными, а MODS, INPFIL, OUTFIL, OPTION могут отсутствовать.

7.6.3. УПРАВЛЯЮЩИЕ ОПЕРАТОРЫ

Общие правила записи. Структура:

⟨признак⟩ ⟨код⟩ ⟨операнды⟩

В качестве признака используется пробел. Составные части оператора разделяются не менее чем одним пробелом. Если программа сортировки используется в качестве подпрограммы, между составными частями оператора должен быть один пробел. Для записи оператора могут использоваться перфокарты продолжения. При использовании перфокарт продолжения нужно следить за тем, чтобы общее количество перфокарт с управляющими операторами программы не превышало 25.

Оператор SORT предназначен для описания управляющих полей при выполнении операции сортировки, указания числа вводных и рабочих файлов и режима работы программы с созданием контрольных точек. Формат:

$$\text{SORT } \begin{cases} \text{FIELDS}=(p_1,m_1,f_1,s_1,\ldots) \\ \text{FIELDS}=(p_1,m_1,s_1,\ldots),\text{FORMAT}=xx \end{cases}, \text{ WORK}=n[,\ \text{SIZE}=n][,\ \text{FILES}=n][,\text{CHKPT}]$$

где FIELDS — описание управляющих полей для операции сортировки:

p_1, m_1, s_1 — см. 7.5.3, операнд FIELDS оператора SORT;

f_1 — формат данных, содержащихся в i-м управляющем поле:

BI (CH), ZD, PD, FI, FL — см. 7.5.3, параметр f_1 операнда FIELDS оператора SORT,

SC — для данных, содержащих русский текст, или данных в специальном коде;

FORMAT — описание формата управляющих полей, если он одинаков для всех полей. Совпадает со значением для f_1;

WORK — количество рабочих устройств для операции сортировки:

n = 3, ..., 9 в случае использования магнитных лент,

n = 1, ..., 8 в случае использования дисков;

SIZE — общее количество записей во вводных файлах (указывается с избытком, если неизвестно точное количество);

CHKPT — режим работы программы, при котором создаются контрольные точки. Организацию повторного запуска с контрольной точки см. в 2.2.5;

FILES — число вводных файлов, предназначенных для сортировки ($1 \leqslant n \leqslant 9$).

Оператор MERGE предназначен для описания управляющих полей при выполнении операции слияния и указания порядка слияния. Формат:

$$\text{MERGE}\ \begin{Bmatrix}\text{FIELDS}=(p_1,m_1,f_1,s_1,\ldots)\\ \text{FIELDS}=(p_i,m_i,s_i,\ldots),\text{FORMAT}=xx\end{Bmatrix},\begin{Bmatrix}\text{FILES}=n\\ \text{ORDER}=n\end{Bmatrix}$$

где FIELDS — описание управляющих полей для операции слияния (см. оператор SORT);

FORMAT — формат управляющих полей (см. оператор SORT);

ORDER — порядок слияния (количество объединяемых файлов: $1 \leqslant n \leqslant 8$);

Оператор RECORD предназначен для описания формата и длин обрабатываемых записей. Формат:

$$\text{RECORD}\ \text{TYPE}=\begin{Bmatrix}F\\ V\end{Bmatrix},\text{LENGTH}=\begin{Bmatrix}l_1,l_2,l_3\\ l_1,l_2,l_3,l_4,l_5\end{Bmatrix}$$

где TYPE — формат обрабатываемых записей:

F — записи фиксированной длины

V — записи переменной длины;

LENGTH — описание длин обрабатываемых записей:

l_1 — длина записей вводного файла;

l_2 — длина записей рабочего файла;

l_3 — длина записей выводного файла;

l_4 — минимальная длина записей вводного файла для записей переменной длины;

l_5 — средняя длина записей вводного файла для записей переменной длины.

Для записей переменной длины параметры l_1, l_2, l_3 означают максимальную длину записей.

Оператор INPFIL предназначен для описания структуры вводных файлов и указания некоторых режимов работы программы, связанных со считыванием вводных файлов, а также действий по установке магнитных лент при открытии и закрытии вводных файлов на магнитных лентах. Формат:

$$\text{INPFIL}\ [\text{EXIT}][,\text{BLKSIZE}=n][,\text{BYPASS}]\left[,\text{VOLUME}=\begin{Bmatrix}n\\ (n_1,n_2,\ldots)\end{Bmatrix}\right]$$

$$\left[,\text{OPEN}=\begin{Bmatrix}\text{RWD}\\ \text{NORWD}\end{Bmatrix}\right]\left[,\text{CLOSE}=\begin{Bmatrix}\text{RWD}\\ \text{UNLD}\\ \text{NORWD}\end{Bmatrix}\right]$$

где EXIT — режим работы программы, при котором вводной файл считывается подпрограммой пользователя для выхода E15;

BLKSIZE — длина блока вводного файла для записей фиксированной длины или максимальная длина блока для записей переменной длины;

BYPASS — пропуск блоков вводного файла, при считывании которых возникают ошибки чтения или ошибки длины;

VOLUME — количество томов, занимаемых вводными файлами: n — количество томов единственного вводного файла для операции сортировки или слияния; n_i — количество томов i-го вводного файла для операции сортировки или слияния ($1 \leqslant i \leqslant 8$ для операции слияния и $1 \leqslant i \leqslant 9$ для операции сортировки);

OPEN — действия по установке магнитных лент перед открытием первого тома каждого вводного файла: RWD — перемотка ленты к точке загрузки; NORWD — перемотка ленты не выполняется.

CLOSE — действия по установке магнитных лент после закрытия последнего тома каждого вводного файла: RWD — перемотка ленты к точке загрузки; UNLD — перемотка и разгрузка ленты; NORWD — перемотка ленты не выполняется.

Оператор OUTFIL предназначен для указания режимов работы программы, связанных с формированием выводного файла, и действий по установке магнитной ленты при открытии и закрытии выводного файла на магнитных лентах. Формат:

$$\text{OUTFIL [EXIT][,BLKSIZE=n][,NOTPMK]}\left[\text{,OPEN}=\left\{\begin{matrix}\text{RWD}\\ \text{NORWD}\end{matrix}\right\}\right]$$

$$\left[\text{,CLOSE}=\left\{\begin{matrix}\text{RWD}\\ \text{UNLD}\\ \text{NORWD}\end{matrix}\right\}\right]$$

где EXIT — режим работы программы, при котором выводной файл записывается подпрограммой для выхода E35;

BLKSIZE — длина блоков выводного файла для записей фиксированной длины или максимальная длина блоков выводного файла для записей переменной длины;

NOTPMK — режим работы программы, при котором в начале каждого тома выводного файла без меток не должна записываться ленточная марка;

OPEN — аналогичен операнду OPEN оператора INPFIL;

CLOSE — аналогичен операнду CLOSE оператора INPFIL.

Оператор OPTION предназначен для указания общих режимов работы программы, типа меток обрабатываемых файлов и размера памяти, отведенной для выполнения программы. Формат:

$$\text{OPTION}\left[\text{PRINT}=\left\{\begin{matrix}\text{ALL}\\ \text{CRITICAL}\\ \text{NONE}\end{matrix}\right\}\right][\text{,LABEL}=(\text{вывод, ввод}_1,\ldots,\text{ввод}_n,$$

$$\text{рабочий})][\text{,STORAGE}=n][\text{,VERIFY}][\text{,ALTWK}]$$

$$[\text{,KEYLEN}=n]\left[\text{,ADDROUT}=\left\{\begin{matrix}\text{A}\\ \text{D}\end{matrix}\right\}\right][\text{,CALCAREA}]$$

где PRINT — режим печати сообщений:

ALL — должны печататься все сообщения программы;

CRITICAL — должны печататься только критические сообщения программы, которые приводят к снятию задания;

NONE — ни одно из сообщений программы не должно печататься;

LABEL — тип меток обрабатываемых файлов (выводного, i-го вводного, рабочих): S — системные метки, N — метки пользователя и нестандартные метки, U — файлы без меток;

STORAGE — объем основной памяти в байтах, отведенной для программы SORT и подпрограмм пользователя, загружаемых программой SORT ($n \geqslant 10K$);

VERIFY — режим проверки каждого блока выводного файла на дисках;

ALTWK — использование сменной ленты для рабочих файлов на магнитных лентах;

KEYLEN — размер (в байтах) области ключа, предшествующего каждой записи вводного файла на дисках ($1 \leqslant n \leqslant 255$);

ADDROUT — режим работы программы, при котором производится сортировка адресов вводного файла на дисках. Выводной файл содержит: A — адреса записей вводного файла в порядке, соответствующем порядку следования рассортированных записей; D — адреса и управляющие поля записей вводного файла в порядке, соответствующем порядку следования рассортированных записей;

CALCAREA — режим работы программы, при котором производится подсчет размера области на дисках, требуемой для рабочих файлов при выполнении заданной операции сортировки.

Оператор MODS предназначен для описания выходов к подпрограммам пользователя, используемым для этапов сортировки-слияния. Формат:

MODS PHn = (<имя фазы>, <инф. для загрузки>,En1, ...,Enm)

где PHn — описание выходов к подпрограммам пользователя, используемых для указанного этапа сортировки-слияния. n = 1, 2, 3 соответственно для этапа внутренней сортировки, внешней сортировки и слияния;

имя фазы — имя фазы в библиотеке абсолютных модулей, в которую входят подпрограммы пользователя, обслуживающие выходы данного этапа;

инф. для загрузки — абсолютный адрес загрузки фазы в основную память (в десятичной системе счисления) или длина фазы в байтах, если фаза самоперемещающаяся. Во втором случае перед длиной указывается буква L;

Enm — номер выхода этапа программы сортировки, используемый для подключения подпрограммы пользователя: E11, E15, E17, E18 для внутренней сортировки, E21, E25 для внешней сортировки, E31, E32, E35, E37, E38, E39 для окончательного слияния.

Оператор END предназначен для указания конца управляющих операторов. Операндов не имеет.

7.6.4. ИСПОЛЬЗОВАНИЕ ПРОГРАММЫ СОРТИРОВКИ В КАЧЕСТВЕ ПОДПРОГРАММЫ

Программа сортировки может быть вызвана любой проблемной программой в качестве подпрограммы. Для вызова программы сортировки может быть использована макрокоманда LOAD. Перед передачей управления из проблемной программы в программу сортировки необходимо в регистры занести следующую информацию:

R1 — адрес списка параметров,
R13 — адрес области для сохранения регистров,
R15 — адрес точки входа в программу сортировки,
R14 — адрес возврата из программы сортировки.

Структура списка параметров. Список параметров состоит из 10 полных слов, содержащих следующую информацию:

адрес оператора SORT или MERGE;
адрес оператора RECORD;
адрес оператора INPFIL;
адрес оператора OUTFIL;
адрес оператора OPTION;
адрес оператора MODS;
адрес таблицы переходов для предварительно загруженных подпрограмм пользователя этапа внутренней сортировки;
адрес таблицы переходов для предварительно загруженных подпрограмм пользователя этапа внешней сортировки;
адрес таблицы переходов для предварительно загруженных подпрограмм пользователя этапа слияния;
адрес кода возврата.

Информация списка параметров должна подготавливаться в указанном порядке. Вместо адресов неиспользуемых операторов указываются нули.

Значения кода возврата (подготавливаются программой сортировки):

X'00' при нормальном завершении сортировки,
X'16' при ненормальном завершении сортировки.

7.6.5. ВЫХОДЫ К ПОДПРОГРАММАМ ПОЛЬЗОВАТЕЛЯ

Назначение и функции. Выходы программы сортировки-слияния обеспечивают подключение в определенных точках программы подпрограмм, написанных пользователем и предназначенных для выполнения некоторых дополнительных функций. Выходы предусмотрены для этапов внутренней и внешней сортировки и слияния. Каждый выход идентифицируется трехзначным обозначением Eij (i — номер этапа, j — обозначение выполняемой функции) и предназначен для выполнения одной или нескольких конкретных функций. Функции выходов приведены в табл. 7.12.

Таблица 7.12

Функции выходов программы сортировки на магнитных лентах и дисках

Функции	Внутренняя сортировка				Внешняя сортировка			Слияние					
	E11	E15	E17	E18	E21	E25	E27	E31	E32	E35	E37	E38	E39
Открытие вводных файлов с нестандартными метками или метками пользователя	×							×					
Открытие рабочих файлов					×								
Открытие файлов, используемых подпрограммами пользователя	×				×			×					
Открытие выводного файла с нестандартными метками или метками пользователя								×					
Закрытие вводных файлов с нестандартными метками или метками пользователя	×		×								×		
Закрытие выводного файла с нестандартными метками или метками пользователя											×		
Закрытие файлов, используемых подпрограммами пользователя	×						×				×		
Чтение вводных файлов		×											
Запись выводного файла										×			
Создание контрольных точек	×											×	
Вставка и (или) удаление записей		×				×				×			
Изменение записей вводных файлов, в том числе и управляющих полей		×							×	×			
Изменение записей выводного файла										×			
Удаление и (или) укорачивание записей		×								×			
Суммирование записей						×				×			
Обработка ошибок чтения вводных файлов				×								×	
Обработка ошибок записи выводного файла													×

Связь с подпрограммами пользователя осуществляется посредством регистров, в которые перед передачей управления к подпрограммам пользователя программа сортировки заносит следующую информацию:

1 — адрес списка параметров. Список параметров состоит из 4 слов. Значения параметров приведены в табл. 7.13;

14 — адрес возврата в программу сортировки;

15 — адрес таблицы перехода для подпрограмм выходов данного этапа сортировки.

Таблица 7.13

Параметры, передаваемые подпрограммам пользователя для выходов SORT

Выход	Номер параметра	Значение
E11, E21, E31	1	Адрес кода действия (табл. 7.14)
	2	Адрес таблицы информации об устройствах
	3	Адрес шифра логического устройства, для которого нужно обработать конечную метку
	4	Адрес шифра логического устройства, для которого нужно обработать начальную метку
E15	1	Адрес сортируемой записи
	2	Адрес кода действия (см. табл. 7.14)
E25	1	Адрес записи, которая должна быть переслана в область вывода
	2	Адрес предыдущей записи в области вывода
	3	Адрес кода действия (см. табл. 7.14)
E35	1	Адрес записи, которая должна быть переслана в область вывода
	2	Адрес предыдущей записи в области вывода
	3	Адрес кода действия (см. табл. 7.14)
	4	Адрес слова для подавления проверки последовательности
E17, E27, E37		Нет параметров
E18, E38	1	Адрес ССВ для файла, при считывании которого обнаружена ошибка
	2	Адрес канальной программы, при выполнении которой обнаружена ошибка
	3	Адрес кода действия (см. табл. 7.14)
E39	1	Адрес ССВ для выводного файла, при выводе которого обнаружена ошибка
	2	Адрес канальной программы, при выполнении которой обнаружена ошибка
E32	1	Адрес очередной записи для слияния

Таблица 7.14

Значения кодов действия

Выход	Значение кода действия	Пояснения
E15, E25, E35	X'00'	Обработать запись
	X'04'	Удалить запись
	X'08'	Закрыть выход
	X'0C'	Вставить запись
E18, E38	X'00'	Удалить блок
	X'04'	Включить блок
	X'08'	Снять задание
E21	X'00'	Закрыть выход
	X'04'	Продолжать активизировать выход
	X'08'	Закрыть выход после выполнения повторного запуска с контрольной точки
	X'0C'	Продолжать активизировать выход после выполнения повторного запуска с контрольной точки

ГЛАВА 8

ПРОТОКОЛИРОВАНИЕ СОСТОЯНИЙ И СОБЫТИЙ

Данная глава содержит информацию по компонентам ДОС ЕС, обеспечивающим возможность протоколирования состояния памяти (Генератор программ разгрузки) и состояния информации о метках используемых файлов (Отображение цилиндра меток), а также протоколирования событий вычислительного процесса, представляющих интерес для программистов (Протоколирование событий) и технического персонала ЭВМ (Хроникер и Статистика тома). Для этих же целей могут быть использованы и отдельные компоненты системы, описанные в предыдущих главах: сервисные функции печати Библиотекаря (гл. 4), функции разгрузки и отображения Отладки (гл. 5). Для сбора учетной информации о загрузке ЭВМ могут использоваться средства учета заданий (п. 8.6.).

8.1. ГЕНЕРАТОР ПРОГРАММ РАЗГРУЗКИ

8.1.1. НАЗНАЧЕНИЕ И ВОЗМОЖНОСТИ

Генератор программ разгрузки — это системная, выполняющаяся в ДОС ЕС программа, которая предназначена для генерации системно-независимых программ разгрузки памяти, отвечающих определенным требованиям пользователя. В качестве носителей системно-независимых программ разгрузки могут использоваться перфокарты или магнитная лента. Генератор программ разгрузки может генерировать два типа программ: программы стандартной разгрузки и программы расширенной разгрузки.

8.1.2. СИСТЕМНО-НЕЗАВИСИМЫЕ ПРОГРАММЫ РАЗГРУЗКИ ПАМЯТИ

Назначение. Программы разгрузки памяти, которые могут выполняться без операционной системы (системно-независимые программы разгрузки) обеспечивают распечатку содержимого основной памяти и регистров. Они удобны в тех случаях, когда аналогичные системные средства (макрокоманды DUMP, PDUMP или режим DUMP) не могут быть использованы из-за неработоспособного состояния системы.

Кроме того, состояние области управляющей программы в момент ошибки может быть зафиксировано только с помощью системно-независимых программ разгрузки, так как при работе системных средств разгрузки это состояние, как правило, изменяется.

Типы программ разгрузки. Программы, выполняющие стандартную или расширенную разгрузку памяти. Эти два типа программ отличаются только результатами своей работы, остальные их свойства могут быть запрошены полностью идентичными. В результат расширенной разгрузки полностью входит результат стандартной разгрузки и отредактированная для удобного чтения информация некоторых таблиц и областей управляющей программы.

Запуск. Системно-независимые программы загружаются в основную память с перфокарт или с магнитной ленты путем выполнения процедуры начальной загрузки с того устройства, на котором находится носитель с текстом программы. Процедура загрузки: подготовить устройство, набрать его адрес на переключателях загрузки, нажать кнопку «Загрузка».

Использование памяти. Любая из сгенерированных программ независимо от обеспечиваемых ею функций использует для своего выполнения область в 640 байт основной памяти.

Адрес загрузки. Адрес, начиная с которого загружается в память для выполнения системно-независимая программа, указывается программистом при ее генерации. Это позволяет иметь программы разгрузки, ориентированные на конфигурацию памяти машины, и управлять местоположением той области, которая отводится для выполнения самой программы разгрузки.

Результаты. В результате выполнения любой из сгенерированных программ разгрузки пользователь получает распечатку содержимого основной памяти, которая может давать наглядную картину состояния любой области памяти, кроме области выполнения самой программы разгрузки (640 байт) и байт — 0—23 основной памяти. Длина распечатываемой области может изменяться от 8К до 16384 К байт. Кроме того, выводится содержимое общих регистров, CSW и CAW, а при расширенной разгрузке выводится также содержимое всех PSW управляющей программы, используемых при обработке прерываний, и значение таймера.

После вывода всех указанных выше результатов программа стандартной разгрузки завершает свою работу, а программа расширенной разгрузки начинает вывод на печать в отредактированном, удобном для чтения формате состояния некоторых таблиц и областей управляющей программы.

При этом для каждого раздела выводятся:

область связи;

расширение области связи;

таблица информации о программе;

таблица информации о подзадачах;

область сохранения раздела, если раздел активен. Если раздел не активен, выводится сообщение PROG NOT ACTIVE;

таблица логических устройств (LUB);

таблица физических устройств (PUB);
список системных логических устройств и логических устройств программиста;
таблица информации об ошибках периферийных устройств;
таблица очереди запросов к устройствам.

8.1.3. СПОСОБ ИСПОЛЬЗОВАНИЯ

Имя программы — DUMPGEN.

Условия использования. Разделы фоновый и переднего плана (10K). Программа самоперемещающаяся.

Логические устройства: SYSRDR, а также см. табл. 8.1.

Таблица 8.1

Устройство	Файл	Назначение
SYSIPT	IJSYSIN	Ввод управляющих операторов
SYSLOG	—	Связь с оператором. Если для SYSLOG назначена не пишущая машинка, задание снимается
SYSPCH	IJSYSPH	Вывод результатов выполнения DUMPGEN, представляющих собой абсолютные модули самоперемещающихся системно-независимых программ разгрузки

Задание на выполнение

Оператор	Пояснения
// JOB	Начало задания
// EXEC DUMPGEN	Выполнение Генератора программ разгрузки
Управляющие операторы	Управляющие операторы, описывающие структуру и режимы работы генерируемой программы разгрузки
/*	Конец шага задания
/&	Конец задания

8.1.4. УПРАВЛЯЮЩИЕ ОПЕРАТОРЫ

Общие правила записи. Структура:

⟨признак⟩ ⟨код⟩⟨операнды⟩

Признак — не менее одного пробела.
Код — ASSGN, OPTN.

Составные части оператора разделяются не менее чем одним пробелом. Для записи оператора используется одна перфокарта.

Оператор определения устройства вывода — ASSGN предназначен для указания физического адреса устройства печати, которое должно использоваться для вывода результатов работы генерируемой программы разгрузки. Формат:

ASSGN SYSLST,X'cuu'

где SYSLST — имя логического устройства для вывода результатов разгрузки;

X'cuu' — адрес устройства печати, которое назначается для SYSLST.

Если оператор ASSGN опущен, то предполагается стандартное назначение X'00E'.

Оператор описания программы разгрузки — OPTN. Операторы OPTN описывают функциональные возможности генерируемой системно-независимой программы, а также задают режим вывода результатов самой программы DUMPGEN. В одном операторе OPTN может быть задан только один операнд, для записи операндов используется ключевой формат. В одном шаге задания может использоваться произвольное количество операторов. Формат:

OPTN операнд

Операнд может принимать одно из значений:

DECKS=m

INTR={NO / YES}

CORE=nK

LOADADR={D'xxxxxxxx' / X'xxxxxx'}

FORMAT={NO / YES}

TAPEIPL={NO / YES}

DECKS — определяет количество экземпляров сгенерированной программы разгрузки, которое должно быть выведено на SYSPCH. Это количество определяется значением m (десятичное число 1—999). Если операнд DECKS опущен, предполагается m = 1.

INTR — определяет возможность вмешательства оператора для указания устройства, на которое должен производиться вывод результатов при выполнении сгенерированной программы. Если INTR=YES, то генерируется программа, которая сразу же после загрузки переходит в состояние ожидания, предоставляя оператору возможность выбрать устройство для вывода результатов. Если оператор нажмет кнопку «Прерывание», то результаты выводятся на устройство, определенное оператором ASSGN. Если же необходимо организовать вывод результатов на другое устройство, то оператор должен в момент ожидания последовательно нажать на этом устройстве кнопки «Не готово» и «Готово». Если INTR опущен или INTR=NO, то вмешательство оператора не обеспечивается, а результаты выводятся на устройство, определенное оператором ASSGN.

CORE — определяет размер области основной памяти (начиная с нулевого адреса), которая должна распечатываться во время выполнения разгрузки, и одновременно адрес загрузки самой системно-независимой программы. n — десятичное число (8—16384). Сама системно-независимая программа загружается в последние 640 байт ука-

занной области. Если значение n является нечетным числом, то оно округляется до ближайшего большего четного числа.

LOADADR — определяет адрес загрузки сгенерированной программы и одновременно область памяти, которая должна распечатываться (в этом случае распечатывается вся память). В качестве LOADADR может быть задан любой десятичный адрес (D'xxxxxxxx') от 128 до 16 766 576 или любой соответствующий адрес в шестнадцатеричном формате (X'xxxxxx').

FORMAT — определяет тип генерируемой системно-независимой программы разгрузки (см. 8.1.2). Если FORMAT = NO или операнд опущен, генерируется программа стандартной разгрузки, если FORMAT = YES, то генерируется программа расширенной разгрузки.

TAPEIPL — обеспечивает генерацию программы, которая может загружаться с магнитной ленты. Операнд TAPEIPL = YES действителен только в том случае, если при выполнении DUMPGEN результат выводится на магнитную ленту. Если TAPEIPL = NO, то генерируется программа, которая может быть загружена для выполнения только с перфокарт. Если TAPEIPL = NO и для SYSPCH назначена магнитная лента, то при выводе абсолютного модуля сгенерированной программы на магнитную ленту перед каждой записью проставляется символ управления карманом типа ASA. Перед выполнением сгенерирования программа должна быть переписана с магнитной ленты на перфокарты.

Примечания:

1. Так как операнды CORE и LOADADR задают адрес загрузки генерируемой программы, то наличие одного из них является обязательным. Все остальные операнды могут отсутствовать.
2. Если заданы и операнд CORE , и операнд LOADADR, то принимается во внимание тот операнд, который был введен последним.
3. Десятичные операнды могут содержать незначащие нули.
4. На одной перфокарте может быть записан только один управляющий оператор.
5. При выводе нескольких экземпляров сгенерированной программы экземпляры отделяются друг от друга пустой перфокартой.

8.2. ОТОБРАЖЕНИЕ ЦИЛИНДРА МЕТОК

8.2.1. НАЗНАЧЕНИЕ И ВОЗМОЖНОСТИ

Программа предназначена для вывода на печать информации о метках файлов, записанной на цилиндр меток по операторам TLBL, DLBL и EXTENT. Информация о метках секретных файлов на дисках (параметр DSF в операторе DLBL) не выводится, при этом никакого предупреждающего сообщения программисту или оператору не выдается.

8.2.2. СПОСОБ ИСПОЛЬЗОВАНИЯ

Имя программы — **LSERV.** Разделы — фоновый и переднего плана (10K). Программа самоперемещающаяся.

Логические устройства: SYSRDR, SYSLST.

Задание на выполнение

Оператор	Пояснения
// JOB	Начало задания
// ASSGN	Назначение периферийных устройств
// EXEC LSERV	Выполнение программы Отображение цилиндра меток
/&	Конец задания

8.2.3. РЕЗУЛЬТАТЫ

На рис. 8.1 приведены все основные заголовки, которые присутствуют в распечатке цилиндра меток; под каждым заголовком указано, какого типа информация за ним следует.

```
1 DOS LABEL CYLINDER DISPLAY
2 SYSRES VOLUME SERIAL NUMBER
3  BG USER LABEL (TEMPORARY) TRACK 0
4     ‹информация о файлах или NONE›
5  BG PARTITION STANDARD LABELS (PERMANENT) TRACK 1
4     ‹информация о файлах или NONE›
3  F2 USER LABEL (TEMPORARY) TRACK 2
4     ‹информация о файлах или NONE›
5  F2 PARTITION STANDARD LABELS (PERMANENT) TRACK 3
4     ‹информация о файлах или NONE›
3  F1 USER LABEL (TEMPORARY) TRACK 4
4     ‹информация о файлах или NONE›
5  F1 PARTITION STANDARD LABELS (PERMANENT) TRACK 5
4     ‹информация о файлах или NONE›
6     STANDARD LABELS (ALL PARTITION—PERMANENT) TRACKS
      6—9 FOR 5052 OR 6—19 FOR 5061
4     ‹информация о файлах или NONE›
7  END OF LABEL CYLINDER DISPLAY
```

Рис. 8.1. Формат распечатки цилиндра меток

Рис. 8.1 содержит следующую информацию:

1 — заголовок распечатки цилиндра меток;

2 — регистрационный номер тома SYSRES;

3 — заголовки временной информации о метках разделов BG, F1, и F2 (TRACK — указывает номер дорожки, на которой расположена информация для соответствующего раздела);

4 — информация о файлах:

CREATION DATE — дата создания;
EXPIRATION DATE — дата истечения срока хранения;
EXTENT INFORMATION — информация об участке;
EXTENT LOWER LIMIT — нижняя граница участка;
EXTENT SEQUENCE NUMBER — порядковый номер участка;
EXTENT TYPE — тип участка;
EXTENT UPPER LIMIT — верхняя граница участка;
FILE INDENTIFIER — индентификация файла;
FILE SERIAL NUMBER — регистрационный номер файла;
FILE TYPE — код;

GENERATION NUMBER — номер поколения;
NUMBER OF TRACKS — количество дорожек;
RELATIVE (TO ZERO) START ADDRESS IN TRACKS — относительный начальный адрес (в дорожках);
RETENTION PERIOD (DAYS) — срок хранения (в днях);
SPLIT CYLINDER — разделенный цилиндр;
SYMBOLIC UNIT — логическое устройство;
VERSION NUMBER — номер варианта поколения;
VOLUME SEQUENCE NUMBER — порядковый номер тома;
VOLUME SERIAL NUMBER — регистрационный номер тома.

NONE присутствует в том случае, когда соответствующая информация о файлах отсутствует;

5 — заголовки постоянной информации о метках разделов BG, F2 и F1 (TRACK — указывает номер дорожки, на которой расположена информация для соответствующего раздела);

6 — заголовок стандартной информации о метках (TRACKS указывает номера дорожек, на которых расположена стандартная информация о метках);

7 — сообщение о конце распечатки цилиндра меток.

8.3. ПРОТОКОЛИРОВАНИЕ СОБЫТИЙ

8.3.1. НАЗНАЧЕНИЕ И ФУНКЦИИ

Программа предназначена для использования при отладке программ пользователя и системных программ. Она выполняет следующие функции:

- Протоколирование прерываний от ввода-вывода. Позволяет проследить активность всех или отдельных периферийных устройств, используемых при выполнении программы. Протоколируются прерывания ввода-вывода и операции SIO для указанных устройств. Информация протокола может использоваться для проверки частей программы, осуществляющих ввод-вывод.
- Протоколирование последовательности вызова фаз. Позволяет проследить последовательность вызова фаз, которые загружаются в память по макрокомандам FETCH и LOAD и по командам SVC 2, SVC 3 из библиотеки абсолютных модулей. Информация протокола может использоваться для контроля вызова фаз при отладке многофазовой программы. Информация о фазах регистрируется в той последовательности, в которой Супервизор загружает эти фазы в память по мере возникновения соответствующих прерываний. При этом регистрируется информация о фазах программ, выполняющихся во всех или в указанных разделах.
- Протоколирование прерываний по SVC. Позволяет проследить все прерывания при обращении к Супервизору, вызванные отдельными или всеми командами SVC, которые выданы программами указанных или всех разделов. Перечень кодов команд SVC (в шестнадцатеричном виде) и функций, выполняемых Супервизором по этим SVC, приведен в табл. 8.2. Протоколирование SVC, которым соответствуют макрокоманды связи с Супервизором,

может быть полезно при отладке программ пользователя, протоколирование прочих SVC — при отладке ДОС ЕС. Если при генерации системы в макрокоманде FOPT был задан параметр PTO = YES, то указанные SVC не протоколируются в том случае, когда занята В-транзитная область.

Таблица 8.2

Коды SVC

Макрокоманда	Код SVC	Функция
EXCP	0	Выполнить канальную программу
FETCH	1	Загрузить и выполнить любую фазу
—	2	Загрузить и выполнить В-транзит
—	3	Загрузить и выполнить А-транзит
LOAD	4	Загрузить фазу
MVCOM	5	Модифицировать область связи
CANCEL	6	Снять задание (задачу)
WAIT	7	Ждать
—	8	Передать управление из В-транзита в проблемную программу
LBRET	9	Передать управление из проблемной программы В-транзиту после SVC 8
SETIME	A	Установить интервал времени
—	B	Выход из В-транзита
—	C	Установить в 0 флажки байта 57 области связи
—	D	Установить в 1 флажки байта 57 области связи
EOJ	E	Завершить выполнение программы
—	F	Та же, что и у SVC 0, но, если очередь к каналам полна, запрос игнорируется
STXIT(PC)	10	Установить связь с подпрограммой пользователя при программном прерывании
EXIT(PC)	11	Выход из подпрограммы обработки программных прерываний

Продолжение табл. 8.2

Макрокоманда	Код SVC	Функция
STXIT(IT)	12	Установить связь с подпрограммой пользователя при прерывании по таймеру
EXIT(IT)	18	Выход из подпрограммы пользователя обработки прерываний по таймеру
STXIT(OC)	14	Установить связь с подпрограммой пользователя при внешних прерываниях или прерываниях по кнопке «Внимание» (связь с оператором)
EXIT(OC)	15	Выход из подпрограммы пользователя для связи с оператором
—	16	По первой команде SVC 16 выдавшая ее программа захватывает систему, запрещая мультипрограммную работу Повторная выдача освобождает систему, разрешая мультипрограммную работу
—	17	Получить адрес загрузки фазы
SETIME	18	Установить связь с TECB пользователя и установить интервал времени
—	19	Выдать команду HIO для устройства телеобработки
—	1A	Проверить адреса границ области на принадлежность разделу
—	1B	Специальный режим выдачи команды HIO для устройств телеобработки
—	1C—22	Резервируются для дальнейших расширений
—	23	Защитить дорожку
FREE	24	Освободить дорожку
STXIT(AB)	25	Установить связь с подпрограммой пользователя при ненормальном завершении выполнения задачи
ATTACH	26	Подключить подзадачу
DETACH	27	Отключить подзадачу
POST	28	Информировать о свершении события
DEQ	29	Освободить ресурс

Продолжение табл. 8.2

Макрокоманда	Код SVC	Функция
ENQ	2A	Защитить ресурс
—	2B	Записать запись SDR в регистрационный файл или скорректировать запись SDR в регистрационном файле
—	2C	Записать запись OBR в регистрационный файл
—	2D	Резервируется для дальнейших расширений
—	2E	Обеспечить программе проверки устройств (OLTEP) возможность работы в состоянии «Супервизор»
—	2F—32	Резервируются для дальнейших расширений
—	33	Предоставить длину указанной фазы программе проверки устройств (OLTEP)
—	34	Предоставить данные о состоянии очереди запросов к устройствам ввода-вывода

• Разгрузка транзитных областей. Эта функция позволяет получить состояние общих регистров и транзитных областей в момент возникновения прерывания по ошибке в программе, имеющего место в любом разделе.

8.3.2. СПОСОБ ИСПОЛЬЗОВАНИЯ

Имя программы — PDAID, PDLIST. PDAID — программа Протоколирование событий, PDLIST — программа вывода на печать результатов выполнения PDAID, записанных на магнитную ленту.

Условия использования. Разделы фоновый и переднего плана (10K). Программа самоперемещающаяся.

Протоколирование событий обеспечивается совместными усилиями Супервизора и программы PDAID. Потенциальная возможность протоколирования событий закладывается в Супервизоре во время генерации, если в макрокомаде FOPT задан параметр CE: резервируется область для работы PDAID (область CE) и в программы Супервизора для обработки соответствующих прерываний включается средство регистрации событий и связи с PDAID. Программа PDAID активизируется по оператору //EXEC PDAID и при своем выполнении обеспечивает настройку системы на тот режим протоколирования, который запрашивает программист с помощью управляющей информации. После завершения настройки системы на режим протоколирования раздел, в котором выполнялась PDAID, становится свободным

и может использоваться для выполнения проблемных программ в режиме протоколирования событий.

Логические устройства: SYSRDR, а также см. табл. 8.3.

Таблица 8.3

Устройство	Файл	Назначение
SYSIPT	IJSYSIN	Ввод управляющей информации для PDAID
SYSLST	IJSYSLS	Вывод результатов
SYSLOG	—	Связь с оператором, ввод управляющей информации для PDAID
SYS005	IJSYS05	Магнитная лента с результатами выполнения программы PDAID. Лента должна быть предварительно установлена к началу файла

Примечание. Устройство для вывода результатов выполнения программы PDAID указывается в управляющей информации для PDAID.

Задание на выполнение PDAID

Оператор	Пояснения
// JOB	Начало задания
// EXEC PDAID	Выполнение программы Протоколирование событий для настройки системы на режим протоколирования
/&	Конец задания

После того как программа Протоколирование событий вызвана для выполнения, на SYSLOG выводится сообщение 4C10D PDAID=, в ответ на которое оператор должен ввести одно из следующих значений: FT, IT, GT, TD, XX, KT (см. табл. 8.4).

Задание на выполнение PDLIST

Оператор	Пояснения
// JOB	Начало задания
// ASSGN	Назначение периферийных устройств
// TLBL	Информация о метках файла на магнитной ленте
// MTC	Установка магнитной ленты
// EXEC PDLIST	Выполнение программы PDLIST
/&	Конец задания

8.3.3. УПРАВЛЯЮЩАЯ ИНФОРМАЦИЯ

Управляющая информация, необходимая для запроса конкретных функций протоколирования, может вводиться с SYSIPT или с SYSLOG и представляет собой последовательность параметров, имеющих ключевой формат. При вводе с SYSIPT параметры в ключевом формате записываются на перфокартах и отделяются один от другого запятыми. При вводе с SYSLOG также используется ключевой формат, однако в этом случае ключевое слово и знак равенства выводятся на SYSLOG программой PDAID, а оператор должен ввести с SYSLOG ответ, представляющий собой значение параметра.

Сводные данные по структуре управляющей информации и некоторые рекомендации по использованию приводятся в табл. 8.4.

Таблица 8.4

Управляющая информация для программы PDAID

Параметр		Значение	Стандартное значение	Пояснение
Ключевое слово	Значение параметра			
PDAID=	IT	Функция IT (протоколирование прерываний от ввода-вывода)	Ввод управляющей информации с SYSIPT	Обязательный параметр
	FT	Функция FT (протоколирование вызовов фаз)		
	GT	Функция GT (протоколирование SVC)		
	TD	Функция TD (разгрузка транзитных областей)		
	XX	Отмена режима протоколирования		
	KT	Ввод управляющей информации с SYSIPT		
OUTPUT DEVICE=	cuu X'cuu'	Физический адрес устройства для вывода протокола (c — канал, uu — устройство)	Протокол будет помещен в основную память	1. Параметры OUTPUT DEVICE и AAA исключают друг друга 2. Можно использовать только один параметр OUTPUT DEVICE
	KT	См. примечания 2 и 4		
	GO	См. примечание 3		
AAA=	X'*llll*', X'hhhhhh'	Границы области памяти, в которую должен быть помещен протокол (X'*llll*' — начало, X'hhhhhh' — конец)	Протокол будет помещен в основную память в область CE	1. Параметры OUTPUT DEVICE и AAA исключают друг друга. 2. Можно использовать только один параметр AAA
	KT	См. примечания 2 и 4		
	GO	См. примечание 3		

Продолжение табл. 8.4

Параметр		Значение	Стандартное значение	Пояснение
Ключевое слово	Значение параметра			
TRACE PARTITION=	SP	Протоколируются прерывания в области Супервизора	Протоколируются прерывания в области Супервизора и во всех трех разделах	1. Параметр допустим только для функций FT и GT 2. Можно использовать не более трех параметров TRACE PARTITION
	BG	Протоколируются прерывания в фоновом разделе		
	F2	Протоколируются прерывания во втором разделе переднего плана		
	F1	Протоколируются прерывания в первом разделе переднего плана		
	KT	См. примечания 2 и 4		
	GO	См. примечание 3		
IGNORE DEVICE=	cuu X'cuu'	Физический адрес устройства, прерывания от которого не должны протоколироваться для функции IT	Протоколируются прерывания от всех устройств	1. Можно использовать не более трех параметров IGNORE DEVICE 2 Параметры IGNORE DEVICE и TRACE DEVICE исключают друг друга
	KT	См. примечания 2 и 4		
	GO	См. примечание 3		
TRACE DEVICE=	cuu X'cuu'	Физический адрес устройства, прерывания от которого должны протоколироваться для функции IT	Протоколируются прерывания от всех устройств	1. Можно использовать не более трех параметров TRACE DEVICE 2 Параметры TRACE DEVICE и IGNORE DEVICE исключают друг друга
	KT	См. примечания 2 и 4		
	GO	См. примечание 3		

Продолжение табл. 8.4

Параметр		Значение	Стандартное значение	Пояснения
Ключевое слово	Значение параметра			
IGNORE SVC=	nn	Код SVC, прерывания по которому не должны протоколироваться для функции GT	Протоколируются прерывания по всем SVC	1. Можно использовать не более шести параметров IGNORE SVC 2. Параметры IGNORE SVC и TRACE SVC исключают друг друга
	KT	См. примечания 2 и 4		
	GO	См. примечание 3		
TRACE SVC=	nn	Код SVC, прерывания по которому должны протоколироваться для функции GT	Протоколируются прерывания по всем SVC	1. Можно использовать не более шести параметров TRACE SVC 2. Параметры TRACE SVC и IGNORE SVC исключают друг друга
	KT	См. примечания 2 и 4		
	GO	См. примечание 3		
—	GO	Признак конца управляющей информации при вводе ее с SYSIPT		См. примечание 5

Примечания:

1. При вводе управляющей информации с SYSLOG запросы с ключевыми словами печатаются в той последовательности, в которой они перечислены в табл. 8.4.

2. Параметр KT представляет собой директиву KT и может быть введен только с SYSLOG как ответ на ключевое слово. Для всех ключевых слов (кроме PDAID) ответ KT означает, что предполагается стандартное значение параметра.

3. Параметр GO может быть введен с SYSLOG только как ответ на ключевое слово. Ответ GO на первый запрос данного ключевого слова означает, что для данного параметра и всех последующих предполагаются стандартные значения. Ответ GO на любой повторный запрос означает, что задание параметров по данному ключевому слову закончено, а для всех последующих параметров предполагаются стандартные значения. В любом случае ввод управляющей информации считается законченным.

4. Ответ KT на последнее ключевое слово из последовательности, указанной в табл. 8.4, означает, что закончен ввод управляющей информации для PDAID.

5. При вводе управляющей информации с SYSIPT GO представляет собой не ключевое слово, а самостоятельный параметр. Если он используется, то должен быть последним параметром во всей управляющей информации для PDAID.

Если управляющая информация должна вводиться с SYSIPT (см. табл. 8.4), то при подготовке перфокарт с управляющей информацией следует иметь в виду следующее:

для записи управляющей информации могут использоваться все колонки перфокарты;

на одной перфокарте могут находиться несколько параметров, отделенных один от другого запятыми;

на одной перфокарте должно находиться целое число параметров;

пробел является признаком конца управляющей информации на данной перфокарте;

параметр GO является признаком конца управляющей информации для PDAID;

если параметр GO опущен, то конец управляющей информации определяется по оператору /*, который должен обязательно присутствовать в качестве последнего оператора входного потока для PDAID;

может быть опущен любой параметр, кроме параметра PDAID = функция;

параметры на карте с управляющей информацией могут располагаться в любой последовательности;

если какой-то из параметров опущен, принимается стандартное значение (см. табл. 8.4).

8.3.4. РЕЗУЛЬТАТЫ

Результаты выполнения функций FT, IT, GT в зависимости от запроса могут быть либо выведены на указанное параметром OUTPUT DEVICE устройство, либо помещены в память (параметр AAA). В качестве устройства вывода может использоваться либо печать, либо магнитная лента. Если протокол записан в память, то он может быть распечатан с помощью системно-независимой программы разгрузки памяти (см. 8.1). В качестве области вывода может быть использована область CE, адрес которой хранится в расширении области связи и может быть найден в протоколе системно-независимой программы рас-

ширенной разгрузки памяти. Сама область CE распределяется следующим образом:

CE + X'B0' — протокол функции FT;
CE + X'D4' — протокол функции IT;
CE + X'E4' — протокол функции GT.

Протоколы выполнения программы PDAID описаны в 8.3.5.

Если при выполнении функций PDAID протоколирование событий выполнялось на магнитную ленту, то распечатать этот протокол можно, выполнив программу PDLIST. Магнитная лента с протоколом должна быть установлена на накопитель, назначенный для SYS005, и подведена к началу информации (PDLIST не выполняет операций по установке магнитной ленты). Протокол выводится на SYSLST в таком же формате, как и при непосредственном выводе на печать.

8.3.5. ПРОТОКОЛЫ

Протоколы на SYSLST. Знаком «x» в протоколах обозначается шестнадцатеричный символ. Протокол функции FT содержит:

nnnnnnnn	Имя фазы	
xx	Код SVC	
C-xxxxxx	C	— признак адреса, xxxxxx — адрес SVC
Px	P	— признак ключа, x — ключ программы:
	1	— программа BG
	2	— программа F2
	3	— программа F1
	7—F	— подзадачи
	0,4—6	— Супервизор
L—xxxxxx	L — признак, xxxxxx — адрес загрузки	
E—xxxxxx	E — признак, xxxxxx — адрес входа.	

Адрес загрузки и адрес входа содержат X'FFFFFF', если фаза не найдена. Если после имени фазы присутствует символ «*», это означает, что потеряна информация не менее чем об одном последующем вызове.

Примечание. Потеря информации может произойти при стечении следующих обстоятельств: область вывода, в которой накапливается блок данных для вывода протокола, заполнена; устройство печати занято; произошло событие, подлежащее протоколированию.

Протокол функции IT при прерывании от ввода-вывода содержит PSW и CSW, соответствующие прерыванию. Протокол функции IT при запуске операции ввода-вывода содержит CSW и следующие данные:

PSW	Признак
x	Код условия
000	Всегда нули
xxxx	Адрес устройства
xxxxxxxx	Адрес CCB

Если после признака PSW присутствует символ «*», это означает, что произошла потеря информации не менее чем об одном последующем прерывании (см. примечание).

Протокол функции GT содержит:

PSW — xx...x	PSW, соответствующее прерыванию. Если вместо признака PSW выводится *SW это означает, что произошла потеря информации не менее чем об одном последующем прерывании (см. примечание)
0 — xxxxxxxx	Содержимое регистра 0
1 — xxxxxxxx	Содержимое регистра 1
N — xx	N — признак, xx — код SVC
T — xx	T — признак, xx — ключ защиты памяти

Протокол функции TD включает: содержимое общих регистров; содержимое байт 0—143 основной памяти, содержимое B-транзитной области и содержимое A-транзитной области.

Вся информация функции TD выводится в шестнадцатеричном виде.

Протоколы в памяти. Протокол функции FT содержит записи по 19 байт. Структура записи:

Байты	Значение
8	Имя фазы
1	Код SVC
3	Адрес SVC
1	Ключ программы
3	Адрес загрузки
3	Адрес входа

Адрес загрузки и адрес входа содержит X'FFFFFF', если фаза не найдена

Протокол функции IT при прерывании от ввода-вывода содержит записи по 16 байт. Каждая запись содержит PSW и CSW, соответствующие прерыванию. Протокол функции IT при запуске операции ввода-вывода содержит записи по 16 байт. Структура записи:

Байты	Значение
1	Первые 4 бита — код условия
1	Нули
2	Адрес устройства
4	Адрес CCB
8	CSW

Протокол функции GT содержит записи по 16 байт. Структура записи:

Байты	Значение
8	PSW, соответствующее прерыванию
1	Ключ защиты памяти
3	Последние 3 байта регистра 0
4	Содержимое регистра 1

8.4. ХРОНИКЕР

8.4.1. НАЗНАЧЕНИЕ И ФУНКЦИИ

Хроникер предназначен для обработки информации об ошибках периферийных устройств, машинных ошибок и ошибок каналов, накопленных управляющей программой в регистрационном файле на SYSREC в процессе функционирования ДОС ЕС. На основе анализа полученной информации могут быть локализованы устойчивые ошиб-

ки технических средств и установлены причины ошибок; могут быть разработаны предложения по модернизации технических средств, улучшению условий их эксплуатации, изменению режима профилактики.

Хроникер выполняет следующие функции:

- Вывод на печать отредактированной информации регистрационного файла.
- Очистку регистрационного файла.
- Создание исторического файла на основе регистрационного файла.
- Обновление исторического файла на основе регистрационного файла.

Регистрационный файл используется для накопления статистической информации об ошибках оборудования, возникающих во время функционирования ДОС ЕС. Файл содержит записи следующего типа:

SDR — записи, содержащие суммарную информацию об ошибках периферийных устройств;

OBR — записи, содержащие информацию о неисправимых ошибках периферийных устройств;

IPL, EOD — записи, обеспечивающие ведение статистики надежности для моделей «Ряда-2»;

MCRR — записи, содержащие информацию о машинных ошибках и ошибках канала для моделей «Ряда-1»;

MCAR — записи, содержащие информацию о машинных ошибках для моделей «Ряда-2»;

CCH — записи, содержащие информацию об ошибках канала для моделей «Ряда-2».

Для каждого физического устройства, учтенного в системе, в регистрационном файле содержится одна запись SDR. Количество записей OBR, MCRR, MCAR, CCH определяется количеством соответствующих ошибок, возникших при функционировании системы. Одна запись IPL строится для каждого случая выполнения начальной загрузки системы. Одна запись EOD строится при выдаче директивы ROD в конце работы с системой.

8.4.2. СПОСОБ ИСПОЛЬЗОВАНИЯ

Имя программы — EREP.

Условия использования. Разделы фоновый и переднего плана (10K). Требуется редактирование для раздела.

Логические устройства: SYSRDR, а также см. табл. 8.5.

Таблица 8.5

Устройство	Файл	Назначение
SYSIPT	IJSYSIN	Ввод управляющих операторов Хроникера
SYSLST	IJSYSLS	Вывод на печать отредактированной информации регистрационного файла
SYSLOG	—	Связь с оператором и ввод управляющих операторов Хроникера
SYSREC	IJSYSRC	Регистрационный файл
SYS009	EREPNEW	Создаваемый исторический файл
SYS009	EREPUP	Обновляемый исторический файл

Задание на выполнение

Оператор	Пояснения
// JOB	Начало задания
// DLBL // EXTENT	Информация о метках и участках файлов на дисках
// TLBL	Информация о метках файлов на магнитной ленте
// ASSGN	Назначение периферийных устройств
// EXEC EREP	Выполнение Хроникера
Управляющие операторы	Управляющие операторы OPTION если используется режим ввода их с SYSIPT
/*	Конец шага задания. Должен присутствовать, если управляющие операторы вводятся с SYSIPT
/&	Конец задания

8.4.3. УПРАВЛЯЮЩИЕ ОПЕРАТОРЫ

Могут вводиться с SYSIPT и SYSLOG. При вводе с SYSIPT признаком конца управляющих операторов служит оператор /*, при вводе с SYSLOG — директива KT. Признаком конца ввода каждого управляющего оператора с SYSLOG служит также директива KT. При этом оператор, вводимый с SYSLOG, не должен содержать более 80 символов. В самом начале выполнения EREP оператору предоставляется возможность указать источник управляющей информации: на SYSLOG выводится сообщение 3E11D ENTER OPTION SOURCE, C=CARD. S = CONSOLE, N = NONE.

В ответ на сообщение оператор должен указать, с какого устройства будет вводиться управляющая информация:

C — ввод информации с SYSIPT;

S — ввод информации с SYSLOG;

N или KT — выполнение стандартной функции (EDIT).

После ответов C, S, N необходимо ввести директиву KT.

Правила записи. Структура:

⟨признак⟩ ⟨OPTION⟩ ⟨операнд⟩

Признак — не менее одного пробела.

Составные части оператора разделяются не менее чем одним пробелом. Для записи оператора используется одна перфокарта или не более 80 символов при вводе с пишущей машинки. В одном операторе содержится один операнд. В одном шаге задания может быть использовано необходимое количество управляющих операторов. Операнд может принимать одно из следующих значений:

EDIT
CLEAR
HIST, NEW [,2]
HIST [,2]
HIST, UPNEW

EDIT — редактирование и вывод на печать записей регистрационного файла (см. 8.4.4). Записи распечатываются в следующей последовательности: SDR, OBR, CCH, MCAR, MCRR. После вывода на печать для счетчиков различных типов ошибок в записях SDR устанав-

ливаются нулевые значения. Функция EDIT автоматически запрашивается при выполнении функций CLEAR и HIST;

CLEAR — очистка регистрационного файла: записи файла заполняются нулями, формат его сохраняется. Функция используется для очистки файла в случае его переполнения. CLEAR автоматически запрашивает выполнение функции HIST, функция HIST автоматически запрашивает выполнение CLEAR.

HIST, NEW [,2] — создание исторического файла. На основании записей регистрационного файла создается исторический файл на магнитной ленте, представляющий собой выборочную копию регистрационного (копируются записи OBR, CCH, MCAR, IPL, EOD). Если задан параметр 2, то создаются два идентичных исторических файла на двух магнитных лентах, которые последовательно устанавливаются на устройство SYS009. Функция HIST,NEW [,2] автоматически запрашивает функции EDIT и CLEAR.

HIST[,2] — обновление исторического файла на основании записей регистрационного файла: записи OBR, CCH, MCAR, IPL, EOD регистрационного файла копируются в конец обновляемого исторического файла. Если задан параметр 2, то обновляются два исторических файла на двух магнитных лентах, которые последовательно устанавливаются на устройство SYS009. Функция HIST [,2] автоматически запрашивает функции EDIT и CLEAR.

HIST,UPNEW — комплексная функция обновления и создания исторического файла. На основе записей регистрационного файла обновляется исходный исторический файл и строится новый исторический файл.

Магнитные ленты для обновляемого и создаваемого исторических файлов последовательно устанавливаются на устройство SYS009.

8.4.4. ПРОТОКОЛЫ

Общая статистика ошибок периферийных устройств. См. рис 8.2.

```
1 - - STATISTICAL DATA EDITING - - -
*****************************************************S****
2 - - - RECORD ENTRY TYPE-STATISTICAL DATA
3 CHANNEL/UNIT ADDRESS 4 DEVICE TYPE
5 (счетчики ошибок)                    (счетчики ошибок)
*****************************************************
```

Рис. 8.2. Протокол записи SDR

Рис. 8.2 содержит следующую информацию:

1 — общий заголовок суммарной статистики об ошибках периферийных устройств на каждом листе распечатки записей SDR;

2 — признак записи SDR;

3 — адрес физического устройства (и DRIVE для EC-5061);

4 — тип устройства: шифр и характеристики (BURST — монопольный режим в мультиплексном канале, SWITCHABLE — переключаемое устройство);

5 — счетчики ошибок:

ALU — ошибка в арифметико-логическом устройстве;
BUS OUT CHK — ошибка в выходной шине;
CHAN TAG LINE — выбранное состояние;
COMMAND REJECT — команда отвергнута;
CRC/END DATA — ошибка циклического контроля;
DATA CONV CHK — ошибка преобразования данных;
EQUIP CHK — ошибка оборудования;
INTRVN REQD — требуется вмешательство;
LRCR/MTE — ошибка продольного контроля;
MISG ADR MKR — пропуск адресного маркера;
NOISE — помеха;
OVERRUN — переполнение;
R/W VRC — ошибка вертикального контроля;
SEEK CHK — ошибка установки;
SER/DESER — ошибка преобразования;
SKEW — ошибка перекоса;
SKEW REG VRC/ENV — ошибка четности в регистре перехода;
STR — счетчики для UNSP и UNSPB;
TEMPY RDS — количество исправимых ошибок при чтении;
TEMPY WRT — количество исправимых ошибок при записи;
TRK CND — контроль состояния дорожки;
UNSAFE — ошибка в накопителе;
UNUSUAL COMND SEQ — неверная последовательность команд;
WD CT 0 — нулевой счетчик слов.

Статистика неисправимых ошибок периферийных устройств. См. рис. 8.3, с. 230.

Рис. 8.3 содержит следующую информацию:

1 — общий заголовок статистики о неисправимых ошибках периферийных устройств на каждом листе распечатки записей OBR;

2 — имя задания;

3 — дата (DAY — день, YEAR — год);

4 — время (HH — часы, MM — минуты, SS — секунды);

5 — адрес физического устройства (и DRIVE для EC-5061);

6 — тип устройства: шифр и характеристики (BURST — монопольный режим в мультиплексном канале, SWITCHABLE — переключаемое устро ство);

7 — логическое устройство;

8 — регистрационный номер тома;

9 — плотность записи или дисковый адрес;

10 — первое CCW в цепочке (CC — код команды канала, DA — адрес данных, FL — флажки, CT — счетчик);

11 — CCW, при котором произошла ошибка;

12 — слово состояния канала (K — ключ, CA — адрес команды канала, US — состояние устройства, CS — состояние канала, CT — счетчик);

13 — состояние устройства:

ATTENTION — внимание;
STATUS MODIFIER — модификатор;

```
                    1 ---OUTBOARD EDITING---
*****************************************************************************************
2 PROGRAM IDENTITY
                                DAY YEAR              HH  MM  SS
                        3 DATE-xxx  xx        4 TIME-xx  xx  xx
5 CHANNEL/UNIT ADDRESS      6 DEVICE TYPE
7 LOGICAL UNIT              9 MODE или DASD ADDRESS
8 VOLUME I.D.
                  CC    DA     FL      CT
10 FIRST CCW      xx  xxxxxx   xx  xx  xxxx
11 FAILING CCW    xx  xxxxxx   xx  xx  xxxx
                                        K    CA     US  CS   CT
                                  12 CCW xx  xxxxxx  xx  xx  xxxx
13 ---UNIT STATUS---                    14 ---CHANNEL STATUS---
ATTENTION              CHANNEL END      PRGM—CTLD IRPT      CHAN DATA CHECK
STATUS MODIFIER        DEVICE END       INCORRECT LENGTH    CHAN CTL CHECK
CONTROL UNIT END       UNIT CHECK       PROGRAM CHECK       I/F CTL CHECK
BUSY                   UNIT EXCEPTION   PROTECTION CHECK    CHAINING CHECK
15 SENSE BYTE DATA
*****************************************************************************************
```

Рис. 8.3. Протокол записи OBR

CONTROL UNIT END — устройство управления кончило;
BUSY — занято;
CHANNEL END — канал кончил;
DEVICE END — устройство кончило;
UNIT CHECK — ошибка в устройстве;
UNIT EXCEPTION — особый случай;

14 — состояние канала:

PRGM-CTLD IRPT — программно управляемое прерывание;
INCORRECT LENGTH — неправильная длина;
PROGRAM CHECK — ошибка в программе;
PROTECTION CHECK — нарушение защиты;
CHAN DATA CHECK — ошибка данных в канале;
CHAN CTL CHECK — ошибка управления каналом;
I/F CTL CHECK — ошибка управления интерфейсом;
CHAINING CHECK — ошибка в цепочке;

15 — байты уточненного состояния.

Статистика ошибок ЕС-1020 и ЕС-1022. См. рис. 8.4—8.5.

Рис. 8.4 содержит следующую информацию:

1 — общий заголовок распечатки информации о машинных ошибках в начале каждого листа;
2 — номер модели ЭВМ;
3 — имя задания;
4 — дата;
5 — время;
6 — устройства, использовавшиеся в момент ошибки;
7 — адреса физических устройств;
8 — старое PSW для прерываний от схем аппаратурного контроля;
9 — общие регистры;
10 — регистры плавающей запятой;

```
      1  ---MACHINE CHECK EDITING---
2 MODEL              3 PROGRAM IDENTITY
          DAY  YEAR                HH  MM  SS
      4 DATE-xxx  xx           5 TIME-xx  xx  xx
                                  SM  KS  IC  CM
6 I/O  UNITS IN USE AT THE TIME OF FAILURE
7 CHANNEL/UNIT ADDRESSES   SM  KS   IC   CM   IA
8 OLD MACHINE CHECK PSW    xx  xx  xxxx  xx  xxxxxx
9 ---GENERAL PURPOSE REGISTERS---
10 ---FLOATING POINT REGISTERS---
11 LOGOUT BYTES
12        MACHINE CHECK ERROR REGISTER
```

Рис. 8.4. Протокол записи MCRR для машинных ошибок (ЕС-1020, ЕС-1022)

```
      1 ---CHANNEL INBOARD EDITING---
2 MODEL              3 PROGRAM IDENTITY
4 CHANNEL/UNIT ADDRESS
                  DAY   YEAR              HH  MM  SS
        5 DATE-xxx    xx          6 TIME-xx   xx  xx
                    CG    DA      FL        CT
7 FIRST CCW         xx    xxxxxx  xx  xx    xxxx
8 FAILING CCW       xx    xxxxxx  xx  xx    xxxx
                                          K     CA      US  CS  CT
                                    9 CSW xx    xxxxxx  xx  xx  xxxx
10 ---UNIT STATUS---                    11 ---CHANNEL STATUS---
ATTENTION           CHANNEL END         PRGM-CTLD IRPT      CHAN DATA CHECK
STATUS MODIFIER     DEVICE END          INCORRECT LENGTH    CHAN CTL CHECK
CONTROL UNIT END    UNIT CHECK          PROGRAM CHECK       I/F CTL CHECK
BUSY                UNIT EXCEPTION      PROTECTION CHECK    CHAINING CHECK
12 I/O UNITS IN USE AT THE TIME OF FAILURE
13 CHANNEL/UNIT ADDRESSES
14 LOGOUT BYTES или NO LOGOUT INDICATED
15 CATALOG NUMBER или MACHINE CHECK ERROR REGISTER
```

Рис. 8.5. Протокол записи MCRR для ошибок канала (EC-1020, EC-1022)

11 — область аппаратурной регистрации;
12 — регистр ошибок.

Рис. 8.5 содержит следующую информацию:

1 — общий заголовок распечатки информации об ошибках канала в начале каждого листа;
1 — номер модели ЭВМ;
3 — имя задания;
4 — адрес физического устройства;
5 — дата;
6 — время;
7 — первое CCW в цепочке;
8 — CCW, при котором произошла ошибка;
9 — слово состояния канала;
10 — состояние устройства;
11 — состояние канала;
12 — устройства, использовавшиеся в момент ошибки;
13 — адреса физических устройств;
14 — область аппаратурной регистрации;
15 — каталожный номер ошибки, или регистр ошибок, или сообщение о некорректности области регистрации.

Статистика ошибок ЕС-1030. См. рис. 8.6—8.7.

Рис. 8.6 содержит следующую информацию:

1 — общий заголовок распечатки информации о машинных ошибках в начале каждого листа;
2 — номер модели ЭВМ;
3 — имя задания;
4 — дата;

```
1  ···MACHINE CHECK DATA EDITING···
2  MODEL              3  PROGRAM IDENTITY
       DAY   YEAR                    HH  MM  SS
4  DATE-xxx    xx              5  TIME-xx  xx  xx
6  I/O UNITS IN USE AT THE TIME OF FAILURE
7  CHANNEL/UNIT ADDRESSES
                                SM   KS   IC   CM   IA
8  OLD MACHINE CHECK PSW        xx   xx  xxxx  xx  xxxxxx
9  ···GENERAL PURPOSE REGISTERS···
10 ···FLOATING POINT REGISTERS···
11 ···CPU EDITED LOG···
12 ···MACHINE CHECK REGISTER···
13 ···STORAGE CHECK REGISTER···
14 ···MACHINE CHECK LOGOUT BYTES···
```

Рис. 8.6. Протокол записи MCRR для машинных ошибок (ЕС-1030)

```
                1  ---CHANNEL INBOARD EDITING---
2  MODEL        3  PROGRAM IDENTITY
                          DAY   YEAR              HH  MM  SS
                     4  DATE-xxx   xx        5  TIME-xx   xx   xx
6  CHANNEL/UNIT ADDRESS
                     CC    DA      FL      CT
7  FIRST CCW         xx   xxxxxx   xx   xx   xxxx
8  FAILING CCW       xx   xxxxxx   xx   xx   xxxx
                                        K     CA     US   CS   CT
                                   9  CSW   xx   xxxxxx   xx   xx   xxxx
10 ---UNIT STATUS---      11 ---CHANNEL STATUS---
12 I/O UNITS IN USE AT THE TIME OF FAILURE
13 CHANNEL/UNIT ADDRESSES
*******************************************************************************
          14 ---COMMON CHANNEL EDITED LOG---
15                    16                    17          18
-AMS REG 1-   -ACCEPTANCE INSTR TAKTS-   -IRPT TAKTS-   -CTRL LATCH
15 -AMS REG 2-
15 -AMS REG 3-
19 -COMMON CHANNEL LOGOUT BYTES-
*******************************************************************************
        20 ---MULTIPLEXOR CHANNEL EDITED LOG---
        (или  SELECTOR CHANNEL EDITED LOG)
          21                    22                    23
---IRPT BFR REG---        ---SUBCHANNEL REG---        ---IF  ID---
          24                    25                    26
---FLAGS REG---           ---COUNT BYTE REG---        ---CTRL LATCH---
                     27   ---TAKT LATCHES---
28 ---CHANNEL LOGOUT BYTES---
```

Рис. 8,7. Протокол записи MCRR для ошибок канала ЕС-1030

5 — время;
6 — устройства, использовавшиеся в момент ошибки;
7 — адреса физических устройств;
8 — старое PSW для прерываний от схем аппаратурного контроля;
9 — общие регистры;
10 — регистры плавающей запятой;
11 — информация аппаратурной регистрации;
12 — регистр машинных ошибок;
13 — регистр ошибок памяти;
14 — область аппаратурной регистрации.

Рис. 8.7 содержит следующую информацию:

1 — общий заголовок распечатки информации об ошибках канала в начале каждого листа;
2 — номер модели ЭВМ;
3 — имя задания;
4 — дата;
5 — время;
6 — адрес физического устройства;
7 — первое CCW цепочки;
8 — CCW, при котором произошла ошибка;
9 — слово состояния канала;
10 — состояние устройства;
11 — состояние канала;
12 — устройства, использовавшиеся в момент ошибки;
13 — адреса физических устройств;
14 — информация аппаратурной регистрации для общего канала;
15 — РООП 1 (регистр обращения к ОП);
16 — такты приема инструкции;
17 — такты прерывания;
18 — управляющие триггеры;
19 — область аппаратурной регистрации для общего канала;
20 — информация аппаратурной регистрации для мультиплексного или селекторного канала;
21 — РБ (регистр буфера прерываний);
22 — РП (регистр подканала);
23 — идентификатор интерфейса;
24 — РФ (регистр флажков);
25 — РСБ (регистр счета байта);
26 — управляющие триггеры;
27 — тактовые триггеры;
28 — область аппаратурной регистрации.

Статистика ошибок ЕС-1035. См рис. 8.8—8.9.

Рис. 8.8 содержит следующую информацию:

1 — общий заголовок распечатки информации о машинных ошибках в начале каждого листа;
2 — номер модели ЭВМ;
3 — серийный номер;
4 — имя задания;
5 — имя программы;

6 — дата;
7 — время;
8 — старое PSW для прерываний от схем аппаратурного контроля;
9 — код прерывания от схем аппаратурного контроля:

SUB CLASS — подкласс;
INTERRUPT TENSE CODES — момент прерывания;
STORAGE AND PROTECTION ERROR CODES — типы ошибок в памяти;
PSW VALIDITY CODES — биты корректности PSW;
MISC VALIDITY CODES — корректность областей;
REGION CODE — код области;

10 — регистры плавающей запятой;
11 — общие регистры;
12 — управляющие регистры;
13 — область аппаратурной регистрации;
14 — информация расширенной аппаратурной регистрации;
15 — регистр машинных ошибок GRMCKA;
16 — регистр машинных ошибок GRMCKB;
17 — счетчики повторений;
18 — первый и второй регистры повторений;
19 — третий регистр повторений;
20 — четвертый регистр повторений;
21 — микрокоманда;
22 — системный регистр;
23 — регистры локальной памяти;
24 — область расширенной аппаратурной регистрации.

Рис. 8.9 содержит следующую информацию:

1 — общий заголовок распечатки информации об ошибках канала в начале каждого листа;
2 — номер модели ЭВМ;
3 — серийный номер;
4 — имя задания;
5 — имя программы;
6 — заголовок распечатки информации (MPX — мультиплексного канала, SEL — селекторного канала, BLK MPX — блок-мультиплексного канала, INT FILE ADAPTER — интегрированного файлового адаптера);
7 — адрес физического устройства;
8 — дата;
9 — время;
10 — CCW, при котором произошла ошибка;
11 — слово состояния канала;
12 — расширенное CSW;
13 — состояние устройства;
14 — состояние канала;
15 — устройства, использовавшиеся в момент ошибки;
16 — адреса физических устройств;
17 — информация зависимой аппаратурной регистрации;
18 — регистр машинных ошибок GRMCKA;

```
          1 ---MACHINE CHECK DATA EDITING---
2 MODEL   3 SERIAL NUMBER  4 JOB IDENTITY   5 PROGRAM IDENTITY
                    DAY  YEAR          HH  MM  SS
          6 DATE-xxx   xx      7 TIME-xx   xx  xx
                               SM  KS   IG   CM   IA
8 OLD MACHINE CHECK PSW   xx   xx   xxxx   xx   xxxxxx
                  ************************************************************
                9 ---MACHINE CHECK INTERRUPT CODE---
                  ************************************************************
10 ---FLOATING POINT REGISTERS---
11 ---GENERAL PURPOSE REGISTERS---
12 ---GONTROL REGISTERS---
13 ---LOGOUT BYTES---
                                14 ---DEPENDENT LOGOUT---
                  ************************************************************
                                15 ---MACHINE CHECK REGISTER A---
                                16 ---MACHINE CHECK REGISTER B---
17 ---RETRY COUNTS---          18 ---RETRY REGS 1,2---
19 ---RETRY REG 3---           20 ---RETRY REG 4---
21 ---CONTROL WORD---
                                22 ---SYSTEM REGISTER---
                                23 ---LOCAL STORAGE REGISTERS---
24 ---MACHINE CHECK EXTENDED LOGOUT BYTES---
```

Рис. 8.8. Протокол записи MCAR (EC-1035)

```
                1 ---CHANNEL INBOARD EDITING---
2 MODEL  3 SERIAL NUMBER  4 JOB IDENTITY  5 PROGRAM IDENTITY
6 xxx CHANNEL
                                    DAY   YEAR        HH  MM  SS
7 CHANNEL/UNIT ADDRESS  8 DATE-xxx   xx  9 TIME-xx    xx  xx
                  CC    DA     FL     CT
10 FAILING CCW    xx   xxxxxx   xx   xx   xxxx
                                                K    CA     US   CS   CT
                                          11 CSW xx  xxxxxx  xx   xx   xxxx
                        12 ---ECSW---
13 ---UNIT STATUS---          14 ---CHANNEL STATUS---
15 I/O UNITS IN USE AT THE TIME OF FAILURE
16 CHANNEL/UNIT ADDRESSES
17 ---DEPENDENT LOGOUT---
                      18 ---MACHINE CHECK REGISTER A---
                      19 ---MACHINE CHECK REGISTER B---
20 ---RETRY COUNTS---          21 ---RETRY REGS 1,2---
22 ---RETRY REG 3---           23 ---RETRY REG 4---
24 ---CONTROL WORD
                         25 ---SYSTEM REGISTER---
                      26 ---LOCAL STORAGE REGISTERS---
                      27 ---DOCUMENTARY CONSOLE WORD---
28 XXX DEPENDENT LOGOUT BYTES
```

Рис. 8.9. Протокол записи ССН (ЕС-1035)

19 — регистр машинных ошибок GRMCKB;
20 — счетчики повторений;
21 — первый и второй регистры повторений;
22 — третий регистр повторений;
23 — четвертый регистр повторений;
24 — микрокоманда;
25 — системный регистр;
26 — регистры локальной памяти;
27 — документальное консольное слово;
28 — область расширенной аппаратурной регистрации.

Статистика надежности. См. рис. 8.10.

```
          1 ---IPL AND EOD DATA EDITING---
    2 MODEL      3 SERIAL NUMBER
                              DAY   YEAR          HH  MM  SS
                        4 DATE-xxx   xx    5 TIME-xx  xx  xx
6 SUB SYSTEM ID                      CUA
7 REASON CODE                        CHANNEL MAP
8 CHANNEL TYPE ASSGNMT       9 HIGHEST STORAGE ADDR
```

Рис. 8.10. Протокол записей IPL и EOD

Рис. 8.10 содержит следующую информацию:
1 — общий заголовок распечатки информации записей IPL и EOD;
2 — номер модели ЭВМ;
3 — серийный номер процессора;
4 — дата;
5 — время;
6 — идентификатор подсистемы;
7 — код причины;
8 — тип канала;
9 — максимальный адрес памяти.

8.5. СТАТИСТИКА ТОМА

8.5.1. НАЗНАЧЕНИЕ И ФУНКЦИИ

Программы Статистики тома предназначены для инициализации файла Статистики тома, в котором управляющая программа накапливает информацию о работе томов магнитной ленты, и обработки накопленной в этом файле статистической информации. Анализ полученной информации позволяет осуществлять контроль за интенсивностью использования накопителей и качественным состоянием носителей. Программы Статистики тома выполняют следующие функции:

- Создание файла Статистики тома.
- Редактирование и вывод на печать файла Статистики тома.
- Очистка файла Статистики тома (стирание статистических данных без уничтожения физических записей).
- Создание и обновление исторического файла Статистики тома на магнитной ленте.
- Редактирование и вывод на печать исторического файла Статистики тома.
- Получение копии исторического файла Статистики тома на новой магнитной ленте.

Файл Статистики тома используется для накопления следующей информации о работе с томами магнитной ленты:

количестве неисправимых ошибок чтения или записи, возникших при работе с каждым томом магнитной ленты;

количестве исправимых ошибок чтения или записи, возникших при работе с каждым томом магнитной ленты;

количестве некачественных записей;

количестве команд SIO;

плотности записи;

количестве служебных операций, выполненных при восстановлении от ошибок.

Для каждого файла на используемом томе магнитной ленты строится одна запись в файле Статистики тома. Каждая такая запись содержит информацию о работе с томом от момента открытия до момента закрытия файла.

8.5.2. СПОСОБ ИСПОЛЬЗОВАНИЯ

Имя программы — ESTVFMT, ESTVUT. ESTVFMT — создание файла Статистики тома, ESTVUT — обработка файла Статистики тома и исторического файла Статистики тома.

Условия использования. ESTVFMT: фоновый раздел (10K). Требуется редактирование для раздела. ESTVUT: разделы фоновый и переднего плана (6K). Программа самоперемещающаяся.

Логические устройства: SYSRDR, а также см. табл. 8.6.

Таблица 8.6

Устройство	Файл	Назначение
SYSLST	IJSYSLS	Вывод на печать отредактированной информации файла Статистики тома или исторического файла Статистики тома
SYSLOG	—	Связь с оператором и ввод управляющей информации Статистики тома
SYSREC	ESTVFLE	Файл Статистики тома
SYS005	TAPEIN	Вводной (старый) исторический файл Статистики тома
SYS006	TAPEOUT	Выводной (новый) исторический файл Статистики тома

Задание на выполнение ESTVFMT

Оператор	Пояснения
/! JOB	Начало задания
// DLBL	Информация о метках и участке на дисках файла
// EXTENT	Статистики тома (ESTVFLE) Задается обязательно
// ASSGN	Назначение для SYSREC
// EXEC ESTVFMT	Выполнение программы создания файла Статистики тома
/&	Конец задания

Задание на выполнение ESTVUT

Оператор	Пояснения
// JOB	Начало задания
// DLBL	Информация о метках и участках файла ESTVFLE
// EXTENT	
// TLBL	Информация о метках файлов на магнитной ленте
// ASSGN	Назначение периферийных устройств
// EXEC ESTVUT	Выполнение программы обработки файла Статистики тома и исторического файла Статистики тома
/&	Конец задания

8.5.3. УПРАВЛЯЮЩАЯ ИНФОРМАЦИЯ

При работе ESTVFMT управляющая информация не используется.

Управляющая информация для ESTVUT вводится с SYSLOG в виде ответов на сообщения. В табл. 8.7 приведены все возможные сочетания ответов и функции, выполняемые программой в каждом случае.

8.5.4. ПРОТОКОЛ

Рис. 8.11 дает представление о том, какая информация выводится на печать в результате выполнения программы ESTVUT:

```
1  MAGNETIC TAPE ERROR STATISTICS BY VOLUME
2          3          4          5          6          7          8
VOLUME                TIME       CHANNEL    TEMP       TEMP       PERM
SERIAL     DATE       OF DAY     /UNIT      READ       WRITE      READ
xxxxxx     xx/xxx     xx.xx.xx   xxx        xxx        xxx        xxx
9          10         11         12         13         14         15
PERM       NOISE      ERASE      CLEANER    SIO'S      TAPE       BLOCK
WRITE      BLOCKS     GAPS       ACTIONS    (USAGE)    DENSITY    LENGTH
xxx        xxx        xxx        xxx        xxxxx      xx         xxx
16 {ESTV DISK FILE DUMPED BUT NOT CLEARED TO ZEROS или
    ESTV DISK FILE DUMPED AND CLEARED TO ZEROS или
    ESTV TAPE FILE DUMPED
```

Рис. 8.11. Формат протокола Статистики тома

Таблица 8.7

Управляющая информация для ESTVUT

Ответ на сообщение		Функция
4R02A INPUT=	4R04A OUTPUT=	
{5052 / 5061}	PRC	Вывод на печать информации файла Статистики тома и очистка этого файла (см. 8.5.4)
{5052 / 5061}	PRNC	Вывод на печать информации файла Статистики тома без очистки этого файла (см. 8.5.4)
{5052 / 5061}	TAPE	Создание исторического файла Статистики тома на магнитной ленте на основе основного (на SYSREC) файла Статистики тома. При этом выполняется очистка основного файла Статистики тома. При выполнении этой функции всегда используются две магнитные ленты: новая (или выводная) лента, т. е. лента для создания исторического файла Статистики тома и старая (или вводная) лента, которая может быть пустой или может содержать созданный ранее исторический файл Статистики тома. Причем если старая лента не пустая, то информация со старой ленты переписывается на новую, а следом за ней записывается информация из основного файла Статистики тома
TAPE	{PRC / PRNC}	Вывод на печать исторического файла Статистики тома (см. 8.5.4)
TAPE	TAPE	Копирование исторического файла Статистики тома на новую магнитную ленту

1 — заголовок распечатки статистики ошибок на томе магнитной ленты;
2 — регистрационный номер тома;
3 — дата;
4 — время дня;
5 — адрес физического устройства;
6 — количество исправимых ошибок при чтении;
7 — количество исправимых ошибок при записи;
8 — количество неисправимых ошибок при чтении;
9 — количество неисправимых ошибок при записи;
10 — количество некачественных записей;
11 — количество скачков;
12 — количество очищающих действий;
13 — количество команд SIO;
14 — плотность записи;
15 — длина блока;
16 — сообщение о конце распечатки.

8.6. ИНТЕРФЕЙС УЧЕТА ЗАДАНИЙ

Статистика задания позволяет учитывать время, затраченное на выполнение той или иной программы, причину завершения программы, емкость основной памяти, используемой программой, и некоторые другие характеристики. Полученная информация может быть использована для повышения эффективности работы системы и проведения необходимых финансовых расчетов с пользователями вычислительной системы. Собираемая статистика помещается в область Супервизора в специальную таблицу статистики задания (см. табл. 8.8).

Таблица 8.8

Статистика задания

Байты таблицы	Длина в байтах	Содержимое
0—7	8	Имя задания из оператора JOB
8—23	16	Информация пользователя (16 символов) из оператора JOB
24—25	2	Идентификатор раздела (BG, F2 или F1)
26	1	Код завершения (см. 2.4)
27	1	Тип статистической записи: S — для любого шага задания, L — для последнего шага задания
28—35	8	Дата: мм/дд/гг или дд/мм/гг в зависимости от того, какой формат указан при генерации системы (мм — месяц, дд — день, гг — две последние цифры года)
36—39	4	Время начала задания: ччммсс F (чч — часы, мм — минуты, сс — секунды, F — знак «+» в упакованном десятичном формате)
40—43	4	Время окончания задания: для записей типа S — нули, для записей типа L формат совпадает с форматом времени начала задания
44—47	4	Резервируются
48—55	8	Имя фазы из оператора EXEC
56—59	4	Наибольший шестнадцатеричный адрес основной памяти, используемый программой (из области связи)

Продолжение табл. 8.8

Байты таблицы	Длина в байтах	Содержимое
60—63	4	Время процессора: 4 двоичных байта, задающие количество единиц времени (одна единица равна 1/300 секунды). Время вычисляется от выхода из программы $ JOBACCT до следующего входа в нее. Время выполнения программы $JOBACCT включается в системное время следующей записи
64—67	4	Системное время: 4 двоичных байта, задающие количество единиц времени (одна единица равна 1/300 секунды). Время работы Супервизора. Это время заносится в таблицу статистики задания для каждого раздела
68—71	4	Время ожидания для данного раздела: 4 двоичных байта, задающие количество единиц времени (одна единица равна 1/300 секунды)
72—n	n—71	Таблицы команд запуска операций ввода-вывода: 6 байт для каждого периферийного устройства, количество которых задается в параметре JA макрокоманды FOPT. Первые два байта — X'0cuu', следующие 4 байта — шестнадцатеричный счетчик команд запуска операций ввода-вывода для шага задания. После генерации системы каждая таблица содержит X'10' и 5 нулей. При обращении к периферийному устройству заносится информация в соответствующую таблицу. Команды запуска операций ввода-вывода при избавлении от ошибок ввода-вывода не учитываются в таблице статистики задания
n+1	1	Байт переполнения (обычно X'20'). Может быть X'30', если количество периферийных устройств, используемых в разделе, больше количества устройств, заданного при генерации системы в параметре JA макрокоманды FOPT

Программа пользователя. Чтобы использовать информацию из таблицы статистики задания, пользователь должен написать программу, которая будет извлекать ее из таблицы, обрабатывать нужным образом, выводить на печать или запоминать на внешнем носителе. Программа должна быть самоперемещающейся и каталогизироваться в библиотеку абсолютных модулей под именем $JOBACCT. Размер этой программы может быть не более 4096 байт. Программа $JOBACCT вызывается в раздел Управлением заданиями в конце каждого шага задания и в конце задания.

При написании программы пользователь может использовать следующую информацию из регистров:

регистр 11 (длина таблицы статистики задания);

регистр 12 (регистр базы для $JOBACCT);

регистр 13 (адрес области сохранения программы пользователя, которая находится в области управляющей программы);

регистр 14 (регистр связи, который используется для обеспечения выхода из $JOBACCT);

регистр 15 (адрес таблицы статистики задания).

Содержимое регистра 14 при выходе из программы $JOBACCT должно совпадать с содержимым этого же регистра в момент входа в в нее. Регистры 0, 1, 13, 14, 15 используются программами логической системы управления вводом-выводом. Если программа $JOBACCT использует логическую систему управления вводом-выводом, необходимо сохранить и восстановить содержимое этих регистров.

Программа $JOBACCT может использовать следующие два метода программирования операций ввода-вывода:

логическую систему управления вводом-выводом без обработки меток или физическую систему управления вводом-выводом;

логическую систему управления вводом-выводом с обработкой меток.

При использовании первого метода в макрокоманде генерации Супервизора FOPT, описывающей функцию обеспечения интерфейса учета заданий, должен быть указан только параметр JA, при использовании второго метода как JA, так и JALIOCS.

ГЛАВА 9

ГЕНЕРАЦИЯ СИСТЕМЫ

9.1. ПЛАНИРОВАНИЕ СИСТЕМЫ

Процесс получения конкретной операционной системы, в максимальной степени отвечающей требованиям пользователя и соответствующей конфигурации вычислительного оборудования конкретной установки ЕС ЭВМ, называется генерацией системы.

В процессе генерации пользователем должны быть решены такие основные проблемы: планирование Супервизора — выбор свойств конкретного ядра Супервизора; планирование библиотек — определение состава, размера и содержимого библиотек конкретного резидентного пакета; планирование системных файлов — определение размеров и выделение места во внешней памяти для этих файлов.

9.2. МАКРОКОМАНДЫ ГЕНЕРАЦИИ СУПЕРВИЗОРА

Построение ядра Супервизора с заданными пользователем функциями и возможностями осуществляется путем трансляции группы специальных макрокоманд, которым соответствуют макроопределения генерации Супервизора. Операндами макрокоманд описываются требуемые функции и возможности. При выборе операндов макрокоманд необходимо следить за их функциональной совместимостью.

Общие правила записи. Структура:

⟨код⟩ ⟨операнды⟩ ⟨комментарии⟩

код — мнемонический код макрокоманды;

операнды — записываются в ключевом формате. Последовательность записи операндов в каждой из макрокоманд не фиксирована;

комментарии — используются для записи описательной информации.

Составные части макрокоманды отделяются одна от другой по крайней мере одним пробелом. Для записи макрокоманды могут использоваться перфокарты продолжения. Макрокоманды генерации должны располагаться в следующей последовательности: SUPVR, CONFG, STDJC, FOPT, PIOCS, ALLOC, IOTAB, DVCGEN, ASSGN, SEND.

Макрокоманды ALLOC, DVCGEN, ASSGN являются необязательными и могут отсутствовать в исходном модуле. Наличие всех остальных макрокоманд является обязательным.

9.2.1. SUPVR

Предназначена для определения режимов работы системы. Операнды:

$$MPS = \left\{\begin{matrix}\underline{NO}\\YES\\BJF\end{matrix}\right\}$$

Режим мультипрограммирования: NO — Супервизор не обеспечивает мультипрограммной работы; YES—Супервизор обеспечивает мультипрограммную работу с возможностью выполнения в разделах переднего плана только одиночных программ; BJF—Супервизор обеспечивает мультипрограммную работу с возможностью использования разделов переднего плана для выполнения программ как в пакетном, так и в одиночном режимах (для каждого из разделов создается своя область связи).

$$TP = \left\{\begin{matrix}\underline{NO}\\ROTAM\\BTAM\end{matrix}\right\}$$

Системы телеобработки: NO — Супервизор не обеспечивает телеобработку; ROTAM—Супервизор обеспечивает возможность использования системы телеобработки ROTAM (обеспечение программируемого мультиплексора EC-8404); BTAM—Супервизор обеспечивает возможность использования базисного метода телекоммуникационного доступа (BTAM).

$$AP = \left\{\begin{matrix}\underline{NO}\\YES\end{matrix}\right\}$$

Мультизадачный режим: NO — Супервизор не обеспечивает мультизадачный режим, YES—Супервизор обеспечивает мультизадачный режим, с возможностью выполнения более одной задачи в пределах раздела; AP = YES предполагает MPS = YES и WAITM = YES.

ERRLOG = операнд 1 Режим сбора информации об ошибках ввода-вывода (записи OBR и SDR) и о надежности оборудования (записи RDE). Операнд 1 может принимать значения NO, n, YES, RDE, (n, RDE), (YES, RDE). n — число записей SDR в регистрационном файле IJSYSRC.

Когда задано ERRLOG=YES, n полагается равным числу строк PUB. ERRLOG = RDE определяет функцию сбора данных о надежности оборудования и действителен только для моделей «Ряда-2». Если RDE задан без n или YES, то предполагается YES. Если RDE задан для модели «Ряда-1», то генерация Супервизора прекращается. Когда при генерации Супервизора задается модель «Ряда-2», то автоматически полагается ERRLOG=YES, если он не задан или задан ERRLOG=NO.

$$MCRR = \left\{\begin{matrix}\underline{NO}\\YES\end{matrix}\right\}$$

Режим обработки машинных ошибок и ошибок канала для моделей «Ряда-1». При MCRR = = YES генерируются подпрограммы MCRR, которые формируют записи OBR в файле

при прерываниях по машинным ошибкам, ошибкам управления канала и интерфейса. Задание в разделе, затронутом ошибкой, снимается без попытки избавления от ошибки. MCRR = YES действительно только для моделей EC-1020, EC-1022 и EC-1030 «Ряда-1».

9.2.2. CONFG

Описывает оборудование ЭВМ и может задавать генерацию дополнительных функций Супервизора. Операнды:

MODEL = {2020 / nnnn} Шифр процессора модели ЕС ЭВМ. Должен быть равен 2020 для EC-1020, 2022 для EC-1022, 2030 для EC-1030, 2040 для EC-1040, 2050 для EC-1050, 2060 для EC-1060 и 2035 для EC-1035. Модель EC-1035 относится к моделям «Ряда-2». Когда MODEL = 2035, дополнительно генерируются функции обработки машинных ошибок и ошибок канала (MCAR/CCH). Если ERRLOG и OLTEP опущены или заданы ERRLOG = NO и OLTEP = NO, то также предполагается ERRLOG = YES и OLTEP = YES. Если задан MCRR = YES, то операнд MODEL может задавать только процессоры моделей «Ряда-1».

SP = {NO / YES} NO — отсутствие защиты памяти; YES — ее наличие. YES полагается, если задан MPS = YES или MPS = BJF в макрокоманде SUPVR.

DEC = {NO / YES} NO — отсутствие десятичной арифметики; YES — ее наличие.

FP = {NO / YES} NO — отсутствие арифметики плавающей запятой; YES — ее наличие.

TIMER = {NO / YES} NO — отсутствие интервального таймера; YES — его наличие.

9.2.3. STDJC

Макрокоманда описывает стандартные режимы работы системы. Операнды:

DECK = {YES / NO} Стандартный режим вывода объектных модулей на SYSPCH.

LIST = {YES / NO} Стандартный режим вывода результатов трансляции на SYSLST.

XREF = {YES / NO} Стандартный режим вывода на SYSLST таблицы перекрестных ссылок.

ERRS = {YES / NO} — Стандартный режим вывода на SYSLST списка всех ошибок в исходной программе Базисного Фортрана, Фортрана IV, ПЛ/1 или Кобола.

LOG = {YES / NO} — Стандартный режим вывода на SYSLST директив и операторов Управления заданиями.

DUMP = {YES / NO} — Стандартный режим вывода на SYSLST содержимого регистров и основной памяти в случае ненормального завершения проблемной программы.

LINES = {56 / nn} — Стандартное количество строк на странице SYSLST (минимум 30, максимум 99).

DATE = {DMY / MDY} — Стандартный формат даты: DMY — день, месяц, год; MDY — месяц, день, год.

CHARSET = {48C / 60C} — Набор знаков алфавита для транслятора ПЛ/1 при вводе данных с SYSIPT (48 или 60 знаков в наборе).

LISTX = {NO / YES} — Стандартный режим печати на SYSLST (в шестнадцатеричном виде) объектных модулей при выполнении трансляторов ПЛ/1, Фортран IV, Кобол.

SYM = {NO / YES} — Стандартный режим вывода таблицы символических имен трансляторами Ассемблер (на SYSPCH), ПЛ/1 и Кобол (на SYSLST).

SPARM = {NO / YES} — Указывается, если требуется поддержка системного параметра &SYSPARM в Ассемблере E (вариант 14K).

9.2.4. FOPT

Макрокоманда определяет дополнительные функциональные возможности Супервизора. Операнды:

OC = {NO / YES} — Возможность использования в проблемных программах специальных подпрограмм для связи с оператором, включаемых в работу с помощью Супервизора. При мультипрограммной работе каждая из одновременно выполняемых программ может иметь свою подпрограмму для связи с оператором.

PC = {NO / YES} — Возможность использования в проблемных программах подпрограмм для обработки программных прерываний. При мультипро-

граммной работе каждая из одновременно выполняющихся программ или задач может иметь свою подпрограмму для обработки программных прерываний. Нужно задавать YES, если используются программы Базисный Фортран, Фортран IV, Кобол, ПЛ/1 и РПГ.

$$IT = \left\{\begin{matrix}\underline{NO}\\BG\\F1\\F2\end{matrix}\right\}$$

Определяет возможные способы использования интервального таймера проблемными программами и указывает раздел, которому назначается таймер. При NO и операнде TIMER = YES в макрокоманде CONFG проблемные программы могут использовать интервальный таймер только для получения текущего времени дня с помощью макрокоманды GETIME. Когда задается BG или F1 или F2, операнд TIMER в макрокоманде CONFG полагается равным YES.

$$TEB = \left\{\begin{matrix}\underline{NO}\\n\end{matrix}\right\}$$

Определяет режим ведения статистики об ошибках, возникающих при работе с магнитными лентами. При NO статистика не ведется, при n статистика об ошибках на магнитных лентах, возникших при выполнении задания, ведется и после завершения задания в виде таблицы выводится на SYSLOG; n должно указывать количество накопителей на магнитной ленте, подключенных к системе, и не превосходить 255.

$$TEBV = \left\{\begin{matrix}\underline{NO}\\\left(\left\{\begin{matrix}\underline{DASD}\\SYSLOG\end{matrix}\right\}, n\right)\end{matrix}\right\}$$

Определяет режим ведения статистики об ошибках на томе магнитной ленты. При закрытии файла, достижении конца тома и завершении задания статистика об ошибках, несущая качественную оценку магнитной ленты, выводится на SYSLOG или SYSREC в зависимости от значения параметра TEBV (SYSLOG или DASD). n — количество накопителей на магнитной ленте, подключенных к системе, и не превышает 255.

$$EVA = \left\{\begin{matrix}\underline{NO}\\(r, w, n)\end{matrix}\right\}$$

При превышении указанного числа (r) ошибок во время чтения тома магнитной ленты или превышении указанного числа (w) ошибок, возникших при записи на том магнитной ленты, оператор информируется об этом посредством сообщения, выданного на SYSLOG. Числа r и w не должны превышать 254. n — количество накопителей на магнитной ленте, подключенных к системе.

Примечание. Значения n в операндах TEB, TEBV и EVA должны быть идентичными, если указаны два или три из этих режимов. Если заданы режимы TEB, TEBV или EVA, а операнд TAPE в макрокоманде PIOCS не задан или принимает значение TAPE = NO, то предполагается TAPE = 9.

SKSEP = {NO, YES, n}

Определяет способ выполнения канальных программ для дисков. При YES или n Супервизор отделяет команду подвода от канальной программы и выполняет ее как отдельное обращение к накопителю. После выполнения требуемого подвода организуется выполнение остальной части канальной программы. n указывает количество накопителей на дисках, которые могут быть подключены к машине, и не может быть меньше количества, описанного макрокомандами DVCGEN.

CE = {NO, YES, n}

Определяет возможность использования программы Протоколирование событий. При YES для области резервируется 600 байт. Задавая n от 600 до 10240, пользователь может указать объем памяти, резервируемой для области CE.

JA = {NO, YES, (n1, n2, n3)}

Определяет возможность ведения статистики заданий. n1, n2, n3 — количество устройств, для которых ведется подсчет команд запуска операций ввода-вывода соответственно в разделах BG, F2, F1. Если JA ≠ NO, то полагается TIMER = YES. Возможность ведения статистики задания для разделов переднего плана генерируется только при MPS = BJF.

JALIOCS = {NO, (s, *l*)}

Задается для программы ведения статистики заданий, написанной пользователем с использованием ЛСУВВ (без обработки меток или с обработкой меток). s — длина области сохранения для программы пользователя, резервируемой в Супервизоре. Максимальное значение — 1024. Если s опущено или JALIOCS = NO, то в Супервизоре резервируется 16 байт; *l* — длина сменной области меток (максимальное значение — 224).

PTO = {NO, YES}

Определяет режим выполнения операций ввода-вывода, запрошенных самим Супервизором. При NO на время ожидания завершения этих операций машина переводится в состояние ожидания. При YES параметр MPS макрокоманды SUPVR должен быть равным YES или BJF.

IDRA = {NO, YES}

Операнд IDRA позволяет расширить возможности PTO: когда задано IDRA = YES, становится возможным выполнять загрузку фаз в то время, когда занята A-транзитная область, тем самым повышается производительность машины. YES действительно в том случае, если MPS = YES или MPS = BJF в макрокоманде SUPVR. Если задано YES, а PTO не задано или равно NO, то принимается PTO = YES.

OLTEP = {NO / YES} Определяет возможность использования программы проверки устройств. Если OLTEP не задан или OLTEP = NO, а в макрокоманде CONFG задана модель «Ряда-2», то принимается OLTEP = YES. Если OLTEP = YES, то полагается OC = YES.

PCIL = {NO / YES} Определяет возможность использования личных библиотек абсолютных модулей.

CBF = {NO / n} Определяет режим выполнения операций вывода на пишущей машинке. При CBF = n информация, подлежащая выводу на пишущую машинку, пересылается из области проблемной программы в специальную область памяти внутри Супервизора (буфер), откуда и будет осуществляться ее вывод. Отметка о завершении операций вывода выполняется после пересылки (до ее действительного запуска и завершения). n определяет количество буферов (1—50), которые создаются в области Супервизора.

CCHAIN = {NO / YES} Определяет режим повторения операций ввода-вывода при возникновении ошибок. При YES пользователь в блоке CCB может запросить повторение операции ввода-вывода, начиная с команды ввода-вывода, при выполнении которой произошла ошибка (выполнение всей канальной программы не повторяется) При NO независимо от требования пользователя при повторении операции ввода-вывода выполняется вся канальная программа.

TRKHLD = {NO / n} Определяет режим защиты дорожек на устройствах прямого доступа от одновременного обращения к ним нескольких задач этого же или другого раздела. n задает количество дорожек, которые можно защитить одновременно. Максимальное значение n не превосходит 255. Если операнд n недействителен (не число) или вне предела 1—255, то n полагается равным 10. При TRKHLD = = n должен быть задан операнд MPS = YES или MPS = BJF.

AB = {NO / YES} Определяет возможность использования подпрограммы обработки ненормальных завершений заданий.

WAITM = {NO / YES} Определяет режим мультиожидания. WAITM полагается равным YES, если задается режим AP = YES.

DASDFP = {NO / (n, n, [,5052])} Определяет необходимость организации Супервизором защиты файлов на дисках. При DASDFP = = NO защита не организуется, n, n

определяют номера каналов, к которым подключены накопители на дисках. Они должны записываться в неубывающем порядке. 5052 определяет, что защита файлов должна быть организована на накопителях ЕС-5052, ЕС-5055, ЕС-5056, ЕС-5061.

SYSFIL = {NO / (5052 [, n1, n2])} Определяет возможность назначения дисков системным логическим устройствам SYSIPT, SYSRDR, SYSPCH, SYSLST. При NO такое назначение невозможно. 5052 определяет возможность назначения этим логическим устройствам накопителей ЕС-5052, ЕС-5055, ЕС-5056, ЕС-5061. n1 устанавливает минимальную величину свободной области на дисках, отведенной SYSLST и необходимой для выполнения каждого задания, и записывается в виде десятичного числа (от 100 до 65635), выражающего величину этой области в количестве записей, вмещающихся в нее. n2 имеет аналогичный смысл для SYSPCH. Если n1 и n2 опущены, они предполагаются равными 1000. При MPS = BJF системным логическим устройствам могут назначаться диски во всех разделах.

9.2.5. PIOCS

Макрокоманда определяет состав оборудования ввода-вывода. Операнды:

SELCH = {YES / NO} Определяет наличие селекторных каналов.

BMPX = {NO / YES} Указывает возможность подключения к мультиплексному каналу устройств, работающих в монопольном режиме.

CHANSW = {NO / TSWTCH} Определяет возможность подключения устройства управления магнитной лентой к двум смежным каналам.

TAPE = {NO / 9} NO — указывает, что в состав оборудования ввода-вывода не входит магнитная лента. 9 — указывает, что в состав оборудования включена девятидорожковая магнитная лента. Если заданы TEB, TEBV или EVA, то принимается TAPE = 9.

9.2.6. ALLOC

Макрокоманда определяет распределение памяти между разделами переднего плана. Операнды: F1 = nK, F2 = nK, n должно быть четным десятичным числом, K = 1024 байта. Использование макрокоманды ALLOC не обязательно.

9.2.7. IOTAB

Макрокоманда описывает требования к таблицам ввода-вывода внутри ядра Супервизора. Операнды:

IODEV = $\left\{\begin{matrix}10\\n\end{matrix}\right\}$ Определяет количество устройств ввода-вывода, подключенных к машине. Значение n не может быть меньше количества внешних устройств, описанных макрокомандами DVCGEN, и должно быть в пределах от 4 до 255.

BGPGR = $\left\{\begin{matrix}10\\n\end{matrix}\right\}$ Определяет количество логических устройств программиста в фоновом разделе и может принимать значения от 10 до 222.

F1PGR = $\left\{\begin{matrix}5\\n\end{matrix}\right\}$ Определяет количество логических устройств программиста в разделах переднего плана: соответственно в разделе F1 и в разделе F2. Эти операнды могут использоваться, если в макрокоманде SUPVR операнд MPS = YES или MPS = BJF, и могут принимать значения от 5 до 222. Сумма значений параметров BGPGR, F1PGR и F2PGR не должна превосходить 222.

F2PGR = $\left\{\begin{matrix}5\\n\end{matrix}\right\}$

JIB = $\left\{\begin{matrix}5\\n\end{matrix}\right\}$ Определяет размер таблицы информации задания. При подсчете количества строк в этой таблице нужно исходить из того, что одна строка требуется для каждого временного назначения логическому устройству, для каждого сменного назначения логическому устройству и для каждого участка открытого файла на дисках при наличии защиты файлов на дисках. Операнд может принимать значение от 5 до 255.

CHANQ = $\left\{\begin{matrix}6 \text{ или } 6+\text{CBF}\\n\end{matrix}\right\}$ Определяет размер таблицы очередей к устройствам (n — число строк в таблице). Минимальное значение n равно 6 плюс значение параметра CBF в макрокоманде FOPT, максимальное — 255.

9.2.8. DVCGEN

Используется для описания внешних устройств, подключенных к машине. Одна макрокоманда используется для описания одного внешнего устройства. Операнды:

CHUN = X'cuu'. Определяет адрес внешнего устройства. c — номер канала (0—6); uu — номер устройства в канале (00—FF).

DVCTYP = xxxxxx Определяет шифр устройства (xxxxxx — шифр устройства).

CHANSW = $\left\{\begin{matrix}\text{NO}\\\text{YES}\end{matrix}\right\}$ YES — указывает, что устройство подключено к двум каналам (операнд CHUN должен

указывать в этом случае канал с меньшим номером).

MODE = X′ss′ Для магнитных лент этот операнд определяет режим плотности записи и принимает значения X′C8′, X′C0′ или X′F8′. Значение X′C8′ устанавливает плотность 32 байта/мм, X′C0′ — 63 байта/мм, X′F8′ — 8 байт/мм. Если этот операнд опущен, устанавливается плотность 63 байта/мм.

9.2.9. ASSGN

Определяет стандартные назначения. Операнды:

SYSxxx, X′cuu′, $\left[\left[,\left\{\begin{matrix}BG\\F1\\F2\end{matrix}\right\}\right]\right]$ SYSxxx — логическое устройство. cuu — адрес конкретного физического устройства ввода-вывода, назначенного указанному логическому устройству. BG, F1 или F2 указывают раздел, в котором должно быть сделано назначение.

9.2.10. SEND

Определяет размер области Супервизора. Операнды: [n]. n должно быть кратным 8 и достаточным для размещения Супервизора с требуемыми функциями, но не превышать 32760 (32K — 8) байт. Если операнд n опущен, размер области Супервизора определяется автоматически во время трансляции Супервизора.

9.3. СТАРТЕР

9.3.1. НАЗНАЧЕНИЕ И СОСТАВ

Программа Стартер является системно-независимой и способна выполнить работу по копированию файлов с пакетов дисков на магнитную ленту, инициализации новых пакетов дисков и восстановлению на них файлов с магнитной ленты.

В состав Стартера входят следующие самостоятельные программы: программа копирования (DISCPY); программа инициализации (INTDSK); программа восстановления (DISRST).

Программа копирования формирует магнитную ленту, которая содержит программу инициализации дисков, программу восстановления и произвольное число файлов, скопированных с дисковых пакетов в заданной последовательности.

Программа инициализации дисков за одно выполнение инициализирует от одного до пяти пакетов дисков, установленных на различных накопителях.

Программа восстановления выполняет восстановление на инициализированных пакетах дисков данных с магнитной ленты, полученной во время копирования программой Стартер.

9.3.2. УПРАВЛЯЮЩИЕ ОПЕРАТОРЫ

Признаком управляющего оператора являются символы //. Код оператора отделяется от признака не менее чем одним пробелом. Пробелом также отделяются последующие операнды от кода оператора.

Стартер обеспечивает прием и обработку следующих операторов управления заданиями: JOB, DATE, ASSGN, EXEC, CONFG, LOG, NOLOG, FILES.

JOB. Указывает на начало управляющей информации для одного задания. Формат:

```
// JOB имя
```

где имя — имя задания, одновременно представляющее собой имя программы, выполняемой в этом задании. Принимает одно из трех значений: DISCPY, INTDSK, DISRST.

DATE. Задает текущую дату дня. Формат:

```
// DATE ггддд
```

где ггддд — дата текущего дня; гг — две последние цифры года (00—99), ддд — порядковый номер текущего дня в году (001—366).

ASSGN. Используется для установления соответствия между символическим именем логического устройства и физическим адресом этого устройства. Формат:

```
// ASSGN SYSxxx, X'cuu', tt [,X'ss']
```

где SYSxxx — символическое имя логического устройства SYSLOG SYSIPT, SYSRDR, SYSOPT, SYS000—SYS005);

X'cuu' — физический адрес устройства (c — номер канала, uu — номер устройства);

tt — тип устройства. Принимает значения: C1 — пишущая машинка, D1 — диски типа ЕС-5052, D3 — диски типа ЕС-5061, T2 — магнитная лента;

X'ss' — устанавливает режим плотности записи на магнитную ленту и принимает значение X'C0'. Если параметр опущен или равен X'C0', устанавливается плотность 32 байт/мм.

EXEC. Указывает, что необходимо начать выполнение программы, имя которой было задано в операторе JOB. Формат:

```
// EXEC
```

CONFG. Служит для сообщения информации об объеме оперативной памяти. Формат:

```
// CONFG kkkk000n
```

где kkkk — указание на объем оперативной памяти данной установки: 0110—64K, 1000—128K, 1010—256K и выше.

n — информирует о возможности использования пишущей машинки: 1 — пишущая машинка используется для связи с оператором, 0 — пишущая машинка не используется. Если CONFG опущен, предполагается использование пишущей машинки для связи с оператором и объем оперативной памяти — 32K.

LOG. Устанавливает режим, при котором все операторы управления заданиями выпечатываются на SYSLOG. Формат:

// LOG

NOLOG. Отменяет режим печати управляющих операторов на SYSLOG. Формат:

// NOLOG

FILES. Может использоваться для установки магнитной ленты к файлу, который содержит программу восстановления, если пропускается этап инициализации дисковых пакетов. Формат:

// FILES SYSxxx, n

где SYSxxx — логическое устройство;
n — число ленточных марок (файлов) для пропуска.

9.3.3. СПЕЦИАЛЬНЫЕ ОПЕРАТОРЫ ДЛЯ ПРОГРАММЫ КОПИРОВАНИЯ

Оператор комментариев является необязательным, но если присутствует, то должен предшествовать оператору модификации. Оператор заносится на магнитную ленту и при последующем восстановлении печатается на SYSLOG. Формат:

* комментарии

Оператор модификации используется для указания файлов, подлежащих копированию. Формат:

// UDS—'идентификатор файла'

где идентификатор файла — идентификатор файла (из метки формата 1) Длина — 44 символа. Если фактически длина идентификатора меньше 44, он дополняется пробелами справа.

9.3.4. СПЕЦИАЛЬНЫЕ ОПЕРАТОРЫ ДЛЯ ПРОГРАММЫ ИНИЦИАЛИЗАЦИИ ДИСКОВ

Оператор модификации задает режимы инициализации дисков. Формат:

// UID Ix.Cn, R=(u,ccccchhh),...

где Ix — задает режим анализа поверхности. x = A — анализируется поверхность всех дорожек пакета; x = N — поверхность пакета не анализируется, x = R — не анализируется поверхность тех дорожек, которые уже обозначены как дефектные. Если оператор опущен, предполагается X = R;

Cn — указывает, сколько раз должен выполняться анализ поверхности каждой дорожки ($1 \leqslant n \leqslant 255$);

R = (u, ccccchhh) — указывает адрес дорожки, которая должна быть отмечена как дефектная без анализа ее поверхности. cccc — номер цилиндра, hhh — номер дорожки, u — фиксирует логическое устройство, на котором установлен дисковый пакет с указанной дефектной дорожкой: u = 1 — SYSOPT, u = 2 — SYS002, u = 3 — SYS003,

u = 4 — SYS004, u = 5 — SYS005. Может быть задано не более 30 операндов. Этот оператор может иметь карты продолжения.

Оператор оглавления тома содержит информацию, необходимую для построения оглавления тома на инициализируемом пакете. Форматы:

```
// VTOC STANDARD
// VTOC STRTADR = (ccccchhh),EXTENT = (n)
```

где STANDARD — стандартное расположение оглавления тома: следом за последней меткой тома на цилиндре 0 до конца цилиндра;

STRTADR = (ccccchhh) — адрес начала расположения оглавления тома на пакете (cccc — номер цилиндра, hhh — номер дорожки);

EXTENT = (n) — число дорожек для оглавления тома.

Оператор меток тома предназначен для описания одной метки тома VOLi. Формат:

```
VOLinnnnnn
```

где VOL — идентификатор метки тома;

i — номер метки тома ($1 \leqslant i \leqslant 8$);

nnnnnn — регистрационный номер тома (6 байт), если i = 1, или любая информация (76 байт), если $2 \leqslant i \leqslant 8$.

Оператор конца отмечает конец ввода управляющих операторов программы инициализации. Формат:

```
// END
```

9.3.5. ПРОГРАММА КОПИРОВАНИЯ

Задание на выполнение

Оператор	Пояснения
// JOB DISCPY	Начало задания
// DATE	Текущая дата
// ASSGN SYSLOG	Необходим, если абсолютный адрес физического устройства для SYSLOG отличен от X'01F'
// ASSGN SYS000	Назначение устройства, на котором установлен первый (единственный) копируемый пакет дисков
// ASSGN SYS001	Назначение устройства, на котором установлен второй копируемый пакет дисков
// ASSGN SYSOPT	Назначение основного устройства, на котором установлена магнитная лента для копирования
// ASSGN SYS002	Назначение устройства, на котором установлена сменная магнитная лента
// CONFG	Используется, если объем основной памяти отличен от 32K или для связи с оператором не используется пишущая машинка
// LOG (NOLOG)	Используется по желанию и занимает любое место до оператора EXEC
// EXEC	Выполнение программы
оператор комментариев	Используется по желанию (только один)
операторы модификации	Указание идентификаторов копируемых файлов

Инструкция оператору. Оператор должен выполнить следующие действия:

пакет перфокарт, содержащий программу копирования, положить в подающий магазин устройства ввода с перфокарт;

нажать кнопки ПУСК И КОНЕЦ КАРТОТЕКИ на пульте управления устройства ввода с перфокарт;

подготовить к работе накопитель (накопители) на магнитных дисках и ленте

на переключателях загрузки набрать абсолютный адрес устройства ввода с перфокарт

нажать кнопку ЗАГРУЗКА.

Пример задания

```
// LOG
// JOB DISCPY
// DATE 76361
// ASSGN SYS000, X'190', D1
// ASSGN SYS001, X'191', D1 (для сменного пакета дисков)
// ASSGN SYSOPT, X'280', T2
// ASSGN SYS002, X'281', T2 (для сменной магнитной ленты)
// EXEC
```

9.3.6. ПРОГРАММА ИНИЦИАЛИЗАЦИИ

Задание на выполнение

Оператор	Пояснения
// JOB INTDSK	Начало задания
// DATE	Текущая дата
// ASSGN SYSLOG	Необходим, если абсолютный адрес физического устройства для SYSLOG отличен от X'01F'
// ASSGN SYSOPT	Назначение устройства, на котором установлен первый инициализируемый пакет дисков
// ASSGN SYS00i	Назначение устройств, на которых установлены второй и последующие пакеты дисков для инициализации (i = 1, 2, ..., 5)
// CONFG	Используется, если объем основной памяти отличен от 32К или для связи с оператором не используется пишущая машинка
// LOG (NOLOG)	Используется по желанию и занимает любое место до оператора EXEC
// EXEC	Выполнение программы
оператор модификации	Установка режима инициализации
// VTOC	Описание оглавления тома для выполнения форматизации
VOLi	Описание метки тома инициализируемого пакета дисков (i =1, 2, ..., 8). Для i =1 оператор VOL должен присутствовать обязательно
// END	Конец управляющих операторов

Задание для пропуска инициализации пакетов дисков

Оператор	Пояснения
// JOB INTDSK	Начало задания
// ASSGN SYSIPT	Назначение устройства, на котором установлена магнитная лента с программой инициализации
// FILES SYSIPT, 1	Используется для пропуска файла с программой инициализации на магнитной ленте

Инструкция оператору. Оператор должен выполнить следующие действия:

установить в точку загрузки предназначенную для восстановления магнитную ленту, первым файлом которой является программа инициализации;

в подающий магазин устройства ввода с перфокарт положить пакет с заданием на выполнение программы;

подготовить к работе накопители на магнитных дисках;

на переключателях загрузки набрать абсолютный адрес накопителя на магнитной ленте, на котором установлена магнитная лента с программой инициализации;

нажать кнопку ЗАГРУЗКА;

после перехода машины в состояние ожидания нажать кнопки КОНЕЦ КАРТОТЕКИ и ПУСК на пульте управления устройства ввода с перфокарт.

Пример задания

```
// JOB INTDSK
// DATE 76284
// ASSGN SYSLOG,X'01F',C1
// ASSGN SYSOPT,X'190',D1
// ASSGN SYS002,X'191',D1
// EXEC
// UID 1A,C1,R = (0007001)
// VTOC STRTADR = (0199000),EXTENT = (10)
  VOL1PRVLIB
// VTOC STRTADR = (0199000),EXTENT = (10)
  V0L1111111
// END
```

Пример задания для пропуска инициализации пакетов дисков

```
// JOB INTDSK
// ASSGN SYSIPT,X'280',T2
// FILES SYSIPT,1
```

9.3.7. ПРОГРАММА ВОССТАНОВЛЕНИЯ

Задание на выполнение

Оператор	Пояснения
// JOB DISRST	Начало задания
// DATE	Текущая дата
// ASSGN SYSLOG	Необходим, если абсолютный адрес физического устройства для SYSLOG отличен от X'01F'
// ASSGN SYS000	Необходим для назначения устройства, на котором установлен первый (или единственный) пакет дисков для восстановления
// ASSGN SYS001	Назначение устройства, на котором установлен второй пакет дисков для восстановления
// ASSGN SYS002	Назначение устройства, на котором установлена сменная магнитная лента для восстановления
// CONFG	Используется, если объем основной памяти отличен от 32K или для связи с оператором не используется пишущая машинка
// LOG (NOLOG)	Используется по желанию и занимает любое место до оператора EXEC
// EXEC	Выполнение программы

Инструкция оператору. Оператор должен выполнить следующие действия:

установить предназначенную для восстановления магнитную ленту перед вторым файлом;

в подающий магазин устройства ввода с перфокарт положить операторы управления заданиями;

подготовить к работе накопитель(и) на магнитных дисках;

на переключателях загрузки набрать абсолютный адрес накопителя на магнитной ленте, на котором установлена магнитная лента с программой восстановления;

нажать кнопку ЗАГРУЗКА;

после перехода машины в состояние ожидания нажать кнопки КОНЕЦ КАРТОТЕКИ и ПУСК на пульте управления устройства ввода с перфокарт.

Пример задания

```
// LOG
// JOB DISRST
// DATE 76317
// ASSGN SYS002,X'280',T2
// ASSGN SYS000,X'190',D1
// ASSGN SYS001,X'191',D1
// EXEC
```

ЛИТЕРАТУРА

1. Единая система ЭВМ. Под общ. ред. А. М. Ларионова. М., «Статистика», 1974.

2. Система математического обеспечения ЕС ЭВМ. Под общ. ред. А. М. Ларионова. М., «Статистика», 1974.

3. Шура-Бура М. Р., Ковалевич Э. В., Марголин М. С. [и др.]. Операционная система ДОС ЕС. Общие положения. М., «Статистика», 1975.

4. Брич З. С., Воюш В. И., Дегтярева Г. С., Ковалевич Э. В. Программирование на языке Ассемблера ЕС ЭВМ. М., «Статистика», 1975.

5. Засмужец Г. Т., Марук З. А., Церлюкевич В. И. Исследование зависимости временных характеристик от параметров сортируемого массива и оборудования. — В сб.: Вопросы радиоэлектроники. Серия ЭВТ, вып. 11, 1975.

УСЛОВНЫЕ ОБОЗНАЧЕНИЯ И ПРИНЯТЫЕ СОГЛАШЕНИЯ

ДКОИ — двоичный код для обмена и обработки информации;
PSW — слово состояния программы;
CSW — слово состояния канала;
CCW — команда ввода-вывода;
cuu — c — номер канала (0—6), uu — номер устройства в канале (00 — FF);
BG — фоновый раздел;
F1 — первый раздел переднего плана;
F2 — второй раздел переднего плана;
СУВВ — система управления вводом-выводом.

В заданиях на выполнение программ, в форматах управляющих операторов, директив и макрокоманд приняты следующие обозначения:

1. Информация, указанная в квадратных скобках, является необязательной и в конкретном применении может отсутствовать.

2. Информация, указанная в фигурных скобках, представляет собой альтернативные значения, одно из которых необходимо выбрать в конкретном применении. Значение, которое выбирается по умолчанию в случае, когда параметр (операнд) отсутствует, обычно подчеркнуто.

3. Многоточие внутри информации указывает, что допускается несколько элементов, непосредственно предшествующих многоточию и следующих за ним.

4. Другие знаки пунктуации (круглые скобки, запятые, знаки равенства, пробелы и т. д.) должны записываться точно так, как указано; пробел указывается пустой позицией.

5. Информация, представленная прописными латинскими буквами (коды операторов и макрокоманд, ключевые слова, идентификаторы параметров и т.д.), должны записываться точно так, как указано.

6. Информация, представленная строчными буквами, должна заменяться необходимыми значениями.

7. Запись (r), (r0), (r1), (r2) и т. п. означает регистровую форму записи операнда: значением операнда является содержимое указанного регистра. В частности, запись (0), (1) в операндной части операторов и макрокоманд указывает, что значением операнда является содержимое соответственно регистра 0 или регистра 1.

8. Информация, которая передается в регистрах системе программой пользователя или программе пользователя системой ДОС ЕС, всегда содержится в правых (младших) байтах регистра, старшие байты регистра равны нулю.

9. В заданиях на выполнение различных программ ДОС ЕС используются следующие соглашения:

для операторов и директив Управления заданиями обычно указываются только коды, для уточнения операндной части следует обращаться к гл. 2 (п. 2.2.6);

обычно указывается один оператор ASSGN для назначения логическим устройствам периферийных устройств; число операторов ASSGN в конкретном задании зависит от количества логических устройств, при этом оператор ASSGN можно не использовать для тех устройств, стандартные назначения которых

удовлетворяют программиста (стандартные назначения устанавливаются во время генерации системы);

количество операторов EXTENT в конкретном задании зависит от числа участков, занимаемых файлом.

10. В табл. 2.2 приводится список системных логических устройств с описанием стандартного использования этих устройств компонентами системы. Для обрабатывающих программ ДОС ЕС под заголовком «логические устройства» дается список всех логических устройств, необходимых при работе программы. При этом описание использования приводится для всех устройств программиста и тех системных устройств, применение которых данной программой отличается от стандартного.

11. Ввиду того, что при перфорации информации компонентами ДОС ЕС в качестве носителя данных могут, кроме перфокарт, использоваться магнитные ленты и диски, на которых записываются образцы перфокарт (карты), наряду с термином «перфокарты» употребляется термин «карты».

ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ

Алфавит Ассемблера 5

Библиотекарь 126
— операторы обновления 130
— программы 128
— режимы выполнения 128
— управляющие операторы 129
— — — коды 130
— — — операнды 132
— функции 126
Библиотеки 123
— абсолютных модулей 123
— исходных модулей 124
— личные 125
— объектных модулей 124
— системные 125
— размер 125
Блок CCB 62

Восстановление информации 159
Вывод оглавления тома 157
Выражение 7
— абсолютное 7
— арифметическое 24
— знаковое 25
— логическое 25
— переместимое 7

Генератор программ разгрузки 209
— — — оператор ASSGN 211
— — — — OPTN 212
Генерация системы 247
— ядра Супервизора 247

Емкость дорожки 72
— пакета дисков 75

Задание 35
Запись 67
— данных на дисках 72, 73
— логическая на перфоленте 71
— неопределенной длины 67
— описания дорожки 72, 73, 153
— переменной длины 67
— расширенная 67
— фиксированной длины 67

Знаковая строка 24

Императивные макрокоманды 86
— — индексно-последовательного доступа 114
— — последовательного доступа 103
— — прямого доступа 108
Имя перехода 22
— символическое простое 7
Инициализация диска 153
— магнитной ленты 162

Книга 124
Команда Ассемблера 15
— машинная 8
— DC 17
Контрольная точка 66
Копирование информации 159, 166, 178

Литерал 7
Логические устройства 31
— — программиста 31
— — системные 32

Макрокоманда 26
— генерации Супервизора 247
— — — ALLOC 254
— — — ASSGN 256
— — — CONFG 249
— — — DVCGEN 255
— — — FOPT 250
— — — IOTAB 255
— — — PIOCS 254
— — — SEND 256
— — — STDJC 249
— — — SUPVR 248
— связи с Супервизором 54
— — — — ATTACH 55
— — — — CANCEL 65
— — — — CCB 62
— — — — CHKPT 66
— — — — COMRG 59
— — — — DEQ 57
— — — — DETACH 56
— — — — DUMP 65

— — — — ENQ 56
— — — — EOJ 65
— — — — EXCP 65
— — — — EXIT 62
— — — — FETCH 58
— — — — GETIME 59
— — — — LOAD 58
— — — — MVCOM 59
— — — — PDUMP 65
— — — — POST 58
— — — — RCB 56
— — — — RELEASE 59
— — — — SETIME 60
— — — — STXIT 60
— — — — TECB 60
— — — — WAIT 57
— — — — WAITM 57
— СУВВ
— — CDMOD 89
— — CHECK 106
— — CLOSE(R) 107, 113, 122
— — CNTRL 105, 113
— — DAMOD 112
— — DAMODV 112
— — DIMOD 95
— — DTFCD 87
— — DTFCN 94
— — DTFDA 108
— — DTFDI 94
— — DTFIS 116
— — DTFMT 96
— — DTFPH 81
— — DTFPR 90
— — DTFPT 91
— — DTFSD 100
— — ENDFL 120
— — ERET 105, 121
— — ESETL 121
— — FEOV 107
— — FEOVD 107
— — FREE 107, 113, 122
— — GET 104, 121
— — ISMOD 119
— — LBRET 104
— — MTMOD 98
— — NOTE 106
— — OPEN(R) 104, 112, 120
— — POINTR 107
— — POINTS 107
— — POINTW 107
— — PRMOD 90
— — PRTOV 106
— — PTMOD 93
— — PUT 104, 121
— — READ 106, 113
— — RELSE 104
— — SDMODxx 102
— — SEOV 86
— — SETEL 120
— — SETL 121
— — TRUNC 104
— — WAITE 113, 121
— — WRITE 106, 113, 120
Макрокоманды перезаписи 184
— — для ввода 187
— — для вывода 188
— — использование регистров 185
— — операнды 189
— — управление макрокомандами вывода 186
Машинные команды 8
— — арифметики с плавающей точкой 11
— — арифметики с фиксированной точкой 8
— — десятичной арифметики 10
— — логические 9
— — пересылки 9
— — привилегированные 10
— — признаки результатов 13
— — проверка условий и передачи управления 10
— — расширенной мнемоники 13
— — сдвига 8
Метод доступа 86
— индексно-последовательного доступа 114
— последовательного доступа 86
— — — для файлов на дисках 99
— — — для файлов на магнитной ленте 95
— — — для файлов, независимых от устройств 94
— — — для файлов на перфоленте 91
— — — для файлов на перфокартах 86
— — — для файлов на печати 90
— — — для файлов на пишущей машинке 93
— прямого доступа 108
Модуль абсолютный 123
— исходный 124
— объектный 124

Назначение запасной дорожки 153, 155
— для логических устройств 37
— — — — временные 37
— — — — постоянные 37
— — — — стандартные 256
Начальная загрузка 28
— — директива ADD 30
— — директива DEL 30
— — директива SET 30
— — процедура 28

Область меток 142
— связи 58
— СЕ 218
Обработка меток 79
— — нестандартных 80

— — пользователя 79
— — системных 79
Обслуживание носителей данных 153, 178
Оглавление тома 79
— — вывод 157
Оператор макросредств 25
— языка Ассемблера 5
— модификации 154, 156, 159, 161, 163, 164, 169, 176
Отладка 144
— аварийное продолжение 145
— карты с корректирующей информацией 149
— оператор ATEOF 150
— оператор PCC 149
— оператор TR 147
— операторы корректировки 149
— операторы CSQ, PHQ 147
Отмена назначения запасной дорожки 155
Отношение 24
Отображение цилиндра меток 213
Очистка диска 158

Пакетная обработка 33
Параметр 21
Перезапись информации 159, 166, 175, 178, 184
Подготовка магнитных лент 153, 162
— пакетов дисков 153
Подстрока знаков 24
Поле идентификации 5
— названия 5
— операндов 5
— операции 5
Программы перезаписи 166, 175
— — задание на выполнение 169, 176
— — имена программ 168, 175
— — обработка меток 173, 178
— — оператор заглавия 173
— — оператор модификации 169, 176
— — оператор перекомпоновки 171, 177
— — стандартные значения параметров 172, 177
— — форматы записей 168, 169, 176
Протоколирование событий 215
— — последовательность вызова фаз 215
— — прерывания от ввода-вывода 215
— — прерывания по SVC 215
— — протоколы в памяти 225
— — протоколы на печати 224
Прототип 26

Разгрузка основной памяти 65
— — — макрокоманды DUMP, PDUMP 65
— — — программа Отладка 149
— — — режим DUMP 47
— — — системно-независимые программы 209
— транзитных областей 218
Регистрационный файл 226
— — записи 226
— — протокол 228
— — протокол записи CCH(EC-1035) 239
— — — — MCAR(EC-1035) 238
— — — — MCRR для машинных ошибок (EC-1020, EC-1022) 232
— — — — MCRR для машинных ошибок (EC-1030) 235
— — — — MCRR для ошибок канала (EC-1020, EC-1022) 233
— — — — MCRR для ошибок канала (EC-1030) 236
— — — — OBR 230
— — — — SDR 228
Редактирование 139
Редактор 139
— оператор ACTION 141
— — ENTRY 144
— — INCLUDE 143
— — PHASE 142
— режим AUTOLINK 141
— форматы адреса загрузки 142
Ресурс 56
— блок управления 56
— защита 56

Связь оператора с системой 34
Сервис 178
— вспомогательные функции 181
— дисковые функции 181
— задание на выполнение 179
— карточные функции 179
— ленточные функции 180
— управляющая информация 182
— управляющие операторы на перфокартах 181
Система управления вводом-выводом 80
— — — — логическая 86
— — — — физическая 80
Системно-независимые программы разгрузки 209
Собственный адрес 71, 72, 153
Создание страховочных копий 159
Сортировка-слияние 192, 199
— — выходы 196, 205
— — на дисках 199
— — на магнитных лентах 192
— — оператор END 196, 204
— — — INPFIL 195, 202
— — — MERGE 194, 202
— — — MODS 196, 204
— — — OPTION 195, 203
— — — OUTFIL 195, 203
— — — RECORD 194, 202

— — — SORT 194, 201
Сравнение лент 163
Стандартные имена модулей ввода-вывода 82
Стартер 256
— восстановление 261
— инициализация 260
— копирование 259
— специальные операторы 258
— управляющие операторы 257
Статистика тома 240
— — управляющая информация 243
Счетчик адреса 7

Терм 6
Трансляторы 138
Трансляция 137
— задание 139

Управляющая программа 28
— — управляющие операторы и директивы 35
— — ALLOC 37
— — ALTER 37
— — ASSGN 37
— — BATCH 38
— — CANCEL 38
— — CLOSE 38
— — DATE 39
— — DLBL 39
— — DSPLY 40
— — DUMP 40
— — DVCDN 40
— — DVCUP 40
— — EXEC 41
— — EXTENT 41
— — HOLD 42
— — JOB 42
— — LBLTYP 42
— — LISTIO 43
— — LOG 44
— — MAP 44
— — MODE 44
— — MSG 46
— — MTC 46
— — NOLOG 46
— — OPTION 46
— — PAUSE 48
— — READ 49
— — RELSE 49
— — RESET 49
— — RSTRT 49
— — SET 50
— — START 51
— — STOP 51
— — TIMER 51
— — TLBL 51
— — UCS 52
— — UNA 53
— — UNBATCH 53
— — UPSI 53
— — /* 54
— — /& 54
— — /* 54
— — KT 54
Управляющие символы 69
— — типа A 69
— — типа C 70
— — типа D 70
Участок на дисках 76
Учет заданий 244

Фаза 123
Файл 75
— исторический 226, 241
— на дисках 76
— на магнитной ленте 75, 77, 78
— на перфоленте 70
— регистрационный 226
Формат дорожки диска 71
— записи операторов Ассемблера 6
— машинной команды 8
Форматизация дорожек диска 153, 158

Характеристика 22
— длины 7, 23
Хроникер 225
— оператор OPTION 227
— регистрационный файл 226

Цилиндр меток 48
— — отображение 213
— — формат распечатки 214

Шаг задания 35
Шифры периферийных устройств 31

Языки программирования 137

ОГЛАВЛЕНИЕ

Предисловие 3
Глава 1. Ассемблер 5
1.1. Основные элементы языка Ассемблера 5
1.1.1. Алфавит 5
1.1.2. Операторы 5
1.1.3. Термы 6
1.1.4. Выражения 7
1.2. Машинные команды 8
1.3. Команды периферийных устройств 15
1.4. Команды Ассемблера 15
1.4.1. Общее описание 15
1.4.2. Команда DC 17
1.5. Макросредства Ассемблера 20
1.5.1. Параметры 21
1.5.2. Имя перехода 22
1.5.3. Характеристика 22
1.5.4. Знаковая строка 24
1.5.5. Подстрока знаков 24
1.5.6. Отношение 24
1.5.7. Выражения 24
1.5.8. Операторы макросредств 25
1.6. Отличия между языками Ассемблеров 27
Глава 2. Управляющая программа 28
2.1. Подготовка системы к работе 28
2.1.1. Процедура начальной загрузки 28
2.1.2. Директивы начальной загрузки 30
2.2. Управление заданиями 31
2.2.1. Логические устройства 31
2.2.2. Пакетная обработка 33
2.2.3. Выполнение одиночной программы 34
2.2.4. Организация связи оператора с системой 34
2.2.5. Управляющие операторы и директивы 35
2.3. Функции Супервизора 54
2.4. Макрокоманды связи с Супервизором 55
2.4.1. Мультиветвление 55
2.4.2. Защита ресурсов 56
2.4.3. Связь между задачами 57
2.4.4. Загрузка фаз 58
2.4.5. Доступ к области связи 58
2.4.6. Отмена временных назначений 59
2.4.7. Работа с таймером 59
2.4.8. Обработка прерываний проблемной программой . . 60
2.4.9. Выполнение операции ввода-вывода 62
2.4.10. Завершение программ (подзадач) 65
2.4.11. Разгрузка основной памяти 65

2.4.12. Создание контрольной точки 66
Глава 3. Управление данными 67
3.1. Организация данных 67
3.1.1. Форматы логических записей 67
3.1.2. Управляющие символы 69
3.1.3. Форматы данных на перфоленте 70
3.1.4. Формат и емкость дорожки дисков 71
3.1.5. Структура файлов на магнитной ленте и дисках . . . 75
3.1.6. Обработка меток 79
3.2. Система управления вводом-выводом 80
3.2.1 Физическая система управления вводом-выводом . . 80
3.2.2. Логическая система управления вводом-выводом . . 86
3.3. Метод последовательного доступа 86
3.3.1. Файлы на перфокартах 86
3.3.2. Файлы на печати 90
3.3.3. Файлы на перфоленте 91
3.3.4. Файлы на пишущей машинке 93
3.3.5. Файлы, независимые от устройств 94
3.3.6. Файлы на магнитной ленте 95
3.3.7. Файлы на дисках 99
3.3.8. Императивные макрокоманды 103
3.4. Метод прямого доступа 108
3.5. Метод индексно-последовательного доступа 114
Глава 4. Обслуживание библиотек 123
4.1. Библиотеки 123
4.1.1. Библиотека абсолютных модулей 123
4.1.2. Библиотека объектных модулей 124
4.1.3. Библиотека исходных модулей 124
4.1.4. Системные и личные библиотеки. Размер участков для библиотек 125
4.2. Библиотекарь 126
4.2.1. Назначение и функции 126
4.2.2. Способ использования 128
4.2.3. Управляющие операторы 129
4.2.4. Задания на выполнение конкретных функций 133
Глава 5. Трансляция и редактирование программ 137
5.1. Трансляция 137
5.1.1. Языки программирования 137
5.1.2. Трансляторы 137
5.1.3. Задание на трансляцию 139
5.2. Редактирование 139
5.3. Редактор 139
5.3.1. Назначение и возможности 139
5.3.2. Способ использования 140
5.3.3. Управляющие операторы 141
5.4. Отладка 144
5.4.1. Назначение и функции 144
5.4.2. Способ использования 145
5.4.3. Управляющие операторы 146
5.5. Примеры заданий на редактирование 150
5.5.1. Трансляция, редактирование с каталогизацией и выполнение 150
5.5.2. Трансляция, редактирование для отладки и выполнение 151
Глава 6. Обслуживание носителей данных 153
6.1. Инициализировать диск 153
6.1.1. Назначение и функции 153
6.1.2. Способ использования 154
6.1.3. Управляющие операторы 154
6.2. Назначить запасную дорожку 155
6.2.1. Назначение и возможности 155
6.2.2. Способ использования 156

6.2.3. Управляющая информация 156
6.3. Вывести оглавление тома 157
6.3.1. Назначение и функции 157
6.3.2. Способ использования 157
6.4. Очистить диск 158
6.4.1. Назначение и функции 158
6.4.2. Способ использования 158
6.4.3. Оператор модификации 159
6.5. Программы копирования и восстановления 159
6.5.1. Назначение и возможности 159
6.5.2. Способ использования 160
6.5.3. Оператор модификации 161
6.5.4. Обработка меток 161
6.6. Инициализировать ленту 162
6.6.1. Назначение и функции 162
6.6.2. Способ использования 162
6.6.3. Управляющие операторы 163
6.7. Сравнить ленты 163
6.7.1. Назначение и возможности 163
6.7.2. Способ использования 164
6.7.3. Оператор модификации 164
6.7.4. Подпрограмма пользователя для обработки несовпадающих блоков 165
Глава 7. Перезапись и сортировка информации 166
7.1. Программы перезаписи 166
7.1.1. Назначение и возможности 166
7.1.2. Способ использования 167
7.1.3. Управляющие операторы 169
7.1.4. Обработка меток 173
7.2. Программы перезаписи для перфоленты 175
7.2.1. Назначение и возможности 175
7.2.2. Способ использования 175
7.2.3. Управляющие операторы 176
7.2.4. Обработка меток 178
7.3. Сервис 178
7.3.1. Назначение и функции 178
7.3.2. Способ использования 179
7.3.3. Функции программы Сервис 179
7.3.4. Управляющие операторы на перфокартах 181
7.3.5. Управляющая информация (режим диалога) 182
7.4. Макрокоманды перезаписи 184
7.4.1. Назначение и возможности 184
7.4.2. Связь между макрокомандами ввода и вывода 185
7.4.3. Формат макрокоманд 186
7.5. Программа сортировки на магнитных лентах 192
7.5.1. Назначение и возможности 192
7.5.2. Способ использования 192
7.5.3. Управляющие операторы 193
7.5.4. Выходы к подпрограммам пользователя 196
7.6. Программа сортировки на магнитных лентах и дисках . . . 199
7.6.1. Назначение и возможности 199
7.6.2. Способ использования 200
7.6.3. Управляющие операторы 201
7.6.4. Использование программы сортировки в качестве подпрограммы 205
7.6.5. Выходы к подпрограммам пользователя 205
Глава 8. Протоколирование состояний и событий 209
8.1. Генератор программ разгрузки 209
8.1.1. Назначение и возможности 209
8.1.2. Системно-независимые программы разгрузки памяти 209
8.1.3. Способ использования 211

8.1.4. Управляющие операторы 211
8.2. Отображение цилиндра меток 213
8.2.1. Назначение и возможности 213
8.2.2. Способ использования 213
8.2.3. Результаты 214
8.3. Протоколирование событий 215
8.3.1. Назначение и функции 215
8.3.2. Способ использования 218
8.3.3. Управляющая информация 219
8.3.4. Результаты 223
8.3.5. Протоколы 224
8.4. Хроникер 225
8.4.1. Назначение и функции 225
8.4.2. Способ использования 226
8.4.3. Управляющие операторы 227
8.4.4. Протоколы 228
8.5. Статистика тома 239
8.5.1. Назначение и функции 239
8.5.2. Способ использования 240
8.5.3. Управляющая информация 241
8.5.4. Протокол 241
8.6. Интерфейс учета заданий 243
Глава 9. Генерация системы 246
9.1. Планирование системы 246
9.2. Макрокоманды генерации Супервизора 246
9.2.1. SUPVR 247
9.2.2. CONFG 248
9.2.3. STDJC 248
9.2.4. FOPT 249
9.2.5. PIOCS 253
9.2.6. ALLOC 253
9.2.7. IOTAB 254
9.2.8. DVCGEN 254
9.2.9. ASSGN 255
9.2.10. SEND 255
9.3. Стартер 255
9.3.1. Назначение и состав 255
9.3.2. Управляющие операторы 256
9.3.3. Специальные операторы для программы копирования 257
9.3.4. Специальные операторы для программы инициализации дисков 257
9.3.5. Программа копирования 258
9.3.6. Программа инициализации 259
9.3.7. Программа восстановления 260
Литература 261
Условные обозначения и принятые соглашения 262
Предметный указатель 264

Операционная система ДОС ЕС. Справочник
Рецензент *Байрон Сафеевич Хусаинов*
Редакторы *Л. Д. Григорьева и Л. А. Табакова*
Мл. редактор *В. К. Кипинская*
Техн. редактор *В. А. Чуракова*
Корректоры *Г. В. Хлопцева, О. Е. Константинова*
Худ. редактор *Н. А. Володина*
Переплет художника *Е. Н. Алексеева*
ИБ № 194

Сдано в набор 28/XII 1976 г. Подписано к печати 26/XII 1977 г.
Формат бумаги 60 × 90/16 Бумага № 1 Объем 17 печ. л. Уч. изд. л. 17,79
Усл. п. л. 17,0 Тираж 50 000 экз. (Допечатка тиража 20 000 экз.)
А14488 (Тематич. план 1977 г. № 94) Заказ № 2254 Цена 1 руб. 10 коп.

Издательство «Статистика», Москва, ул. Кирова, 39.
Отпечатано с матриц Московской типографии № 4 Союзполиграфпрома
при Государственном комитете Совета Министров СССР
по делам издательств, полиграфии и книжной торговли
г. Москва, И-41, Б. Переяславская ул., дом № 46

Scanned for lib.gva.org.ua

(c) 2025 gva.org.ua