

НИИ «НАУЧНЫЙ ЦЕНТР»

ПРОГРАММНОЕ  
ОБЕСПЕЧЕНИЕ  
ДВК

КНИГА 6

СИСТЕМНЫЕ БИБЛИОТЕКИ  
И СООБЩЕНИЯ СИСТЕМЫ

МОСКВА 1990 год

## АННОТАЦИЯ

Данная книга включает в себя три главы. Глава «СИСТЕМНАЯ МАКРОБИБЛИОТЕКА» описывает назначение и условия применения системной макробιβлиотеки и содержит описание программных запросов.

В главе «БИБЛИОТЕКА СИСТЕМНЫХ ПОДПРОГРАММ» содержится описание возможностей подпрограмм и способа их использования.

Подпрограммы библиотеки используют средства ОС ФОДОС-2, дополняют и расширяют ее возможности; обеспечивают выполнение действий аналогичных выполнению программных запросов; осуществляют обработку текстовых данных и целых чисел двойной точности, а также ряд вспомогательных операций. Обращение к подпрограммам библиотеки осуществляется на языке ФОРТРАН и языке АССЕМБЛЕР.

Глава «СООБЩЕНИЯ СИСТЕМЫ» содержит диагностические сообщения, которые выдает система ФОДОС-2, транслятор с языка ФОРТРАН и интерпретатор с языка БЕЙСИК-М, а также редактор текста (EDIT), экранный редактор текста (K13) и редактор командной строки (SL).

Каждое сообщение, описанное в главе приводится в той форме, в какой оно воспроизводится на терминале или в листинге.

Сообщения, выдаваемые системой, относятся как к программам пользователя, так и к ошибкам технических средств комплекса.

# СИСТЕМНАЯ МАКРОБИБЛИОТЕКА РУКОВОДСТВО ПРОГРАММИСТА

## 1. НАЗНАЧЕНИЕ И УСЛОВИЯ ПРИМЕНЕНИЯ

Ядром операционной системы ФОДОС-2 (главной управляющей программой) является монитор. В зависимости от конфигурации технических средств и потребностей пользователя может использоваться один из трех мониторов ФОДОС-2: монитор одного задания (SJ), монитор основного-фоновое задание (FB) и монитор расширенной памяти (XM). Все изложенное в данном документе, если не оговорено особо, относится к любому монитору ФОДОС-2.

Монитор выполняется параллельно с прикладными программами (программами пользователя) и обеспечивает выполнение следующих функций:

- инициализация и управление характеристиками (параметрами) операционной системы;
- распределение системных ресурсов и получение информации о состоянии операционной системы;
- интерпретация команд;
- операции с файлами;
- управление вводом-выводом;
- обеспечение связи между заданиями;
- завершение или временный приостанов программы;
- работа с расширенной памятью.

Связь пользователя с ФОДОС-2 осуществляется главным образом с помощью команд клавиатурного монитора и программных запросов.

Программные запросы обеспечивают разнообразные возможности для манипулирования файлами, выполнения операций ввода-вывода, управления выполнением и приостановкой программ и реализации других стандартных системных функций.

Программный запрос представляет собой макрокоманду, помещаемую пользователем в исходный текст программы на

языке АССЕМБЛЕР (программа может содержать несколько программных запросов).

Каждая макрокоманда заменяется последовательностью обычных символических команд. Этот процесс называется макрорасширением и выполняется во время ассемблирования.

Для формирования макрорасширения АССЕМБЛЕР использует макроопределения, включенные в системную макробиблиотеку SYSMAC.SML, которая находится на системном томе и может быть использована любой программой.

Большинство программных запросов используют машинную команду EMT, которая обеспечивает передачу управления монитору из программы, как только при выполнении этой программы очередь доходит до команды EMT.

Младший байт команды EMT содержит код в интервале от 340 до 377 (восьмеричное), определяющий местоположение передаваемых монитору аргументов (см. табл. 1).

Таблица 1

Значение младшего байта	Программный запрос
1	2
340—357 360—373 374	Программный запрос с аргументами в стеке и/или в R0 Используется только монитором ФОДОС-2 Программный запрос с одним аргументом. Регистр R0 содержит функциональный код в старшем байте и номер канала в младшем байте
375	Программный запрос с несколькими аргументами. Регистр R0 используется в качестве указателя на блок аргументов
376 377	Используется только монитором ФОДОС-2 Резервный. ФОДОС-2 игнорирует этот EMT и возвращает управление программе пользователя

## 2. ОБРАЩЕНИЕ К СИСТЕМНОЙ МАКРОБИБЛИОТЕКЕ

Для выполнения операций средствами операционной системы программы пользователя обращаются к монитору через программные запросы. В данном разделе описан формат и назначение программных запросов операционной системы.

Имена всех программных запросов начинаются символом «Точка» (.), что отличает их от макрокоманд, определяемых пользователем.

Все программные запросы, используемые в программе, должны быть определены в директиве .MCALL.

В ФОДОС-2 используются два формата программных запросов:

- 1) .PRGREQ ARG1, ARG2, ..., ARGN
- 2) .PRGREQ AREA, ARG1, ..., ARGN

В этих форматах PRGREQ представляет собой имя программного запроса, AREA — указатель на адрес блока аргументов, ARG1, ARG2, ..., ARGN — аргументы программного запроса.

Программные запросы первого формата вызывают генерацию команды EMT с кодом 374 или одной из команд EMT 340—357.

Программные запросы, использующие команду EMT 374, помещают в младший байт регистра R0 номер канала, а в старший — функциональный код (см. рис. 1).

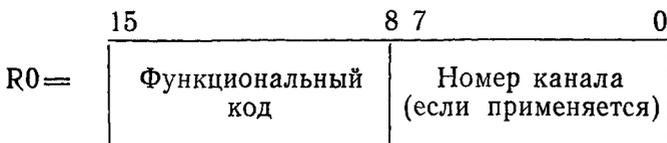


Рис. 1

Одним из программных запросов, использующих EMT 374, является запрос .DATE. Макроопределение для этого запроса следующее:

```
.MACRO .DATE
    MOV     #10.*^0400,%0
    EMT     ^374
.ENDM
```

Соответствующий запросу .DATE функциональный код, равный 10., помещается в старший байт регистра R0, а номер канала (0) — в младший байт регистра R0.

Программные запросы, использующие EMT 340—357, помещают аргументы (если они заданы) в стек и/или в R0.

Программный запрос .CSIGEN является примером программного запроса, использующего EMT 344. Ниже представлено упрощенное макроопределение для этого запроса:

```
.MACRO .CSIGEN DEVSPC,DEFEXT,CSTRNG,LINBUF
.IFF NB <LINBUF> MOV     LINBUF,—(6.)
    MOV     DEVSPC,—(6.)
.IFF NB <LINBUF> INC     (6.)
    MOV     DEFEXT,—(6.)
.IF B CSTRNG
    CLR     —(6.)
.IFF
```

```

.IF IDN  CSTRNG,#0
        CLR          —(6.)
.IFF
        MOV          CSTRNG,—(6.)
.ENDC
.ENDC
        EMT          ^0344
.ENDM

```

Все указанные аргументы помещаются в стек пользователя (см. рис. 2) и используются монитором при выполнении запроса .CSIGEN.

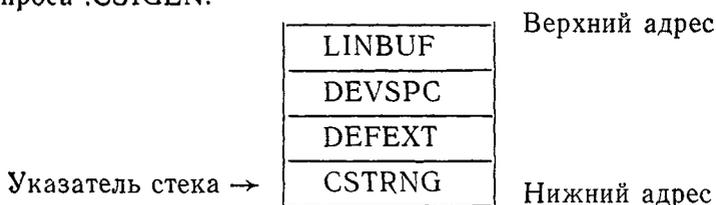


Рис. 2

Программные запросы второго формата вызывают генерацию команды EMT с кодом 375.

Формат блока аргументов для EMT 375 представлен на рис. 3.

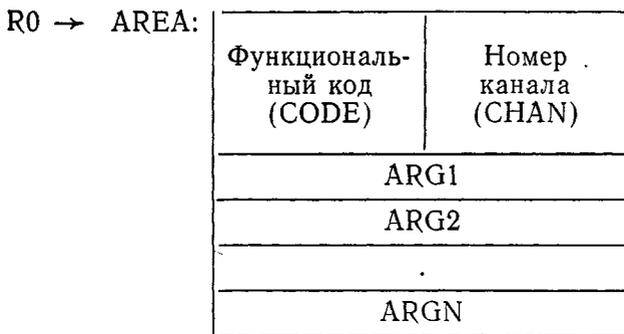


Рис. 3

Некоторые аргументы программных запросов могут быть опущены. Если отсутствует аргумент AREA, то предполагается, что он содержится в R0. Если не указаны (опущены) какие-либо из аргументов от ARG1 до ARGN, то содержимое соответствующих этим аргументам слов в блоке аргументов не изменяется.

Например, при выполнении запроса

```
.PRGREQ AREA,ARG1,ARG2
```

регистр R0 содержит адрес блока аргументов, второе слово содержит ARG1, третье — ARG2.

При выполнении запроса

```
.PRGREQ AREA
```

старший байт первого слова блока аргументов содержит функциональный код. Содержимое других слов блока аргументов не изменяется.

Если в запросе

```
.PRGREQ, ARG1
```

опущен аргумент AREA, то предполагается что он содержится в R0 (функциональный код в этом случае в блок аргументов не заносится).

Блок аргументов может формироваться МАКРОАССЕМБЛЕРОМ во время трансляции программы или непосредственно пользователем, например:

- 1) MOV #ARG1,AREA+2  
MOV #ARG2,AREA+4  
.PRGREQ #AREA
- 2) MOV #AREA,R0  
.PRGREQ #ARG1,#ARG2
- 3) MOV #AREA,R0  
MOV #ARG1,2(R0)  
MOV #ARG2,4(R0)  
MOV #CODE\*400!CHAN,(R0)  
.PRGREQ

Во всех вышеприведенных примерах определяется один и тот же программный запрос.

В примере, который следует ниже, приведен один из возможных способов формирования блока аргументов для программного запроса .TWAIT.

```
        .MCALL .PRINT, .TWAIT
START:  .TWAIT #EMTBLK
        .PRINT #MSG
        BR     START
EMTBLK: .BYTE  0,24
        .WORD  TIME
TIME:   .WORD  0,10.*50
        .ASCIZ (через 10 секунд — печать сообщения)
        .END   START
```

## 2.1. Адресация аргументов

Все аргументы программных запросов используются в ка-

честве источника в команде MOV и блок аргументов должен содержать непосредственно значения аргументов.

Поэтому необходимо помнить, что аргументы программных запросов должны быть допустимыми в языке АССЕМБЛЕРА и точно представлять требуемую величину. Если величина является константой или символическим именем пользователя, то должен быть использован непосредственный метод адресации (#), например:

```
.PRGREQ #AREA,#4
```

Если величина является содержимым регистра, то должна быть использована мнемоника регистра (RN). Если величина является содержимым ячейки памяти, то должна быть использована метка этой ячейки, например:

```
.PRGREQ LIST,NUMBER
```

```
LIST:      .WORD      AREA  
NUMBER:   .WORD      4
```

**ПРИМЕЧАНИЕ.** При выполнении запроса, содержимое регистров общего назначения, кроме R0, не изменяется. Указатель стека не переустанавливается. Исключением является вызов интерпретатора командной строки (запросы .CSIGEN, .CSISPC).

**2.2. Ключевой аргумент CODE** позволяет использовать один и тот же блок аргументов для различных программных запросов, имеющих одни и те же аргументы, но различные функциональные коды. Использование этого ключевого аргумента уменьшает размер программы и время ее выполнения. Аргумент CODE может иметь два значения: CODE=SET (занести функциональный код) и CODE=NOSET (не заносить функциональный код).

**Пример:**

```
.WRITW #AREA,#CHAN,#BUFF,#WCNT,#BLK
```

```
.  
.  
.
```

```
.READW #AREA,CODE=SET
```

```
.  
.  
.
```

### 3. ВХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ

Аргументы программных запросов являются входными данными для монитора ФОДОС-2. В табл. 2 приведены их условные обозначения. В этом разделе также приведены некоторые основные понятия, которые используются при обращении к МАКРОБИБЛИОТЕКЕ.

Таблица 2

Условное обозначение аргумента	Назначение аргумента
1	2
ADDR	Адрес; значение адреса зависит от используемого запроса
AREA	Адрес блока аргументов EMT
BLK	Номер блока относительно начала файла, с которого должна выполняться операция ввода-вывода
BUF	Адрес буфера памяти
CBLK	Адрес блока из пяти слов, в котором хранится информация о состоянии канала
CHAN	Номер канала
CRTN	Точка входа подпрограммы завершения
DBLK	Адрес блока устройства
NUM	Число, значение которого зависит от используемого запроса
WCNT	Счетчик слов; количество слов, которые необходимо передать в/из буфера при выполнении операции ввода-вывода
SEQN	Последовательный номер файла (для операций с магнитной лентой)
CHRCNT	Счетчик символов
FUNC	Числовой код, определяющий выполняемую функцию
JOBBLK	Адрес блока, содержащего имя системного задания в кодах КОИ-7
JOBDEV	Адрес блока, содержащего имя устройства и имя системного задания
UNIT	Логический номер терминала

#### 3.1. Номер канала

Номер канала (CHAN) представляет собой логический идентификатор файла и используется монитором при выполнении запрашиваемых операций. Номер канала может иметь значение от 0 до 254 (десятичное) (0—376 (восьмеричное)). Канал с номером 255 (десятичное) резервируется для системных операций.

Каждый открытый файл монитор логически связывает с определенным каналом (элементом структуры монитора, точно определяющим файл в операционной системе). Для обра-

щения к файлу достаточно указать номер соответствующего канала.

**3.2. Блок устройства (DBLK)** представляет собой блок информации в кодах RADIX-50, который содержит физическое или логическое имя устройства, имя и тип файла. Размер блока — 4 слова.

**Пример:**

```
.RAD50/DX1/  
.RAD50/FIL/  
.RAD50/NAM/  
.RAD50/TYP/
```

Имя устройства, имя и тип файла должны быть левоопределенными в каждом поле операнда (т. е. непосредственно следовать за ограничителем операнда директивы .RAD50).

Допускается запись блока устройства в следующем виде: .RAD50/DX1FILNAMTYP/

При этом каждое пропущенное поле операнда должно заменяться пробелом.

### 3.3. Блок аргументов EMT

Общий вид блока аргументов команды EMT показан на рис. 4. Четный байт 1-го слова содержит номер канала, определенный в макрокоманде. Если номер канала не требуется, то четный байт содержит 0. Нечетный байт 1-го слова блока содержит функциональный код макрокоманды (каждому программному запросу соответствует свой функциональный код). Остальные слова блока содержат аргументы (если они заданы), определенные в макрокоманде.

#### Блок аргументов EMT

AREA:

Функциональный код	Номер канала
1-й аргумент	
2-й аргумент	
⋮	
N-ый аргумент	

Рис. 4

**3.4. Подпрограмма завершения** создается пользователем и выполняется при наступлении события. Событие может за-

ключаться в завершении операции ввода-вывода, может быть вызвано прерыванием от внешнего устройства, истечением заданного интервала времени и т. д.

На входе в подпрограмму завершения регистр R0 содержит слово состояния канала, а R1 — восьмеричный номер канала для операции.

В режиме SJ подпрограммы завершения являются асинхронными и могут прерывать друг друга. В режимах FB и XM, подпрограммы завершения не прерывают друг друга, а организуются в очередь.

### 3.5. Требование USR

Для выполнения некоторых программных запросов необходимо, чтобы компонента монитора USR находилась в оперативной памяти. Требование USR к программным запросам описано в табл. 3. Заметим, что запрос CLOSE на устройст-

Таблица 3

Программный запрос	Требование USR		
	2	3	4
	Режим SJ	Режим FB	Режим XM
1			
.CDFN	*		
1)	Да	Нет	Нет
.CLOSE	Да	Да	Да
.CSIGEN	Да	Да	Да
.CSISPC	Да	Да	Да
.DELETE	Да	Да	Да
.DSTATUS	Да	Да	Да
.ENTER	Да	Да	Да
.EXIT	Да	Да	Да
.FETCH/.RELEASE	Да	Да	Да
.GTLIN	Да	Да	Да
.HRESET	Да	Нет	Нет
2)			
.LOCK	Да	Да	Да
.LOOKUP	Да	Да	Да
	*	*	
.QSET	Да	Да	Да
.RENAME	Да	Да	Да
.REOPEN	Нет	Нет	Нет
	*		
.SRESET	Да	Нет	Нет
3)			
.TLOCK	Да	Да	Да

1) если канал был открыт по .ENTER;

2) если USR находится в состоянии свопинга;

3) если USR не используется другим заданием.

вах нефайловой структуры (LP:, TT:, PC:, и т. д.) не требует наличия **USR** в памяти.

Программа **USR** не является реентерабельной; если **USR** используется одним заданием (в режиме **FB**), другое задание, требующее **USR**, будет ждать ее освобождения.

Запросы, помеченные в табл. 3 звездочкой, всегда требуют загрузки новой копии **USR** перед их выполнением (даже, если **USR** в этот момент находится в памяти).

#### 4. СООБЩЕНИЯ

При обработке запросов, монитор может обнаружить условия ошибок. Программные запросы информируют об этом пользователя, используя **C**-разряд, байт ошибок (52-ой байт в области связей системы) и печать на терминале сообщений монитора. **C**-разряд очищается после выполнения запроса и устанавливается в случае ошибки. Для обнаружения ошибки за запросом должна следовать команда **VCS** или **VCC**. Когда установлен **C**-разряд, байт ошибок содержит код ошибки.

При возникновении серьезных ошибок (в этом случае монитор печатает соответствующее сообщение на терминале) управление передается монитору. Для анализа подобных ошибок рекомендуется использовать программный запрос **.SERR**. В приложении 1 перечислены коды ошибок программных запросов и причины, вызывающие эти ошибки.

Информация о сообщениях дана в [3].

#### 5. ПРОГРАММНЫЕ ЗАПРОСЫ

Существуют следующие типы программных запросов:

- запросы для операций с файлами;
- запросы для организации ввода-вывода данных;
- запросы общего назначения;
- запросы расширенной памяти;
- запросы для драйверов.

В табл. 4 перечислены программные запросы, которые могут использоваться в любом из режимов (**SJ**, **FB**, **XM**). Запросы, помеченные в табл. 4 символами «**MT**», можно использовать только при наличии в системе мультитерминальной поддержки.

В табл. 5 перечислены программные запросы, которые используются основным/фоновым заданием при наличии расширенной памяти. Программные запросы, помеченные символами «**XM**» можно использовать только в режиме **XM**.

Описание программных запросов для программ — драйверов устройств дается в [1].

Описание запросов расширенной памяти дается в [2].

Таблица 4

Программный запрос	Код EMT	Функциональный код	Назначение
1	2	3	4
.ABTIO	374	13	Прерывает операции ввода-вывода на указанном канале
.CDFN	375	15	Определяет дополнительные каналы для операций ввода-вывода
.CHAIN	374	10	Передает управление другой фоновой программе
.CLOSE	374	6	Закрывает канал и освобождает его для использования в других операциях
.CMKT	375	23	Аннулирует один или несколько запросов .MRKT
.CSIGEN	344	—	Вызывает работу интерпретатора командной строки в общем режиме
.CSISPC	345	—	Вызывает работу интерпретатора командной строки в специальном режиме
.CSTAT	375	27	Позволяет получить информацию о состоянии определенного канала
.CTIMIO	—	—	Используется в программах-драйверах устройств для отмены запросов, требующих наличия таймера
.DATE	374	12	Пересылает текущую дату в регистр R0
.DELETE	375	0	Стирает файл с тома указанного устройства
.DRAST			Определяет начало подпрограммы обработки прерываний
.DRBEG			Создает таблицу заголовка драйвера
.DRBOT			Определяет начало первичного загрузчика в драйвере
.DRDEF			Определяет параметры драйвера и устройства
.DREND			Формирует таблицу адресов подпрограмм монитора; определяет логический конец текста драйвера
.DRFIN			Передает управление монитору по завершении обслуживания элемента очереди
.DRSET			Используется для формирования таблиц параметров драйвера
.DRVTB			Используется для описания таблицы векторов прерываний и подпрограмм их обработки

1	2	3	4
.DSTATUS	342	—	Позволяет получить информацию о состоянии определенного устройства
.ENTER	375	2	Резервирует область на томе указанного устройства для создания нового файла
.EXIT	350	—	Завершает выполнение программы пользователя и передает управление клавиатурному монитору
.FETCH	343	—	Загружает драйвер устройства
.FPROT	375	43	Устанавливает (отменяет) защиту файла
.GTIM	375	21	Определяет текущее время дня
.GTJB	375	20	Определяет параметры выполняемого задания
.GTLIN	345	—	Принимает входную строку с терминала или из косвенного файла
.GVAL	376	34	Пересылает в R0 содержимое указанной ячейки монитора
.HERR	374	5	Отменяет действие запроса .SERR
.HRESET	357	—	Прекращает все операции ввода-вывода и выполняет функции запроса .SRESET
.INTEN	—	—	Сообщает монитору о том, что произошло прерывание от устройства и устанавливает приоритет процессора на соответствующий уровень
.LOCK	346	—	Делает программу USR резидентной в памяти до тех пор, пока не будет выполнен .EXIT или .UNLOCK. Если необходимо, свопирует программу пользователя
.LOOKUP	375	1	Открывает существующий файл для ввода и/или вывода на указанном канале
.MFPS	—	—	Выбирает содержимое ССП
.MRKT	375	22	Планирует запуск подпрограммы завершения через определенный интервал времени
.MTATCH(MT)	374	37	Устанавливает связь между заданием и определенным терминалом
.MTDTCH(MT)	374	37	Прерывает связь указанного терминала с заданием и освобождает его для использования другим заданием
.MTGET(MT)	374	37	Определяет состояние указанного терминала
.MTIN(MT)	374	37	Действует в мультитерминальном режиме аналогично .TTYIN
.MTOUT(MT)	374	37	Действует в мультитерминальном режиме аналогично .TTYOUT

1	2	3	4
.MTPRNT (MT)	374	32	Действует в мультитерминальном режиме аналогично PRINT
.MTPS	—		Устанавливает ССП
.MTRCTD (MT)	374	37	Обеспечивает вывод на указанный терминал
.MTSET (MT)	374	37	Устанавливает и модифицирует характеристики терминала в мультитерминальном режиме
.MTSTAT (MT)	375	37	Пересылает программе информацию о состоянии мультитерминальной системы
.PEEK	375	34	Определяет содержимое ячеек памяти
.POKE	375	34	Изменяет содержимое ячеек памяти
.PRINT	351	—	Выдает на печать строку символов КОИ-7
.PURGE	374	3	Освобождает канал
.PVAL	375	34	Изменяет содержимое ячеек монитора
.QELDF	—	—	Определяет мнемонические обозначения для элементов очереди ввода-вывода
.QSET	353	—	Увеличивает размер очереди ввода-вывода для монитора
.RCTRLD	355	—	Обеспечивает вывод на терминал
.READ	375	10	Передает данные по указанному каналу в буфер памяти и возвращает управление программе пользователя, как только запрос на чтение данных будет поставлен в очередь ввода-вывода
.READC	375	10	Передает данные по указанному каналу в буфер памяти и возвращает управление программе пользователя, как только запрос на чтение данных будет поставлен в очередь ввода-вывода. После окончания чтения управление передается подпрограмме завершения
.READW	375	10	Передает данные по указанному каналу в буфер памяти и возвращает управление программе пользователя после окончания операции
.RELEASES	343	—	Удаляет драйвер устройства из памяти
.RENAME	375	4	Переименовывает файл. Недопустим в операциях с магнитной лентой
REOPEN	375	6	Обеспечивает восстановление параметров, которые запомнились по запросу .SAVSTATUS и повторно открывает канал для ввода-вывода

1	2	3	4
.SAVESTATUS	375	5	Запоминает параметры состояния открытого файла в определенной области памяти и освобождает канал
.SCCA	374	35	Позволяет блокировать выполнение команды СУ/С
.SDTTM	375	40	Устанавливает системную дату и/или время
.SERR	374	4	Блокирует прерывание программы при обнаружении монитором ошибки
.SETTOP	354	—	Определяет верхнюю границу памяти для выполняемого задания
.SFDAT	375	42	Изменяет дату создания файла
.SFPA	375	30	Устанавливает в программе пользователя адрес подпрограммы обработки прерывания от процессора плавающей запятой
.SPFUN	375	32	Выполняет специальные функции для устройств
.SRESET	352	—	Восстанавливает каналы ввода-вывода и удаляет драйверы устройств из памяти
.SYNCH	—	—	Обеспечивает возможность выполнения программных запросов в подпрограмме обработки прерывания
.TIMIO	—	—	Планирует запуск подпрограммы завершения через заданный интервал времени
.TLOCK	374	7	Определяет, используется ли USB в данный момент другим заданием. Если USB не используется, то выполняется .LOCK
.TRPSET	375	3	Устанавливает обработку прерываний по 4-ому или 10-ому векторам в программе пользователя
.TTINR	340	—	Передают один символ из буфера терминала в регистр R0
.TTYIN	341	—	Передают один символ из регистра R0 в буфер терминала
.TTYOUT	375	24	Приостанавливает выполнение задания на указанный интервал времени
.TTOUTR	347	—	Освобождает программу USB, если ранее был выполнен запрос .LOCK и, если необходимо, вновь загружает программу пользователя
.UNLOCK	374	0	Ожидает завершения операции ввода-вывода на указанном канале
.WAIT	375	11	Передаёт данные по указанному каналу на устройство и возвращает управление программе пользователя, как только запрос на передачу данных
.WRITC			

1	2	3	4
.WRITE	375	11	будет поставлен в очередь ввода-вывода. После окончания записи, управление передается подпрограмме завершения
.WRITW	375	11	Передаёт данные по указанному каналу на устройство и возвращает управление программе пользователя, как только запрос на передачу данных будет поставлен в очередь ввода-вывода Передаёт данные по указанному каналу на устройство и возвращает управление программе пользователя после окончания передачи данных

Таблица 5

Программный запрос	Код ЕМТ	Функциональный код	Назначение
1	2	3	4
.CHCOPY	375	13	Открывает канал для ввода и логически связывает его с открытым каналом другого задания
.CNTXSW	375	33	Определяет ячейки памяти для операции переключения контекста
.CRAW(XM)	374	36	Создает виртуальное адресное окно
.CRRG(XM)	374	36	Создает динамическую область физической памяти
.DEVICE	375	14	Позволяет запретить прерывание от устройства в режиме FB после окончания выполнения программы
.ELAW(XM)	374	36	Аннулирует виртуальное адресное окно
.ELRG(XM)	374	36	Аннулирует динамическую область физической памяти и делает ее свободной для использования другим заданием
.GMCX(XM)	374	36	Пересылает программе информацию о состоянии определенного окна
.MAP(XM)	374	36	Отображает предварительно созданное окно виртуального адреса
.MWAIT	374	11	Переводит текущее задание в состояние ожидания до завершения передачи данных
.PROTECT	375	31	Закрепляет один из векторов (адреса от 0 до 476) за данным заданием

1	2	3	4
.PCVD .PCVDC .RCVDW	375	26	Осуществляют прием сообщений или данных, передаваемых другим заданием в режиме FB. Аналогичны запросам .READ, .READC, .READW
.RDBBK(XM)	—	—	Создает блок определения области
.RDBDF(XM)	—	—	Определяет мнемонические обозначения для блока определения области
.RSUM	374	2	Возобновляет выполнение задания, которое было прервано по запросу .SPND
.SDAT .SDATC .SDATW	375	25	Осуществляют передачу сообщений или данных другой программе. Аналогичны запросам .WRITE, .WRITC, .WRITW
.SPCPS	375	41	Позволяет изменить адрес возврата из прерывания в подпрограмме завершения
.SPND	374	1	Приостанавливает выполнение задания
.UNMAP(XM)	374	36	Отменяет действие запроса .MAP
.UNPROTECT	374	31	Отменяет действие запроса .PROTECT
.WDBBK(XM)	—	—	Создает блок определения окна
.WDBDF(XM)	—	—	Определяет мнемонические обозначения для блока определения окна

## 5.1. Программные запросы для операций с файлами

### 5.1.1. .CHCOPY (режим FB, XM)

Запрос .CHCOPY открывает канал для ввода данных и логически связывает его с каналом другого задания, открытым для ввода или вывода. Запрос может использоваться основной, фоновой программами, а также системным заданием. Запрос .CHCOPY допустим только для файлов, которые находятся на дисках; однако система не обнаружит ошибки, если используется другое устройство. Для того, чтобы закрыть

канал, открытый по .CHCOPY, необходимо использовать запросы .CLOSE или .PURGE.

Формат макрокоманды:

.CHCOPY AREA,CHAN,OCHAN[,JOBBLK]

где AREA — адрес блока из 3-х слов аргументов EMT;

CHAN — номер канала, используемого заданием для чтения данных;

OCHAN — номер канала другого задания;

JOBBLK — адрес блока из трех слов, содержащего логическое имя задания.

Формат блока аргументов EMT:

```
AREA: .BYTE CHAN,13
      .WORD OCHAN
      .WORD JOBBLK
```

### Пример:

; пример состоит из двух программ. Первая из программ (основная) выполняет передачу данных во вторую — фоновую программу.

.TITLE CHCOPF.MAC

; основная программа

```
      .MCALL .ENTER, .PRINT, .SDATW, .RCVDW, .CLOSE, WRITW
      .MCALL .EXIT
STARTF: MOV #AREA,R5
      .ENTER R5,#0,#FILE,#5 ;создать файл
      .WRITW R5,#0,#RECRD,#256 ,#4 ;записать блок
      BCS ENTERR
      .SDATW R5,#BUFR,#2 ;передать сообщение
      ;
      .RCVDW R5,#BUFR,#1
      .CLOSE #0
      .PRINT #FEXIT
      .EXIT
ENTERR: .PRINT #ERMSG
      .EXIT
FILE: .RAD50 /DK QUFILB/
      .RAD50 /TMP/
ARE/: .BLKW 5
BUFR: .WORD 0 ;номер канала
      .WORD 4 ;номер блока
RECRD: .BLKW 256.
ERMSG: .ASCIZ /ошибка .ENTERR/
FEXIT: .ASCIZ /конец основной программы/
      .END STARTF
      .TITLE CHCOPF.MAC
; фоновая программа
      .MCALL .CHCOPY, .RCVDW, .READW, .EXIT, .PRINT, .SDATW
STARTF: MOV #AREA,R5
      .RCVDW R5,#MSG,#2 ;получить сообщение
      BCS 1$
```

```

.CHDCOPY R5,#0,MSG+2      ;номер канала из MSG+2
BCS      2$
.READW   R5,#0,#BUFF,#256,MSG+4 ;читать блок
BCS      3$
;
.SDATW   R5,#MSG,#1
.PRINT   #BEXIT
.EXIT
1$:      MOV     #NOJOB,R0
         BR     4$
2$:      MOV     #NOCH,R0
         BR     4$
3$:      MOV     #RDERR,R0
4$:      .PRINT
         .EXIT
AREA:    .BLKW   5
MSG:     .BLKW   3          ;буфер для сообщения
BUFF:    .BLKW   256.
BEXIT:   .ASCIZ  /конец фоновой программы/
NOJOB:   .ASCIZ  /нет основного задания/
NOCH:    .ASCIZ  /не открыт канал для основного задания/
RDERR:   .ASCIZ  /ошибка чтения/
.END     STARTB

```

5.1.2. Запрос **.CLOSE** завершает действие на указанном канале и освобождает его для использования в другой операции. Для выполнения запроса драйвер соответствующего устройства должен находиться в памяти.

**Формат макрокоманды:**

**.CLOSE CHAN**

Формат блока аргументов EMT:

AREA: .BYTE CHAN,6

Для того, чтобы сделать файл постоянным на томе необходимо на соответствующем канале, открытом для вывода, выполнить запрос **.CLOSE**. Запрос **.CLOSE** на неоткрытом канале игнорируется.

**.CLOSE**, выполняемый на канале, который был открыт по **.ENTER**, вызывает корректировку справочника устройства для определения файла как постоянного на этом устройстве. Если файл открыт по **.LOOKUP**, то корректировки справочника не происходит.

Если устройство, связанное с указанным каналом, уже содержит файл с тем же именем и типом, что и вновь созданный файл, то старая копия файла стирается, когда новый файл становится постоянным.

Если на устройстве уже существует защищенный файл с тем же именем, то текущий файл закрывается, но появляются два файла с одинаковыми именами.

**ПРИМЕЧАНИЕ.** Монитор информирует программу пользователя об ошибке при выполнении запроса `.CLOSE` только в том случае, если до этого был выполнен запрос `.SERR`.

Пример использования `.CLOSE` см. в описании запроса `.CSISPC`.

**5.1.3. Запрос `.DELETE`** удаляет файл из справочника указанного устройства. `.DELETE` недопустим для магнитных лент.

Формат макрокоманды:

`.DELETE AREA,CHAN,DBLK`

где `AREA` — адрес блока из 2-х слов аргументов `EMT`;

`DBLK` — адрес блока устройства, содержащего спецификацию удаляемого файла.

**ПРИМЕЧАНИЯ:**

1. Обращение к открытому каналу приводит к ошибке;
2. Удаленный файл в листинге справочника устройства будет описан как неиспользуемая область (`UNUSED`) того же размера;
3. Запрос `.DELETE` на устройствах нефайловой структуры игнорируется;
4. При выполнении `.DELETE` необходимо, чтобы драйвер соответствующего устройства находился в памяти;
5. После выполнения `.DELETE`, указанный канал освобождается для дальнейшего использования.

Пример использования `.DELETE` см. в описании запроса `.CSISPC`.

**5.1.4. Запрос `.ENTER`** резервирует область на определенном устройстве и заносит в справочник устройства запись о создании временного файла. По запросу `.ENTER` монитор идентифицирует номер канала, определенный в запросе, с указанным файлом. Заметим, что если программа имеет оверлейную структуру, то 15-й канал используется оверлейным драйвером и не должен модифицироваться программой пользователя.

Формат макрокоманды:

`.ENTER AREA,CHAN,DBLK,LEN[,SEQN]`

где `AREA` — адрес блока из 4-х слов аргументов `EMT`;

`DBLK` — адрес блока устройства, в котором описан создаваемый файл;

`LEN` — размер файла; (см. табл. 6);

`SEQN` — последовательный номер файла для магнитной ленты; значения аргумента `SEQN` и соответствующие им действия монитора описаны в табл. 7.

. Формат блока аргументов EMT:  
 AREA: .BYTE CHAN,2  
 .WORD DBLK  
 .WORD LEN  
 .WORD SEQN

Таблица 6

Значение аргумента	Действие монитора
1	2
0	Отводит для файла половину наибольшей свободной области или полностью вторую по величине свободную область на томе, в зависимости от того, какая из них больше
M	Отводит для файла M блоков
-1	Отводит для файла наибольшую свободную область на томе

Таблица 7

Значение аргумента	Действие монитора
1	2
0	Перематывает магнитную ленту и производит поиск метки LEOT (логический конец ленты)
N	Устанавливает магнитную ленту в положение, соответствующее файлу с последовательным номером N
-1	Производит поиск LEOT
-2	Перематывает магнитную ленту и производит поиск LEOT или файла с тем же именем, что и вновь создаваемый файл

Файл, созданный по .ENTER не является постоянным до тех пор, пока на соответствующем канале не будет выполнен запрос .CLOSE. Запросы .LOOKUP и .SAVSTATUS не действительны для файлов, которые не являются постоянными. Однако чтение такого файла возможно посредством обращения к номеру соответствующего блока. Если на соответствующем канале выполняется .CLOSE, то любой существующий на томе постоянный, незащищенный файл с тем же именем и типом стирается и новый файл становится постоянным. Хотя область для файла резервируется при выполнении .ENTER, фактическая длина файла определяется после выполнения .CLOSE.

Каждое задание может иметь до 255 одновременно открытых файлов. Если требуется, все 255 файлов могут быть открыты для вывода посредством .ENTER. Для выполнения .ENTER необходимо, чтобы драйвер соответствующего устройства был в памяти. После выполнения .ENTER регистр R0 будет содержать размер области тома, отведенный для создания нового файла.

**Пример:**

.TITLE ENTER.MAC  
 ; в следующем примере с использованием запроса .ENTER  
 ; выполняется копирование файла ENTER.SAV на устройство  
 ; DK: .

```

.MCALL .LOOKUP,.ENTER,.WRITW,.READW,.CLOSE
.MCALL .PRINT,.EXIT
ERRBYT = 52
START: .LOOKUP #AREA,#0,#ENTER ;связать 0-ой канал
;с файлом ENTER.SAV
      BCS 5$
      MOV R0,R3 ;размер файла в R3
      .ENTER #AREA,#1,#NEWFIL,R3 ;открыть новый файл
      BCS 6$
1$:   CLR BLK
      .READW #AREA,#0,#BUFFR,#256,BLK ;читать блок
      BCC 2$
      TSTB @ #ERRBYT ;конец файла?
      BEQ 3$ ;да
      MOV #RERR,R0
      BR 7$
2$:   .WRITW #AREA,#1,#BUFFR,#256,BLK ;записать блок
      INC BLK
      BCC 1$
      MOV #WERR,R0
      BR 7$
3$:   .CLOSE #1
      MOV #DONE,R0
      BR 7$
5$:   MOV #NOFIL,R0
      BR 7$
6$:   MOV #NOENT,R0
7$:   .PRINT
      .EXIT
AREA: .WORD 0
BLK: .WORD 0,0,0,0
BUFFR: .BLKW 256.
ENTER: .RAD50 /DK/
      .RAD50 /ENT/
      .RAD50 /ER/
      .RAD50 /SAV/
NEWFIL: .RAD50 /DK/
      .RAD50 /NEWENT/
      .RAD50 /SAV/
NOFIL: .ASCIZ /файл не найден/

```

```

NOENT: .ASCIZ /ошибка .ENTER/
WERR: .ASCIZ /ошибка .WRITW/
RERR: .ASCIZ /ошибка .READW/
DONE: .ASCIZ /копирование завершено/
      .EVEN
      END START

```

### 5.1.5. LOOKUP

Используются две формы запроса .LOOKUP. Первая форма .LOOKUP связывает указанный канал с устройством и/или существующим файлом для выполнения операций ввода-вывода. Используемый канал будет «занят» до тех пор, пока не будет выполнен один из следующих запросов: .CLOSE, .SAVESTATUS, .SRESET, .HRESET, .PURGE, .CSIGEN (если номер канала в диапазоне от 0 до 10 (восьмеричное)).

Заметим, что если программа имеет оверлейную структуру, то 15-ый канал используется оверлейным драйвером и не должен изменяться программой пользователя.

Если первое слово имени файла в блоке устройства (DBLK) содержит ноль и устройство есть устройство файловой структуры, то абсолютный блок 0 этого устройства определяется как «Начало файла» (в этом случае говорят о нефайловом .LOOKUP). Этот метод допускает ввод-вывод для любого физического блока на устройстве. Если имя файла определено для устройства нефайловой структуры, то имя файла игнорируется.

Для выполнения .LOOKUP драйвер соответствующего устройства должен быть в памяти. После выполнения .LOOKUP, регистр R0 будет содержать длину открытого файла в блоках. После выполнения .LOOKUP для устройств несправочной структуры, регистр R0 будет содержать ноль.

Формат макрокоманды:

```
.LOOKUP AREA,CHAN,DBLK[,SEQN]
```

где AREA — адрес блока из 3-х слов аргументов EMT; SEQN — последовательный номер файла для магнитной ленты; значения, которые может иметь SEQN и соответствующие им действия монитора описаны в табл. 8.

Формат блока аргументов EMT:

```

AREA: .BYTE CHAN,1
      .WORD DBLK
      .WORD SEQN

```

**Пример 1:**

```
.TITLE LOOKUP.MAC
```

; в данном примере, использующем запрос .LOOKUP, определяется наличие на устройстве DK: рабочего файла, содержащего очередь файлов для вывода на указанное устройство.

; Если рабочий файл существует, то его размер (в блоках)  
; печатается на системном терминале.

```

START:   .MCALL   .LOOKUP,.PRINT,EXIT
         .LOOKUP  #AREA,#0,#QUSPEC ;открыть файл
         BCC     1$
         .PRINT   #NOFILE           ;файла нет
         .EXIT
1$:      MOV     #SIZE,R1
         CALL    CNV10
         .PRINT   #BUFF
         .EXIT
CNV10:   MOV     R0,—(SP)           ;подпрограмма
         CLR     R0                ;преобразования размера
1$:      INC     R0                ;файла для вывода на
         SUB     #10.,@ SP         ;терминал
         BGE    1$
         ADD     #72,@ SP
         DEC     R0
         BEQ    2$
         CALL    CNV10
2$:      MOVB   (SP)+,(R1)+
         RETURN
AREA:    .BLKW   3
QUSPEC:  .RAD50  /DK QUFILE/
         .RAD50  /TMP/
BUFF:    .ASCII  /размер файла QUFILE.TMP = /
SIZE:    .ASCIZ  /
NOFILE:  .ASCIZ  /на DK: нет файла QUFILE.TMP/
         .EVEN
         .END    START

```

Таблица 8

Значение аргумента	Действие монитора
1	2
-1	Производит поиск файла с текущего положения магнитной ленты
0	Перематывает магнитную ленту и производит поиск файла
N (N>0)	Устанавливает магнитную ленту в положение, соответствующее файлу с последовательным номером N

Вторая форма .LOOKUP связывает канал с указанным заданием для обмена информацией в режимах FB и XM.

Формат макрокоманды:

.LOOKUP AREA,CHAN,JOBDES

где AREA — адрес блока из 2-х слов аргументов EMT;

JOBDES — адрес блока из 4-х слов; 1-ое слово содержит имя

драйвера очереди сообщений (MQ) в кодах RADIX-50; 2, 3, 4 слова содержат логическое имя системного задания в кодах КОИ-7. Если 2-ое слово содержит 0, то канал (CHAN) может использоваться только запросами .READ/C/W для получения сообщения от любого задания.

Формат блока аргументов EMT:

```
AREA: .BYTE CHAN,1
      .WORD JOBDES
```

**Пример 2:**

```
.TITLE SJLOOK.MAC
```

```
; в данном примере запрос .LOOKUP открывает канал для
; системного задания QUEUE.REL.
```

```
; Пример выполняется под управлением мониторов FB или
; XM.
```

```
;
```

```
START: .MCALL .LOOKUP,PRINT,EXIT,.WRITW,.READW
        .LOOKUP #AREA,#0,#QMSG ;открыть канал для QUEUE
        BCC 1$
        .PRINT #NOJOB
        .EXIT
1$: .WRITW #AREA,#0,#RMSJ,#6
    BCS 2$
    .READW #AREA,#0,#RMSJ,#6
    BCS 2$
    .PRINT #QRUN
    .EXIT
2$: .PRINT #MSGERR
    .EXIT
AREA: .BLKW 5
QMSG: .RAD50 /MQ/
      .ASCIZ /QUEUE/
      .WORD 0,0
RMSJ: .WORD 0
      .ASCII /SJLOOK/
MSGERR: .ASCIZ /ошибка/
NOJOB: .ASCIZ /QUEUE не загружена/
QRUN: .ASCIZ /QUEUE работает!/
      .EVEN
      .END START
```

**5.1.6. Запрос .PURGE** освобождает указанный канал для использования в другой операции без выполнения каких-либо функций запросов .HRESET, .SRESET, .SAVSTATUS или .CLOSE. Временный файл, открытый на этом канале по .ENTER, аннулируется. Запрос .PURGE на свободном канале игнорируется.

Формат макрокоманды:

```
.PURGE CHAN
```

Формат блока аргументов EMT:

AREA: .BYTE CHAN,3

Пример использования .PURGE см. в описании запроса .HERR.

### 5.1.7. .RENAME

Запрос .RENAME изменяет имя файла (недопустим в операциях с магнитной лентой).

Формат макрокоманды:

.RENAME AREA,CHAN,DBLK

где AREA — адрес блока из 2-х слов аргументов EMT;

DBLK — адрес блока, содержащего имя устройства и имя файла, которые требуется изменить, и новое имя устройства и имя файла в кодах RADIX-50.

Формат блока аргументов EMT:

AREA: .BYTE CHAN,4  
.WORD DBLK

Например,

```
.RENAME #AREA,#7,#MXLK  
BCS RNMERR
```

```
.  
.  
DBLK: .RAD50 /MX1OLDFILMAC/  
.RAD50 /MX1NEWFILMAC/
```

Если файл с тем же именем, что и новое, уже существует на томе, то он удаляется. Повторное определение имени устройства MX1 в новой спецификации файла необходимо для правильного выполнения операции и не должно опускаться. После выполнения .RENAME используемый канал освобождается. Запрос .RENAME требует, чтобы драйвер соответствующего устройства был в памяти перед выполнением операции. Запрос .RENAME не выполняется для защищенных файлов.

### Пример.

```
.TITLE RENAME.MAC
```

, в данном примере запрос .RENAME используется для изменения имени файла.

```
.MCALL .CSISPC,.RENAME,.FETCH,.PRINT,.SRESET, EXIT  
ERRBYT = 52  
START: .CSISPC #FILESF,#DEFEXT ;ввод командной строки  
.FETCH #DRVLOD,#FILESF ;загрузить драйвер
```

```

BCS      2$
MOV      #FILESP,R2
MOV      #FILESP+46,R3
MOV      @R2,#FILESP+36
.REPT    4
MOV      (R2)+,(R3)+
.ENDR
.RENAME  #AREA,#0,#FILESP+36 ;переименовать файл
BCC      1$
DECB    @#ERRBYT           ;анализ ошибки
BEQ      3$
MOV      #ILLOP,R0
BR       4$
1$:      .SRESET
        .EXIT
2$:      MOV      #FERR,R0
        BR       4$
3$:      MOV      #NDFIL,R0
4$:      .PRINT
        BR       1$
AREA:    .BLKW    5
DEFEXT:  .WORD    0,0,0,0
NOFIL:   .ASCIZ   /файл не найден/
ILLOP:   .ASCIZ   /недопустимая операция/
FERR:    .ASCIZ   /ошибка .FETCH/
        .EVEN
FILESP:  .BLKW    39.*2
DRVL0D   =.
        .END     START

```

**5.1.8. Запрос .REOPEN** повторно связывает указанный канал с файлом, для которого был выполнен запрос **.SAVESTATUS**. Совместное использование запросов **.SAVESTATUS** и **.REOPEN** требуется для одновременного обращения к большому числу файлов. Файлы открываются по **.LOOKUP** и параметры их состояния запоминаются по **.SAVESTATUS**. Когда требуются данные из файла, **.REOPEN** обеспечивает программе возможность чтения этих данных. Не является обязательным выполнение **.REOPEN** на том же канале, на котором были выполнены **.LOOKUP** и **.SAVESTATUS**.

Формат макрокоманды:

```
.REOPEN AREA,CHAN,CBLK
```

где **AREA** — адрес блока из 2-х слов аргументов **EMT**;  
**CBLK** — адрес блока из 5-ти слов, где запоминается информация о состоянии канала.

Формат блока аргументов **EMT**:

```
AREA:  .BYTE  CHAN,6
       .WORD  CBLK
```

Пример см. в описании запроса **.SAVESTATUS**.

**5.1.9. Запрос .SAVESTATUS** запоминает пять слов информации о состоянии канала в указанной пользователем области памяти. Эти слова содержат полную информацию, которая требуется операционной системе для точного определения файла. После выполнения .SAVESTATUS указанный канал освобождается. Если необходимо получить данные по этому каналу, то используется запрос .REOPEN. Запрос .SAVESTATUS может использоваться в том случае, если файл был открыт по .LOOKUP. Если был использован .ENTER, то .SAVESTATUS игнорируется и генерируется ошибка. Запрос .SAVESTATUS недопустим в операциях с файлами, которые находятся на магнитной ленте.

Запросы .SAVESTATUS и .REOPEN используются для работы с большим числом файлов при ограниченном числе каналов.

Формат макрокоманды:

**.SAVESTATUS AREA,CHAN,CBLK**

где AREA — адрес блока из 2-х слов аргументов EMT;

CBLK — адрес блока из 5-ти слов, в который будет записана информация о состоянии канала; адрес должен быть указан в пределах программы пользователя. (см. ПРИЛОЖЕНИЕ 2).

Формат блока аргументов EMT:

AREA: .BYTE CHAN,5  
.WORD CBLK

**Пример:**

```

.TITLE      SAVEST.MAC
.MCALL     .CSIGEN, .SAVESTATUS, .READW, .WRITW, .PRINT
.MCALL     .PURGE, .REOPEN, .CLOSE, .EXIT
ERRBYT = 52
START:     .CSIGEN  #DSPACE,#DEFEXT ;ввод командной строки
           MOV      #3, R4           ;входной канал
           MOV      #AREA,R3        ;указатель на блок
                                           ;состояния канала
1$:        MOV      #SAVBLK,R5
           .SAVEST  R3,R4,R5        ;сохранить информацию
                                           ;о канале
           BCS      2$
           ADD      #12,R5          ;указатель на следующий
                                           ;блок информации
           INC      R4
           CMP      #8, R4          ;входные файлы все?
           BGE      1$              ;нет
2$:        MOV      #SAVBLK,R5
           BEQ      7$
4$:        .REOPEN  R3,#3,R5        ;повторно использовать
                                           ;канал 3
           CLR      BLK

```

5\$:	.READW	R3,#3,#BUFF,#256.	.BLK ;читать блок
	BCC	6\$	
	TSTB	Ⓒ #ERRBYT	;конец файла?
	BNE	8\$	
	.PURGE	#3	;освободить канал
	ADD	#12,R5	
	TST	Ⓒ R5	;последний файл?
	BNE	4\$	
	.CLOSE	#0	;закреть вых. канал
	.PRINT	#DONE	
	.EXIT		
6\$:	.WRITW	R3,#0,#BUFF,#256.	.WBLK ;записать блок
	INC	WBLK	
	INC	BLK	
	BCC	5\$	
	MOV	#WERR,R0	
	BR	9\$	
7\$:	MOV	#NOIMP,R0	
	BR	9\$	
8\$:	MOV	#RERR,R0	
9\$:	.PRINT		
	.EXIT		
AREA:	.BLKW	5	
BLK:	.WORD	0	
WBLK:	.WORD	0	
SAVBLK:	.BLKW	30.	
DEFEXT:	.WORD	0,0,0	
NOIMP:	.ASCIZ		/нет больше входных файлов/
WERR:	.ASCIZ		/ошибка записи/
RERR:	.ASCIZ		/ошибка чтения/
DONE:	.ASCIZ		/конец программы/
	.EVEN		
BUFF:	.BLKW	256.	
DSPACE	=.		
	.END	START	

## 5.2. Программные запросы для организации ввода-вывода данных

### 5.2.1. .READ, .READC, .READW

Система ФОДОС-2 предусматривает три режима выполнения операций ввода-вывода:

```
.READ /.WRITE
.READC /.WRITC
.READW /.WRITW
```

При использовании запросов .READ и .READC должны быть определены дополнительные элементы для очереди ввода-вывода.

Если запросы READ/C/W используются для обмена информацией между заданиями, то размер буфера (BUF) должен на одно слово превышать число слов для чтения (WCNT),

т. к. первое слово буфера (BUF) будет содержать действительное количество переданных слов.

Запрос .READ используется для передачи определенного количества слов по указанному каналу в буфер памяти. Управление передается программе пользователя немедленно после начала выполнения запроса .READ. После завершения передачи данных никаких действий не происходит.

Формат макрокоманды:

.READ AREA, CHAN, BUF, WCNT, BLK

где AREA — адрес блока из 5-ти слов аргументов EMT;

BUF — адрес буфера для чтения данных;

WCNT — количество слов для чтения;

BLK — номер блока для операции чтения; если был использован файловый .LOOKUP, то BLK — номер блока относительно начала файла; если был использован нефайловый .LOOKUP, то BLK — абсолютный номер блока устройства; программа пользователя должна модифицировать BLK перед его повторным использованием.

Формат блока аргументов EMT:

```
AREA:  .BYTE   CHAN,10
        .WORD   BLK
        .WORD   BUF
        .WORD   WCNT
        .WORD   1
```

**Пример:**

.TITLE READ.MAC

; пример использования запросов .READ, .WRITE.

```
        MCALL   .READ, .WRITE, .CLOSE, .PRINT, .CSIGEN
        MCALL   WAIT, .SRESET, .EXIT
ERRBYT = 52
ENABL   LSB
START   CSIGEN   #DSPACE, #DEFEXT ;ввод командной строки
        MOV     #AREA, R5
        CLR     IOBLK
1$      READ     R5, #3           ;читать с 0-го блока
        BCS     6$           ;ошибка чтения
        ;
        BIT     #1, IOBLK
        BNE     2$
        PRINT   #MES
        ;
2$      WAIT     #3           ;ждать завершения операции
        BCS     5$
        WRITE   R5, #0       ;записать блок
        BCS     3$
        INC     IOBLK
        ;
```

	.WAIT	#0	;ждать завершения операции
	BCC	1\$	
.3\$	MOV	#WERR,R0	
4\$	.PRINT		
	BR	7\$	
5\$:	MOV	#RERR,R0	
	BR	4\$	
6\$	TSTB	ⓐ #ERRBYT	
	BNE	5\$	
	.PRINT	#MES1	
	.CLOSE	#0	
7\$	.SRESET		;удалить загруженный драйвер
	.EXIT		
AREA:	.WORD	0	
IOBLK:	.WORD	0	
	.WORD	BUFF	
	.WORD	256.	
	.WORD	0	
BUFF:	.BLKW	256.	
DEFEXT:	.WORD	0,0,0	
MES1:	.ASCIZ	/операция ввода-вывода завершена/	
MES:	.ASCIZ	<15><12>/выполнение программы продолжается/	
WERR:	.ASCIZ	/ошибка .WRITE/	
RERR:	.ASCIZ	/ошибка .READ/	
	.EVEN		
DSPACE	=.		
	.END	START	

Если ввод выполняется с терминала и  $BLK=0$ , то на терминале печатается « $\wedge$ » (код 136).

Когда программе пользователя нужно выполнить чтение данных, может быть использован запрос .WAIT. Это гарантирует то, что данные будут прочитаны полностью. Если во время передачи данных произошла ошибка, запрос .WAIT сообщает о ней.

Запрос .READC передает определенное количество слов по указанному каналу в память. Управление программе пользователя будет передано сразу после начала выполнения .READC. Попытка чтения данных после конца файла вызывает передачу управления вызывающей программе с установлением C-разряда и записью нуля в байт ошибок (52-байт). После выполнения .READC управление будет передано подпрограмме завершения, адрес которой указан в запросе. Если в подпрограмме завершения выполняется команда RTS PC, то управление возвращается программе пользователя.

Формат макрокоманды:

```
.READC AREA,CHAN,BUF,WCNT,CRTN,BLK
где AREA — адрес блока из 5-ти слов аргументов EMT;
BUF — адрес буфера для чтения данных;
WCNT — количество слов для чтения;
```

CRTN — адрес подпрограммы завершения (адрес должен быть больше 500 (восьмеричное));

BLK — см. запрос .READ

Формат блока аргументов EMT:

```
AREA:  .BYTE   CHAN,10
        .WORD   BLK
        .WORD   BUF
        .WORD   WCNT
        .WORD   CRTN
```

При запуске подпрограммы завершения справедливо следующее:

1) регистр R0 содержит слово состояния канала; если в R0 установлен разряд 0, то произошла аппаратная ошибка во время передачи данных;

2) регистр R1 содержит номер канала для операции; эта информация необходима, когда одна и та же подпрограмма завершения используется в нескольких операциях.

Пример:

.TITLE READC.MAC

; пример использования запросов .READC, .WRITC.

```
.MCALL    .READC, .WRITC, .CSIGEN, .PRINT
.MCALL    .CLOSE, .WAIT, .SRESET, .EXIT
ERRBYT    = 52
.ENABL    LSB
START:    .CSIGEN  #DSPACE, #DEFEXT ;ввод командной строки
          CALL     IOXFER
          .PRINT   #MES
1$:       MOV      #—1, R5
          DEC      R5
          BNE     1$
          TSTB    EOF
          BEQ     1$ ;0 — ошибка записи
          BLT     RERR ;<0 — ошибка чтения
          .CLOSE  #0 ;>0 — конец файла
          MOV     #DONE, R0
          BR      EXIT
WERR:    MOV     #WRERR, R0
          BR      EXIT
RERR:    MOV     #RDERR, R0
EXIT:    .PRINT
          .SRESET ;удалить загруженные
          ;драйверы

          .EXIT
WRDONE:  .WAIT   #0
          BCS    3$
IOXFER:  .READC  #AREA, #3, , #4$ ;читать блок
          BCC   7$
          TSTB  @ #ERRBYT ;конец файла?
          BEQ   6$ ;да
2$:      DECB   EOF
```

```

3$:      DECB      EOF
         RETURN
4$:      .WAIT     #3
         BCS       2$
         .WRITC    #AREA,#0,,#WRDONE;записать блок
         BCS       3$
5$:      INC       BLOK
         RETURN
6$:      INCB      EOF
7$:      RETURN
AREA:    .WORD     0           ;блок аргументов EMT
BLOK:    .WORD     0           ;номер блока
         .WORD     BUFF       ;адрес буфера
         .WORD     256.       ;счетчик слов
         .WORD     0           ;адрес подпрограммы завершения
BUFF:    .BLKW     256.
DEFEXT:  .WORD     0,0,0,0
DONE:    .ASCIZ    /конец программы/
MES:     .ASCIZ    /выполнение программы продолжается/
WRERR:   .ASCIZ    /ошибка .WRITC/
RDERR:   .ASCIZ    /ошибка .READC/
EOF:     .BYTE     0
         .EVEN
DSPACE   =.
         .END      START

```

Запрос `.READW` передает определенное количество слов по указанному каналу в память. Управление передается программе пользователя после выполнения запроса `.READW` или при обнаружении ошибки.

Формат макрокоманды:

```
.READW AREA,CHAN,BUF,WCNT,BLK
```

где `AREA` — адрес блока из 5-ти слов аргументов `EMT`;  
`WCNT` — количество слов для чтения (каждый запрос может передать максимально 32К слов);  
`BLK` — см. запрос `.READ`

Формат блока аргументов `EMT`:

```

AREA:    .BYTE     CHAN,10
         .WORD     BLK
         .WORD     BUF
         .WORD     WCNT
         .WORD     0

```

Если после завершения запроса будет установлен `C`-разряд, то это указывает на то, что на канале произошла аппаратная ошибка. Если ошибки не было, то данные находятся в памяти, начиная с указанного адреса. В режиме `FB` одно из заданий может выполняться в то время как другое задание ожидает завершения операции.

**Пример:**

```
.TITLE READW.MAC
```

```

; пример использования запросов .READW, .WRITW.
      .MCALL   .CSIGEN, .READW, .WRITW, .PRINT, .CLOSE
      .MCALL   .SRESET, .EXIT
ERRBYT = 52
START: .CSIGEN  #DSPACE, #DEFEXT
      CLR      IOBLK
      MOV      #AREA, R5
READ:  .READW  R5, #3
      BCC     2$
      TSTB    @ #ERRBYT
      BEQ     3$
      MOV     #RERR, R0
1$:    .PRINT
      BR      4$
2$:    .WRITW  R5, #0
      INC     IOBLK
      BCC     READ
      MOV     #WERR, R0
      BR      1$
3$:    .CLOSE  #0
      .PRINT  #DONE
4$:    .SRESET
      .EXIT
DEFEXT: .WORD  0,0,0,0
AREA:   .WORD  0
IOBLK:  .WORD  0
        .WORD  BUFF
        .WORD  256.
        .WORD  0
BUFF:   .BLKW  256.
RERR:   .ASCIZ /ошибка .READW/
WERR:   .ASCIZ /ошибка .WRITW/
DONE:   .ASCIZ /конец программы/
        .EVEN
DSPACE =.
        .END      START

```

### 5.2.2. .WRITE, .WRITC, .WRITW

При использовании запросов .WRITE и .WRITC должны быть определены дополнительные элементы для очереди ввода-вывода.

Запрос .WRITE передает указанное количество слов из памяти по указанному каналу. Управление программе пользователя передается сразу же после того, как запрос будет поставлен в очередь.

Формат макрокоманды:

**WRITE** AREA, CHAN, BUF, WCNT, BLK

где AREA — адрес блока из 5-ти слов аргументов EMT;

BUF — адрес буфера памяти, который используется для вывода;

WCNT — количество слов для записи;

BLK — номер блока для операции записи; если используется

файловый .LOOKUP или .ENTER, то BLK — номер блока относительно начала файла; если был использован нефайловый .LOOKUP или .ENTER, то BLK — абсолютный номер блока устройства; программа пользователя должна модифицировать BLK перед его повторным использованием.

Формат блока аргументов EMT:

```
AREA: .BYTE CHAN,11
       .WORD BLK
       .WORD BUF
       .WORD WCNT
       .WORD 1
```

Запрос .WRITC передает заданное количество слов из памяти по указанному каналу. Управление передается программе пользователя немедленно после начала выполнения .WRITC. После выполнения запросов, управление будет передано подпрограмме завершения. По команде RTS PC управление возвращается программе пользователя.

Формат макрокоманды:

```
.WRITC AREA,CHAN,BUF,WCNT,CRTN,BLK
```

где AREA — адрес блока из 5-ти слов аргументов EMT;

BUF — адрес буфера памяти, который используется для вывода;

WCNT — количество слов для записи;

CRTN — адрес подпрограммы завершения;

BLK — см. запрос .WRITE

Формат блока аргументов EMT:

```
AREA: .BYTE CHAN,11
       .WORD BLK
       .WORD BUF
       .WORD WCNT
       .WORD CRTN
```

Запрос .WRITW передает заданное количество слов из памяти по указанному каналу. Управление передается программе пользователя после завершения .WRITW.

Формат макрокоманды:

```
.WRITW AREA,CHAN,BUF,WCNT,BLK
```

где AREA — адрес блока из 5-ти слов аргументов EMT;

BUF — адрес буфера, который используется для вывода;

WCNT — количество слов для записи;

BLK — см. запрос .WRITE.

Формат блока аргументов EMT:

```
AREA: .BYTE CHAN,10
       .WORD BLK
       .WORD BUF
```

```
.WORD   WCNT
.WORD   0
```

### 5.2.3. .RCVD, .RCVDC, .RCVDW (режим FB, XM)

Существуют три формы запроса получения данных, которые используются вместе с запросами .SDAT. Запросы .RCVD аналогичны запросам .READ, но в отличие от них осуществляют прием данных из другой программы, а не с устройства. При использовании запросов .RCVD и .RCVDC должны быть определены дополнительные элементы для очереди ввода-вывода.

Запрос .RCVD используется для получения данных или сообщений, посылаемых другой программой. Управление вызывающей программе передается сразу же после начала выполнения запроса .RCVD. При необходимости, с .RCVD используется запрос .MWAIT, который задерживает выполнение программы до окончания операции.

Формат макрокоманды:

```
.RCVD AREA, BUF, WCNT
```

где AREA — адрес блока из 5-ти слов аргументов EMT;

BUF — адрес буфера для сообщений (данных);

WCNT — количество передаваемых слов.

Формат блока аргументов EMT:

```
AREA:  .BYTE   0,26
        .WORD   <ЗАРЕЗЕРВИРОВАНО>
        .WORD   BUF
        .WORD   WCNT
        .WORD   1
```

После выполнения монитором запроса .RCVD первое слово буфера (BUF) будет содержать действительное количество переданных слов. Таким образом, область памяти для приема данных должна быть, по крайней мере, на одно слово больше размера передаваемого сообщения.

Комбинация .SDAT/.RCVD может быть использована для передачи как нескольких слов, так и целого буфера. Операция по запросу .RCVD заканчивается только после выполнения .SDAT из другой программы.

Пример использования .RCVD см. в описании запроса .MWAIT.

Запрос .RCVDC осуществляет прием данных и запускает подпрограмму завершения после получения сообщения (аналогично .READC).

Формат макрокоманды:

```
.RCVDC AREA, BUF, WCNT, CRTN
```

где CRTN — адрес подпрограммы завершения.

Формат блока аргументов EMT:

```
AREA: .BYTE 0,26  
      .WORD <ЗАРЕЗЕРВИРОВАНО>  
      .WORD BUF  
      .WORD WCNT  
      .WORD CRTN
```

Запрос .RCVDW осуществляет прием данных и ожидает окончания операции. Выполнение вызывающей программы задерживается до окончания передачи сообщения другой программой.

Формат макрокоманды:

```
.RCVDW AREA,BUF,WCNT
```

Формат блока аргументов EMT:

```
AREA: .BYTE 0,26  
      .WORD <ЗАРЕЗЕРВИРОВАНО>  
      .WORD BUF  
      .WORD WCNT  
      .WORD 0
```

Как и в .RCVD, первое слово буфера для .RCVDC и .RCVDW будет содержать действительное количество переданных слов.

Пример использования .RCVDW см. в описании запроса .CHCOPY.

**5.2.4. Запросы .SDAT, .SDATC, .SDATW** используются вместе с запросами .RCVD, .RCVDC, .RCVDW для передачи сообщений или данных между программами под управлением мониторов FB или XM.

Запросы .SDAT, .SDATC, .SDATW аналогично запросам .WRITE, .WRITC, .WRITW, только осуществляют передачу данных не на устройство, а другой программе. При использовании запросов .SDAT и .SDATC должны быть определены дополнительные элементы для очереди ввода-вывода.

.SDAT

Формат макрокоманды:

```
.SDAT AREA,BUF,WCNT
```

где AREA — адрес блока из 5-ти слов аргументов EMT;

BUF — адрес буфера для сообщений (данных), которые будут переданы;

WCNT — количество слов для передачи.

Формат блока аргументов EMT:

```
AREA: .BYTE 0,25  
      .WORD <ЗАРЕЗЕРВИРОВАНО>  
      .WORD BUF  
      .WORD WCNT  
      .WORD 1
```

Пример использования .SDAT см. в описании запроса .MWAIT.

.SDATC

Формат макрокоманды:

.SDATC AREA,BUF,WCNT,CRTN

где AREA — адрес блока из 5-ти слов аргументов EMT;

BUF — адрес буфера для сообщений (данных), которые будут переданы;

WCNT — количество слов для передачи;

CRTN — адрес подпрограммы завершения.

Формат блока аргументов EMT:

AREA: .BYTE 0,25  
.WORD <ЗАРЕЗЕРВИРОВАНО>  
.WORD BUF  
.WORD WCNT  
.WORD CRTN

.SDATW

Формат макрокоманды:

где AREA — адрес блока из 5-ти слов аргументов EMT;

BUF — адрес буфера для сообщений (данных), которые будут переданы;

WCNT — количество слов для передачи.

Формат блока аргументов EMT:

AREA: .BYTE 0,25  
.WORD <ЗАРЕЗЕРВИРОВАНО>  
.WORD BUF  
.WORD WCNT  
.WORD 0

Пример использования .SDATW см. в описании запроса .CHCOPY.

**5.2.5. Запросы .TTYIN и .TTINR** используются для передачи символов с терминала в программу пользователя.

Макрорасширение для .TTYIN:

EMT 340

BCS .—2

Макрорасширение для .TTINR:

EMT 340

Запрос .TTYIN ожидает подачу символа или строки символов, в то время как пользователь может или ожидать, или продолжать выполнение программы, используя .TTINR. Если по завершению выполнения .TTINR устанавливает C-разряд, то это значит, что в буфере терминала символ отсутствует (пользователь еще не напечатал его). В режимах FB, XM и SJ (без мультитерминальной поддержки) .TTINR не устанавливает

ливают С-разряд, если не установлен 6-ой разряд в слове состояния задания (ССЗ).

Существуют два режима ввода с терминала. Режим ввода определяется содержимым 12-го разряда в ССЗ. Если содержимое 12-го разряда ноль, то говорят об общем режиме ввода-вывода. В этом режиме выполняются следующие действия:

1) монитор обеспечивает «Эхо» всем символам, вводимым с терминала;

2) стирание символов и строк осуществляется с помощью команд ЗБ и СУ/У соответственно;

3) для передачи символов программе необходимо подать ВК, ПС или команды СУ/З или СУ/С; возврат каретки и перевод строки также передаются программе пользователя.

Если 12-ый разряд в ССЗ равен единице, то устанавливается специальный режим ввода-вывода. В этом режиме выполняются следующие действия:

1) монитор не выдает «Эхо» символам, вводимым с терминала, кроме СУ/С и СУ/Д;

2) СУ/У и ЗБ не выполняют своих функций;

3) символы сразу же передаются программе пользователя.

В специальном режиме программа пользователя должна сама выполнять печать вводимых символов, если это необходимо (исключение составляют СУ/С и СУ/Д).

12-ый разряд в ССЗ должен устанавливаться программой пользователя. Этот разряд очищается, если управление передается монитору.

Действия СУ/Ф, СУ/В и СУ/Х не зависят от содержимого 12-го разряда, если ранее не была подана команда SET TT NOFB.

Команды СУ/С и СУ/Д выполняются монитором, если только в режиме FB (ХМ) ранее не была выполнена команда SET TT NOPAGE.

Если в режиме FB (ХМ) сделан запрос ввода, а символа нет в наличии, то выполнение программы блокируется до тех пор, пока в буфере ввода не появится символ. Это справедливо для .TTYIN и .TTINR и не зависит от режима ввода. Для того, чтобы продолжить выполнение программы (и обеспечить установку С-разряда в случае отсутствия символа в буфере ввода), необходимо установить перед .TTINR 6-ой разряд в ССЗ. 6-ой разряд очищается после окончания выполнения программы.

Вводимые по .TTYIN, .TTYINR символы можно редактировать с помощью SL, если предварительно были поданы

команды SET SL ON и SET SL TTYIN и содержимое 4-го и 12-го разрядов ССЗ равны нулю.

При редактировании посредством SL содержимое 6-го разряда ССЗ не существенно и выполнение запроса .TTINR будет завершено только после завершения операции редактирования.

**ПРИМЕЧАНИЕ.** Запрос .TTYIN не обслуживает косвенные файлы. Для этой цели необходимо воспользоваться запросом .GTLIN. По .TTYIN невозможно получить символ из косвенного файла.

Формат макрокоманды:

.TTYIN CHAR

.TTINR

где CHAR — адрес ячейки, куда пересылается код вводимого символа; если CHAR указан в запросе, то код пересылается в указанную ячейку и в R0; если CHAR не указан, то в R0.

Пример использования .TTYIN, .TTINR см. в описании запросов .TTYOUT, .TTOUTR.

**5.2.6. Запросы .TTYOUT, .TTOUTR** используются для передачи символов из R0 в буфер терминала. Действия этих запросов аналогичны действиям запросов .TTYIN, .TTINR. Если буфер терминала заполнен, то запрос .TTYOUT ожидает появления в буфере свободного места для выводимого символа, тогда как .TTOUTR не ожидает свободного места и символ, находящийся в R0, на терминал не выводится. Если после завершения .TTOUTR установлен С-разряд, то это говорит о том, что в буфере нет свободного места и что символ не был выведен на терминал. В режимах FB, XM запрос .TTOUTR не устанавливает С-разряд. В этом случае выполнение программы блокируется до появления в буфере свободного места. Для того, чтобы продолжить выполнение программы и обеспечить установку С-разряда, необходимо в ССЗ установить 6-ой разряд.

**ПРИМЕЧАНИЕ.** Если в основном задании установлен 6-ой разряд в ССЗ, то запросы .TTYIN и .TTYOUT будут вызывать блокировку фонового задания. Заметим, что каждое задание в основном или фоновом режиме имеет свое собственное ССЗ и поэтому они могут использовать различные режимы работы терминала независимо друг от друга.

Формат макрокоманды:

.TTYOUT CHAR

.TTOUTR

где CHAR — адрес ячейки, содержащей код символа, который будет пересылаться в R0 с последующим выводом на термини-

нал; если CHAR не указан, на терминал выводится содержимое R0.

**Пример:**

```

        .TITLE      TTIO.MAC
        .MCALL      .TTYIN,.TTYOUT,.TTINR,.TTOUTR,.EXIT
START:   JSW        =44 ;слово состояния задания
        MOV        #BUFFER,R1
        CLR        R2
INLOOP:  .TTINR
        BCS        NOCHR
CHRIN:   MOV        R0,(R1) + ;переслать символ в буфер
        INC        R2
        CMPB       R0,#12
        BNE        INLOOP
        MOV        #BUFFER,R1
OUTLOOP: MOV        (R1),R0 ;переслать символ в R0
        .TTOUTR    ;вывести символ на TT:
        BCS        NOROOM
CHRROUT: DEC        R2
        BEQ        START
        INC        R1
        BR         OUTLOOP ;печатать следующий символ
NOCHR:   .TTINR
        BCC        CHRIN ;есть символ?
        ;
        ;
        ;
        BR         NOCHR
NOROOM: MOV        (R1),R0
        BCC        CHRROUT ;при наличии символа —
        ; ;печатать его
        ;
        ;
        ;
        BIC        #100,@ #JSW ,очистить 6 разряд и
        .TTYOUT    (R1) ;печатать символ
        BIS        #100,@ #JSW
        BR         CHRROUT
BUFFER:  .BLKB     128.
        .END      START

```

**5.2.7. Запрос .PRINT** вызывает печать строки символов КОИ-7 на системном терминале. В режимах FB, XM каждому сообщению предшествует побуждающий знак B> или F>, в зависимости от того, какое задание его печатает (B> — фоновое, F> — основное). Это происходит в случае переключения системного терминала на обслуживание другого задания.

Если ввод на терминал выполняется системным заданием, то в качестве побуждающего знака печатается имя системного задания.

Строка текста, которая выводится на печать, может оканчиваться как нулевым (0) байтом, так и байтом 200. В первом случае, в отличие от второго, вывод автоматически заканчивается <BK> и <PC>.

Управление возвращается программе пользователя после того как все символы будут записаны в буфер вывода монитора.

Основное задание печатает сообщение сразу же, независимо от состояния фонового задания.

Таким образом, для печати срочных сообщений должен использоваться запрос .PRINT, а не .TTYIN или .TTYOUT.

Если фоновое задание печатает побуждающий знак и звездочку, то системный терминал будет принадлежать ему и запросы .TTYOUT из основного задания выполняться не будут, пока для фонового не будет выполнен <BK>.

Формат макрокоманды:

.PRINT ADDR

где ADDR — адрес строки, которую необходимо напечатать.

**Пример:** .TITLE PRINT.MAC

; в этом примере запрос .PRINT используется для печати символов, вводимых с терминала.

```
.MCALL .GTLIN, .PRINT, .EXIT
START: .GTLIN  #BUFF, #PROMPT      ; ввод символов с
                                           ; терминала
      TSTB      BUFF                ; строка пустая?
      BEQ       1$                  ; да
      .PRINT    #BUFF                ; нет — печать
                                           ; строки
      CLRB      BUFF
      BR        START
1$: .EXIT
BUFF: .BLKW     41.
PROMPT: .ASCII  /введите строку:  /<200>
      .END START
```

**5.2.8. Запрос .MTIN** является мультитерминальной формой запроса .TTYIN. Запрос .MTIN пересылает один или более символов из буфера ввода монитора в буфер, указанный пользователем. Терминал должен быть логически связан с текущим заданием. Если запрос выполнен успешно, то модифицированный адрес буфера пользователя будет записан в RO. Если разряд TCBIT установлен в слове M. TSTS, то запрос .MTIN устанавливает C-разряд и передает управление выполняющей программе, если ввод с терминала невозможен (т.е. нет строки, если разряд TTSPC очищен или нет симво-

ла в буфере ввода, если разряд TTSPC установлен). В противном случае, .MTIN будет ожидать операции ввода и, тем самым, выполнение задания будет приостановлено.

Формат макрокоманды:

.MTIN AREA,ADDR,UNIT[,CHRCNT]

где AREA — адрес блока из 3 слов аргументов EMT;

ADDR — адрес буфера пользователя;

UNIT — логический номер терминала (ЛНТ);

CHRCNT — количество пересылаемых символов (от 1 до 255);

если этот аргумент опущен, то передается один символ.

Формат блока аргументов EMT:

AREA: .BYTE 2,37  
.WORD ADDR  
.BYTE UNIT,CHRCNT

**5.2.9. Запрос .MTOUT** является мультитерминальной формой запроса .TTYOUT. Запрос .MTOUT пересылает один или более символов из буфера пользователя в буфер вывода терминала. Терминал должен быть логически связан с заданием. Модифицированный адрес буфера пользователя будет записан в R0, если запрос выполнен успешно. Если в буфере вывода нет свободного места и разряд TCBIT установлен в слове M.TSTS, то .MTOUT устанавливает C-разряд и передает управление вызывающей программе. Если разряд TCBIT очищен, выполнение задания приостанавливается до тех пор, пока не появится свободное место в буфере вывода.

Формат макрокоманды:

.MTOUT AREA,ADDR,UNIT[,CHRCNT]

где AREA — адрес блока из 3 слов аргументов EMT;

ADDR — адрес буфера пользователя;

UNIT — логический номер терминала;

CHRCNT — количество передаваемых символов (от 1 до 255);

если этот аргумент опущен, то передается один символ.

Формат блока аргументов EMT:

AREA: .BYTE 3,37  
.WORD ADDR  
.BYTE UNIT,CHRCNT

**5.2.10. Запрос .MTPRNT** является мультитерминальной формой запроса .PRINT. Он осуществляет вывод одной или нескольких строк символов на указанный терминал.

Формат макрокоманды:

.MTPRNT AREA,ADDR,UNIT

где AREA — адрес блока из 3 слов аргументов EMT;

ADDR — адрес строки, которую необходимо напечатать (т. е. .ASCII /строка/ <200> или .ASCIZ /строка/);

UNIT — логический номер терминала.

Формат блока аргументов EMT:

```
AREA:  .BYTE  7,37
        .WORD  ADDR
        .BYTE  UNIT
```

5.2.11. Запрос .SPFUN используется главным образом драйверами устройств для выполнения специальных функций, зависящих от этих устройств.

Формат макрокоманды:

```
.SPFUN AREA,CHAN,FUNC,BUF,WCNT,BLK[,CRTN]
```

где AREA — адрес блока из 6 слов аргументов EMT;

FUNC — код функции, которая будет выполняться;

BUF — адрес буфера; этот аргумент должен быть равен нулю, если для операции буфер не требуется;

CRTN — адрес подпрограммы завершения; если CRTN не указан, то его значение автоматически приравнивается нулю; если CRTN равен нулю, то управление программе пользователя будет передано после завершения операции; если CRTN равен 1, управление будет передано программе пользователя немедленно после начала операции; если CRTN=N, то N — адрес подпрограммы завершения ( $N > 500$  (восьмеричной)).

Аргументы CHAN, BLK, WCNT те же, что и в запросах .READ/.WRITE. Они используются только тогда, когда необходимо произвести запись на магнитную ленту с расширенным межфайловым промежутком.

Формат блока аргументов EMT:

```
AREA:  .BYTE  CHAN,32
        .WORD  BLK
        .WORD  BUF
        .WORD  WCNT
        .BYTE  377,FUNC
        .WORD  CRTN
```

В табл. 9 приводятся функции запроса .SPFUN и соответствующие им коды.

Таблица 9

Функция запроса	Код			
	MT	MX	MY	LD
1	2			
Перемотка до точки загрузки	373	—	—	—
Запись метки EOF	377	—	—	—
Перемотка вперед на один блок	376	—	—	—
Перемотка назад на один блок	375	—	—	—
Запись	371	376	376	—

1	2			
Чтение	370	377	377	—
Запись с расширенным межфайловым промежутком	374	—	—	—
Обратная перемотка в автономном режиме	372	—	—	—
Запись со стиранием маркера данных	—	375	375	—
Определение размера устройства (тома)	—	—	373	373
Чтение (запись) таблицы трансляции	—	—	—	372

Для использования запроса .SPFUN необходимо, чтобы драйвер соответствующего устройства находился в памяти и канал был связан с файлом по запросу .LOOKUP.

**Пример:**

.TITLE SPFUN.MAC

; пример использования запроса .SPFUN.

```

START: .MCALL      .SPFUN, .FETCH, .LOOKUP, .PRINT, .CLOSE, .EXIT
       .FETCH     #DSPC, #MT           ;загрузить драйвер MT:
       BCS        1$
       .LOOKUP    #AREA, #4, #MT      ;открыть канал
       BCS        2$
       .SPFUN     #AREA, #4, #373, #0 ;перемотка до точки загрузки
       .BCS       3$
       .CLOSE     #4
       .EXIT
1$:    MOV        #FERR, R0
       BR         5$
2$:    MOV        #LKERR, R0
       BR         5$
3$:    MOV        #SPERR, R0
5$:    .PRINT
       .EXIT
AREA:  .WORD      0
MT:    .RAD50     /MT/
       .WORD      0, 0, 0
FERR:  .ASCIZ     /ошибка .FETCH/
LKERR: .ASCIZ     /ошибка .LOOKUP/
SPERR: .ASCIZ     /ошибка .SPFUN/
       .EVEN
DSPC   =.
       .END      START

```

### 5.3. Запросы общего назначения

5.3.1. Запрос .CDFN используется для переопределения числа каналов ввода-вывода. Каждое задание независимо от того является ли оно основным или фоновым, первоначально обеспечивается 16 (десятичное) каналами ввода-вывода (от 0 до 15). Запрос .CDFN позволяет увеличить число каналов до

255 (десятичное) (от 0 до 254). Канал с номером 255 используется монитором. Область памяти для новых каналов определяется в пределах программы пользователя. Под каждый канал ввода-вывода отводится пять слов (семь слов для монитора ХМ). В режиме ХМ область памяти для дополнительных каналов, указанных в запросе .CDFN, должна располагаться в нижних 28К оперативной памяти. Рекомендуется, чтобы запрос .CDFN выполнялся в начале программы перед операциями ввода-вывода. Если в программе используется более одного запроса .CDFN, то области каналов для них не должны перекрывать друг друга, но могут иметь один и тот же начальный адрес. Два запроса .SRESET и .HRESET восстанавливают первоначальное число каналов (16 (десятичное)). Следовательно все запросы .CDFN должны выполняться повторно после использования .SRESET или .HRESET.

Заметим, что запрос .CDFN определяет новые каналы. Область памяти для ранее определенных каналов не используется. Так, запрос .CDFN для 20 (десятичное) каналов ввода-вывода (в то время как 16 первоначальных каналов уже определены) создает 20 новых каналов ввода-вывода. Область для 16-ти первоначальных каналов не используется, но содержимое старых каналов будет скопировано в новые.

Для программы с оверлейной структурой 15-й канал используется оверлейным драйвером и не должен изменяться.

Формат макрокоманды:

CDFN AREA,ADDR,NUM

где AREA — адрес блока из 3 слов аргументов EMT;

ADDR — адрес области памяти для новых каналов;

NUM — число создаваемых каналов.

Формат блока аргументов EMT:

```
AREA:   .BYTE    0,15
        .WORD    ADDR
        .WORD    NUM
```

**Пример:**

TITLE CDFN.MAC

В этом примере запрос .CDFN используется для определения 32 новых каналов.

```
START:  .MCALL   .CDFN, .PRINT, .EXIT, .HRESET
        .CDFN   #AREA, #CHAN, #255. ;определить 32.канала
        BCC     1$
        PRINT   #MSG1                ;печать сообщения
        .EXIT
        .PRINT  #MSG2                ;выход из программы
        .HRESET
        .EXIT
        .BLKW   3                    ;печать сообщения
        ;выход из программы
        ; блок аргументов EMT
```

```

CHAN:  BLKW      5*255.                ;область для каналов
MSG1:  .ASCIZ   /ошибка .CDFN/
MSG2:  .ASCIZ   /определены 32. новых канала/<15> <12>
        .ASCIZ   /возврат к 16. каналам/
        .EVEN
        .END      START

```

**5.3.2. Запрос .CHAIN** позволяет одной фоновой программе инициировать выполнение другой фоновой программы без вмешательства оператора. Этот процесс может повторяться, и позволяет организовать последовательное выполнение большого числа программ.

Ячейки 500—507 должны содержать имя устройства и имя файла (в кодах RADIX-50), который необходимо вызвать. Ячейки 510—777 используются для передачи информации между последовательно выполняемыми программами.

Формат макрокоманды: .CHAIN

#### ПРИМЕЧАНИЯ.

1) Содержимое областей оперативной памяти, за исключением области памяти, которую занимает резидентный монитор и ячеек 500—777, может изменяться после выполнения .CHAIN; если .CHAIN выполняется из виртуального задания, то содержимое ячеек 500—777 не сохраняется.

2) Каналы ввода-вывода остаются открытыми после выполнения запроса .CHAIN для использования новой программой. Однако, каналы ввода-вывода, открытые по запросу .CDFN не могут использоваться. Так как по запросу .CHAIN монитор восстанавливает первоначальное число каналов (16 (десятичное)), то программа, которая оставляет файлы открытыми при выполнении .CHAIN не должна использовать .CDFN. Нерезидентные драйверы устройств удаляются из памяти при выполнении .CHAIN, поэтому они должны быть вновь загружены новой программой.

3) По 8-ому разряду ССЗ программа может определить, как она была загружена — по запросу .CHAIN или по команде RUN. Разряд устанавливается во время выполнения программы, если она была загружена с помощью .CHAIN. Если программа, занимающая ячейки от 500 до 777, предназначена для загрузки по запросу .CHAIN, то в ней должен быть установлен 8-ой разряд в ССЗ во время ее трансляции. В этом случае программа будет загружена правильно. Если разряд не установлен, ячейки 500—777, содержимое которых сохраняется от предыдущей программы, могут вызвать неправильную работу новой программы.

4) Для запроса .CHAIN характерны те же ошибки, что и для команды RUN.

### Пример:

; в этом примере программа CHAIN1.SAV загружается  
; из программы CHAIN.SAV посредством запроса .CHAIN

```
.TITLE      CHAIN.MAC
.MCALL     .CHAIN, .TTYIN, .PRINT
START:     MOV      #500,R1
           MOV      #CHPTR,R2
           .REPT    4
           MOV      (R2)+,(R1)+      ;имя файла в ячейки 500—507
           .ENDR
           .PRINT   #PROMT
LOOP:      .TTYIN                                ;передать символы с ТТ: в
                                                ;ячейки 510—777
           MOVB    R0,(R1)+
           CMPB    R0,#12                ;конец строки?
           BNE     LOOP                  ;нет
           CLRB   @R1
           .CHAIN  ;запуск программы CHAIN1
CHPTR:    .RAD50  /DK/
           .RAD50  /CHAIN1/
           .RAD50  /SAV/
PROMT:    .ASCII  /введите данные для программы CHAIN1: /<200>-
           .END    START
; программа
.TITLE    CHAIN1.MAC
.MCALL    .PRINT, .EXIT
JSW = 44
CHAIN$ = 400
STARTC   BIT    #CHAIN$,@ #JSW ;8 разряд установлен?
           BEQ    1$ ;нет
           .PRINT #CHAIND      ;да — печать сообщения
           MOV    #510,R0
           .PRINT                                ;печать содержимого ячеек
                                                ;510—777
           .EXIT
1$       .PRINT    #NOCHN
           .EXIT
CHAIND   .ASCII  /содержимое ячеек 510—777 : /<200>
NOCHN    .ASCIZ  /не установлен 8-ой разряд !/
           .END    STARTC
```

**5.3.3. Запрос .СМКТ** аннулирует один или несколько запросов .MRКТ.

Формат макрокоманды:

**СМКТ** AREA, ID [, TIME]

где AREA — адрес блока из 3 слов аргументов ЕМТ;  
ID — число, используемое для идентификации с запросом  
MRКТ, который необходимо аннулировать; если несколько  
запросов .MRКТ имеют одинаковый ID, то аннулируется тот  
из них, которому соответствует наименьший неистекший интервал  
времени; если ID=0, то аннулируются все запросы  
MRКТ;

TIME — адрес блока из 2 слов, в который монитор записывает величину неиспользованного времени в аннулированном запросе; первое слово содержит нижнюю границу интервала времени, второе — верхнюю границу; если адрес не указан, такой записи не происходит, если ID=0, аргумент TIME игнорируется.

Формат блока аргументов EMT:

```
AREA: .BYTE 0,23  
      .WORD ID  
      .WORD TIME
```

ПРИМЕЧАНИЕ. Аннулированный запрос .MRKT освобождает соответствующий элемент очереди.

Пример использования .CMKT см. в описании запроса .MRKT.

#### 5.3.4. .CNTXSW (режим FB, XM)

При передаче управления от одного задания другому, монитор выполняет так называемую операцию переключения контекста. Она заключается в сохранении системой необходимых параметров для однозначного определения и выполнения задания. Эти параметры включают все регистры общего назначения, ячейки 34—52, а также все регистры ППЗ, если запрос .SFPA выполнялся с отличным от нуля адресом подпрограммы обработки прерывания от ППЗ.

Запрос .CNTXSW используется для определения ячеек памяти, которые будут дополнительно включены в операцию переключения контекста.

Иногда для обоих заданий необходимо независимое использование определенной ячейки памяти, которая не включена в обычную операцию переключения контекста. Например, если программа использует команду IOT, то она должна установить в векторе 20 адрес своей подпрограммы обработки прерывания по этой команде. А если оба задания — основное и фоновое — используют команду IOT, то вектор 20 должен всегда содержать соответствующий с выполняемым заданием адрес. Это достигается путем включения ячеек 20 и 22 в список адресов запроса .CNTXSW.

Если запрос .CNTXSW указан в программе несколько раз, то монитор использует список адресов последнего из них.

Список адресов .CNTXSW не может быть в области свопинга USR и его нельзя изменять в процессе работы программы.

В режиме XM запрос .CNTXSW игнорируется для виртуальных заданий, т. к. они не используют память совместно с другими заданиями. Для виртуальных заданий векторы IOT,

TRAP, BPT и вектор 250 автоматически включаются монитором XM в операцию переключения контекста, и, следовательно, могут быть использованы другим заданием без .CNTXSW.

Формат макрокоманды:

.CNTXSW AREA,ADDR

где AREA — адрес блока из 2 слов аргументов EMT;  
ADDR — адрес списка адресов (должен быть ограничен нулем); адреса в списке должны быть четными и в пределах от 2 до 476 или в области задания пользователя, или в пределах от 160000 или 177776.

Формат блока аргументов EMT:

AREA: .BYTE 0,33  
.WORD ADDR

Пример:

.TITLE CNTXSW.MAC

; в этом примере запрос .CNTXSW используется для включения ячеек 20, 22, 177302, 177304, 177310 в операцию переключения контекста.

```
.MCALL .CNTXSW, .PRINT, .EXIT
START: .CNTXSW#AREA,#SWLIST
        BCC 1$
        .PRINT #ADDRER
        .EXIT
1$:     .PRINT #CNTOK
        .EXIT
SWLIST: .WORD 20
        .WORD 22
        .WORD 177302
        .WORD 177304
        .WORD 177310
        .WORD 0
AREA:   .BLKW 2
ADDRER: .ASCIZ /ошибка .CNTXSW/
CNTOK:  .ASCIZ /адреса включены в операцию контекста/
        .END START
```

5.3.5. Запрос .CSIGEN вызывает работу интерпретатора командной строки (CSI) в общем режиме. В этом режиме CSI осуществляет синтаксический анализ командной строки и выполняет следующие функции:

- 1) загружает драйверы указанных устройств;
- 2) выполняет запросы .LOOKUP и/или .ENTER для указанных файлов;
- 3) записывает в стек информацию о переключателях;

4) выполняет запрос .CLOSE на каналах 0—10 (восьмеричное).

Область, которая отводится для драйверов устройств, должна быть достаточной для одновременного размещения всех необходимых драйверов. Если этой области недостаточно, то программа пользователя может быть разрушена.

После выполнения запроса .CSIGEN регистр R0 будет указывать на первую свободную ячейку памяти после драйверов устройств.

Когда управление возвращается программе пользователя после выполнения запроса .CSIGEN, все указанные файлы будут открыты для ввода и/или вывода. Три выходных файла будут открыты соответственно на 0, 1 и 2-ом каналах; шесть входных — будут связаны с каналами 3—10 (восьмеричное).

Например, в командной строке

\* ,LP: =F1,F2

0-ой и 2-ой каналы остаются свободными, 1-й канал связывается с LP: , 3-й и 4-й каналы связываются соответственно с файлами F1 и F2 на DK: , каналы 5—10 (восьмеричное) остаются свободными.

Программа пользователя может определить открыт или нет канал посредством запроса .WAIT на соответствующем канале и получить сообщение об ошибке, если канал не открыт.

Переключатели и их аргументы записываются в стек.

Формат макрокоманды:

.CSIGEN DEVSPC,DEFTUP,CSTR[LINBUF]

где DEVSPC — адрес области памяти для драйверов устройств;

DEFTUP — адрес блока из 4 слов, который содержит типы файлов в кодах RADIX-50. Содержимое этого блока используется монитором, если в командной строке не указаны типы файлов;

CSTR — адрес командной строки. Она не должна содержать BK, PC, но должна оканчиваться нулевым байтом. Если CSTR не указан или 0, то ввод осуществляется с терминала; под управлением монитора FB (XM), если ввод осуществляется с системного терминала, будет автоматически выполнен запрос .UNLOCK для USR;

LINBUF — адрес, по которому будет записана копия введенной командной строки; эта область определяется пользователем и имеет длину 81 (десятичное) байт; командная строка в этой области ограничивается нулевым байтом.

## ПРИМЕЧАНИЯ.

1) Блок, адрес которого указан в DEFTUP, в 1-ом слове содержит тип файла для всех входных файлов, во 2, 3 и 4-ом словах — типы файлов соответственно для 0, 1 и 2-го выходных файлов. Если в блоке не указан тип файла для определенного канала, соответствующее слово должно содержать ноль. Все типы файлов должны быть записаны в кодах RADIX-50.

2) Запрос CSIGEN автоматически воспринимает командную строку из косвенного файла, если указан ввод с терминала (CSTR=0) и программа, использующая .CSIGEN, вызывается через косвенный файл.

3) Аргумент LINBUF предоставляет пользователю возможность восстановления первоначально введенной командной строки.

4) Если ошибка происходит при вводе командной строки с терминала, то CSI печатает сообщение об ошибке на терминале, если при вводе из памяти — устанавливается C-ряд и код ошибки (52-ой байт).

### Пример:

```
.TITLE CSIGEN.MAC
; пример программы с использованием запроса .CSIGEN
; выполняет копирование указанного файла
.MCALL .CSIGEN, .READW, .WRITW, .CLOSE, .SRESET, .EXIT
.MCALL .PRINT
.ERRBYT =52
START: .CSIGEN #DSPACE, #DEXT ;ввести командную строку
       MOV     R0, BUFF      ;указатель на свободную
                               ;память
                               ;номер блока
       CLR     INBLK
       MOV     #AREA, R5
READ:  .READW  R5, #3, BUFF, #400, INBLK ;читать блок
       BCC    2$
       TSTB   @ #ERRBYT      ;конец файла?
       BEQ    EOF            ;да
       MOV     #INERR, R0
1$     .PRINT                    ;печать сообщения
       CLR     R0
       .EXIT
2$     .WRITW  R5, #0, BUFF, #400, INBLK ;записать блок
       BCC    NOERR
       MOV     #WTERR, R0
       BR     1$
NOERR: INC     INBLK          ;увеличить счетчик и
       BR     READ           ;читать следующий блок
101    .CLOSE  #0            ;конец файла — закрыть
       .CLOSE  #3            ;входной и выходной
                               ;каналы
       .SRESET                ;удалить драйверы из
                               ;памяти
```

```

.EXIT
DEXT:   WORD      0,0,0,0
BUFF:   WORD      0
INBLK:  .WORD     0
AREA:   .BLKW     5
INERR:  .ASCIZ    /ошибка ввода/
WTERR:  .ASCIZ    /ошибка вывода/
        .EVEN
DSPACE  =
        .END      START

```

**5.3.6. Запрос .CSISPC** вызывает работу CSI в специальном режиме. В этом режиме он осуществляет синтаксический анализ командной строки и передает спецификации файлов и переключатели, указанные в программе пользователя. В специальном режиме CSI не выполняет действий, аналогичных функциям запросов .FETCH, .CLOSE, .ENTER, .LOOKUP.

Переключатели и их аргументы пересылаются в стек.

Если программа, использующая .CSISPC, вызывается через косвенный файл и CSTR=0, то .CSISPC автоматически выбирает входную строку из косвенного файла.

Формат макрокоманды:

```
.CSISPC OUTSPC,DEFTUP[,CSTR][,LINBUF]
```

где OUTSPC — адрес блока из 39 слов для спецификаций файлов командной строки;

DEFTUP — адрес блока из 4 слов, который содержит типы файлов в кодах RADIX-50. Содержимое этого блока используется монитором, если в командной строке не указаны типы файлов;

CSTR — адрес командной строки. Она не должна содержать BK, PC, но должна оканчиваться нулевым байтом. Если CSTR не указан или 0, то ввод осуществляется с терминала; LINBUF — адрес, по которому будет записана копия введенной командной строки; эта область определяется пользователем и имеет длину 81 байт; командная строка в этой области ограничивается нулевым байтом.

Блок, адрес которого указан в OUTSPC, содержит спецификацию для 9 файлов: по 5 слов для каждого из 3 выходных файлов и по 4 слова для каждого из 6 входных файлов. Пять слов для выходного файла имеют следующее содержимое: 1-ое слово — имя устройства; 2-ое и 3-е слова — имя файла; 4-ое слово — тип файла; 5-е слово — длина выходного файла. Если какой-то файл отсутствует, то соответствующие слова блока содержат нули.

Например,

```
*DX1:LIST.MAC[15]=PC:
```

после выполнения .CSISPC первые 5 слов для блока, адрес

которого указан в OUTSPC, будут иметь следующие значения:

16337 RADIX-50 для DX1  
46173 RADIX-50 для LIS  
76400 DARIX-50 для T  
50553 RADIX-50 для MAC  
00017 длина файла

Первые 4 слова для спецификаций входных файлов (начиная с адреса OUTSPC+36) будут выглядеть так:

62170 RADIX-50 для PC  
0 имя файла не указано  
0  
0

Если при вводе с терминала произошла ошибка, USR не печатает на терминале сообщение об ошибке, а устанавливает С-разряд в ССП.

**Пример:**

**.TITLE CSISPC.MAC**

; в этом примере используется специальный режим работы  
; интерпретатора командной строки.  
; Запрос .DSTATUS определяет загружен ли драйвер выход-  
; ного устройства и, если нет, то используется запрос .FETCH  
; для загрузки драйвера в память.  
; Затем указанный файл удаляется по запросу .DELETE.

```
START:      .MCALL .CSISPC, .DSTATUS, .FETCH, .PRINT, .EXIT, .DELETE
            .MOV   SP, R5      ;сохранить текущий указатель стека
            .CSISPC #OUTSP, #DEFEXT ;ввод командной строки
            .MOV   R5, SP      ;восстановить указатель стека
            .DSTAT #STAT, #OUTSP ;проверить наличие драйвера
                                ;выходного устройства
            .TST   STAT+4      ;драйвер загружен?
            .BNE   2$          ;да
            .FETCH #DEVLOD, #INSPEC;нет—загрузить драйвер
            .BCC   2$
            .PRINT #FEFAIL
            .EXIT
2$         .DELETE #AREA, #0, #INSPEC;удалить файл
            .BCC   3$
            .PRINT #NOFILE
            .BR    START
3$         .PRINT #FILDEL
            .EXIT
AREA      .BLKW 2
STAT      .BLKW 4
DEFEXT    .WORD 0,0,0,0
FEFAIL:   .ASCIZ /ошибка .FETCH/
NOFILE    .ASCIZ /файл не найден/
FILDEL    .ASCIZ /файл удален/
EVEN
```

```

OUTSP:   .BLKW  5*3      ;спецификации вых. файлов
INSPEC:  .BLKW  4*6      ;спецификации вх. файлов
DEVLOD:  .BLKW  1        ;адрес загрузки драйвера
          .END    START

```

### 5.3.6.1. Передача информации о переключателях в программе пользователя

В общем и специальном режимах работы CSI переключатели и их аргументы пересылаются в стек. Переключатель представляет собой наклонную черту (/), за которой следует имя переключателя (символ КОИ-7). За переключателем может следовать аргумент, перед которым ставится разделитель (:). Аргумент может быть восьмеричным или десятичным числом (числом с точкой), или комбинацией буквенно-цифровых символов (от одного до трех), первый из которых — буква. В командной строке переключатели могут быть связаны с файлами.

Например: \*DK:FOO/A,MX1:FILE.OBJ/A:100

В этом случае имеется два переключателя A, первый из которых связан с входным файлом FOO, второй связан с входным файлом FILE:OBJ и имеет аргумент 100 (восьмеричное).

В табл. 10 приводится формат содержимого стека после выполнения запроса .CSIGEN или .CSISPC. Указатель стека будет содержать адрес первого слова.

Таблица 10

Слово	Содержимое	Значение
1	2	3
1	N	Число переключателей в командной строке
2	Имя переключателя и номер файла	Четный байт содержит имя переключателя. Разряды 8—14 содержат номер (от 0 до 10) файла, с которым связан данный переключатель. Разряд 15 содержит 1, если переключатель имеет аргумент
3	Аргумент переключателя или имя следующего переключателя	Если установлен 15-й разряд в слове 2, то слово 3 содержит аргумент переключателя. В противном случае — имя следующего переключателя

Например, после обработки CSI командной строки

```
*FILE/B:20.,FIL2/E=MX1:INPUT/X:SY:20
```

стек будет содержать значения, описанные в табл. 11.

Таблица 11

Содержимое	Значение
1	2
4	Число переключателей (переключатель X имеет два аргумента и рассматривается как два переключателя)
101530	Последний переключатель (X) связан с 3-им файлом и имеет аргумент
20	Аргумент переключателя X
101530	Следующий переключатель (X) связан с 3-им файлом и имеет аргумент
75250	Аргумент переключателя X-SY в кодах RADIX-50
505	Следующий переключатель (E) связан с 1-м файлом и не имеет аргумента
100102	Переключатель (B) связан с 0-ым файлом и имеет аргумент
24	Аргумент переключателя B

**ПРИМЕЧАНИЕ.** Во многих случаях программе пользователя не требуется обработка переключателей посредством CSI. Однако, возможен непреднамеренный ввод переключателей пользователем. В этом случае рекомендуется сохранить значение указателя стека перед вызовом CSI, а затем восстановить его. Заметим, что командная строка без переключателей перемещает указатель стека на одно слово.

**5.3.7. Запрос .ABTIO** позволяет текущему заданию запретить все операции ввода-вывода на указанном канале, не прерывая выполнения программы.

Запрещается использовать .ABTIO в подпрограммах завершения.

Формат макрокоманды: .ABTIO CHAN

где CHAN — номер канала.

**5.3.8. .CSTAT (режим FB, XM).** Запрос .CSTAT передает программе пользователя информацию о состоянии канала.

Формат макрокоманды:

CSTAT AREA,CHAN,ADDR

где AREA — адрес блока из двух слов аргументов EMT;

CHAN — номер канала, информацию о котором требуется получить;

ADDR — адрес блока из 6 слов для информации о состоянии канала.

Формат блока аргументов:

AREA: .BYTE CHAN,27  
.WORD ADDR

**Пример:** .TITLE CSTAT.MAC

; в этом примере запрос .CSTAT используется для определения устройства, связанного с указанным каналом.

```
START: .MCALL .CSTAT, .CSIGEN, .PRINT, .EXIT
        MOV     SP,R5
        .CSIGEN #DEVSDC, #DEFEXT ; ввод команд-
                                   ; ной строки
        MOV     R5,SP
        .CSTAT  #AREA, #0, #ADDR ; получить ин-
                                   ; формацию о
                                   ; канале
        BCS     NOCHAN           ; канал не от-
                                   ; крыт
        MOV     #ADDR+10,R5
        MOV     (R5)+,R0
        ADD     (PC)+,R0
        .RAD50 / 0/
        ADD     (R5),R0          ; получить
                                   ; имя устрой-
                                   ; ства
        MOV     R0,DEVNAM
        .EXIT
NOCHAN: .PRINT  #MSG
        .EXIT
MSG:    .ASCIZ  /нет выходного файла — канал не открыт/
        .EVEN
AREA:   .BLKW  5
ADDR:   .BLKW  6
DEVNAM: .WORD  0                ; имя устрой-
                                   ; ства
DEFEXT: .WORD  0,0,0,0
DEVSDC =.                        ; адрес за-
                                   ; грузки
                                   ; драйвера
        .END     START
```

**5.3.9. Запрос .DATE** передает информацию о текущей дате из системной ячейки даты в регистр R0. Ячейка даты имеет следующий формат:

разряды 13—10 — месяц;

разряды 9—5 — день;

разряды 4—0 — год.

Значение года в разрядах с 0 по 4 является фактическим годом минус 1972.

Формат макрокоманды:

**.DATE**

Нулевое значение в R0 указывает на то, что пользователь дату не определил.

**Пример: .TITLE DATE.MAC**

; пример подпрограммы обработки даты (введенной по команде DATE) с использованием запроса .DATE, к которой можно обратиться из программы пользователя.

; вых. данные: R0=месяц (1—12)  
; R1=день (1—31)  
; R2=год (две последние цифры)  
; ошибки: R0=0, если дата не введена

```
DATE: .MCALL .DATE
      .DATE ; получить дату из R0
      MOV R0,R2
      BEQ 1$ ; ветвление, если дата не введена
      BIC #37,R2 ; выбрать год
      ADD #72,R2
      MOV R0,R1
      ASL R1 ; выбрать день
      ASL R1
      ASL R1
      SWAB R1
      BIC #37,R1 ; выбрать месяц
      SWAB R0
      ASR R0
      ASR R0
      BIC #37,R0
1$: RETURN ; выход из подпрограммы
   .END
```

**5.3.10. Запрос .DEVICE** позволяет записать определенные значения по указанным адресам после выхода из программы (по .EXIT или CY/C). Эта функция в основном используется для записи в регистры устройств необходимых значений.

В программе может быть несколько запросов .DEVICE и пользователь может связать все таблицы адресов в одну табл.

Запрос .DEVICE игнорируется, если он выдается из виртуального задания в режиме XM.

Формат макрокоманды:

**DEVICE AREA,ADDR[,LINK]**

где AREA — адрес блока из 2 слов аргументов EMT;  
 ADDR — адрес таблицы адресов и значений;  
 LINK — произвольный аргумент, который позволяет связать таблицы адресов и значений последовательных запросов .DEVICE. Если аргумент LINK опущен, то уже существующая таблица заменяется новой при обнаружении очередного запроса .DEVICE.

Таблица адресов и значений может иметь два формата — связанный и не связанный. Оба формата должны оканчиваться нулем. Связанный формат должен иметь в первом слове также ноль. На рис. 5 показаны различные форматы таблицы адресов и значений.

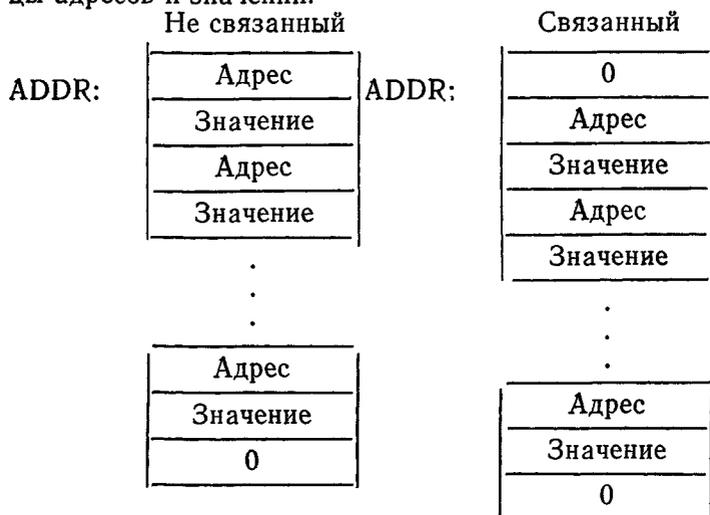


Рис. 5

5.3.11. Запрос .DSTATUS используется для получения информации об определенном устройстве.

Формат макрокоманды:

.DSTATUS DSTBLK,DEVNAM

где DSTBLK — адрес блока из 4 слов состояния устройства;  
 DEVNAM — адрес имени устройства в кодах RADIX-50. Первым словом блока состояния является слово состояния устройства. Возможные значения разрядов слова состояния устройства приведены в табл. 12.

Таблица 12

Разряд	Содержимое	Значение
0—7	0	Кассета магнитных дисков

1	2	3
	2	Логическое устройство регистрации ошибок
	3	Построчно-печатающее устройство
	4	Системный терминал или драйвер BATCH
	6	Гибкие диски с двойной плотностью
	7	Перфоляпточное устройство ввода и вывода
	11	Накопитель на магнитной ленте
	21	Пакет магнитных дисков DP (29 Мгбайт)
	22	Гибкие диски
	25	Фиктивное устройство
	34	Кассетная магнитная лента
	41	Устройство печати последовательного типа
	42	Драйвер обмена между заданиями
	46	Драйвер логического диска
	47	Драйвер расширенной памяти
	50	Диски винчестерского типа или гибкие мини-диски
	51	Редактор командной строки
10	1	Драйвер устройства допускает использование запроса .SPFUN
	0	Не допускается использование запроса .SPFUN
11	—	Резервный
12	1	Устройство несправочной структуры
13	1	Устройство только для записи
14	1	Устройство только для чтения
15	1	Устройство с произвольным доступом к данным
	0	Устройство с последовательным доступом к данным

Второе слово представляет собой длину драйвера устройства в байтах.

Третье слово представляет собой адрес загрузки драйвера плюс 6. Если 0 — драйвера в памяти нет.

Четвертое слово представляет собой емкость устройства (в блоках) для устройств с произвольным доступом к данным или 0 для устройств с последовательным доступом к данным.

Имя устройства в DEVNAM может быть логическим именем, присвоенным пользователем.

Пример использования .DSTATUS см. в описании запроса .CSISPC.

**5.3.12. Запрос .EXIT** вызывает окончание выполнения программы пользователя и передает управление KMON.

Перед выходом из программы выполнение всех иницированных операций ввода-вывода и/или подпрограмм завершения будет закончено. Все невыполненные запросы .MRKT и .CMKT будут аннулированы.

Если часть фоновой программы занимает область памяти KMON и USSR, то она будет записана в блоки для временного хранения данных на системное устройство (если ранее была выполнена команда SET EXIT SWAP). После этого KMON и USSR загружаются в память и управление передается клавиатурному монитору. Если была выполнена команда SET EXIT NOSWAP, то программа пользователя просто перезаписывается.

Если R0=0 во время выполнения .EXIT, то выполняется операция аналогичная .HRESET, запрещающая последующее выполнение команд REENTER, START или CLOSE.

Запрос .EXIT позволяет программе передать команды клавиатурному монитору, используя ячейки 500—777, для их последующего выполнения. Для этого необходимо:

1) ячейка 510 должна содержать общее количество байтов в команде, которая будет передана KMON;

2) команды записываются, начиная с ячейки 512; строка должна быть записана в символах КОИ-7 и не должна содержать <BK> или <ПС>; например:

```
. =510  
.WORD B—A  
A: ASCIZ /COPY A.MAC W.MAC/  
.ASCIZ /DELETE A.MAC/  
B= .
```

3) программа пользователя должна установить 5 или 11 разряд в ССЗ перед выполнением .EXIT; содержимое R0 должно быть нулевым.

Когда запрос .EXIT используется для передачи команд необходимо учитывать следующее:

1) если это свойство используется программой, которая вызывается с помощью косвенного файла, то все команды следующие в косвенном файле за командой, инициирующей пуск этой программы, выполняться не будут;

2) косвенный файл может быть вызван, используя это свойство, только в том случае, если передаваемая KMON строка содержит единственную спецификацию файла; попытка передачи нескольких косвенных файлов или комбинаций косвенных файлов с другими командами KMON приводит к неправильным результатам;

3) .EXIT в подпрограмме завершения недопустим. Заметим, что запрос .EXIT изменяет текущее положение указателя стека.

Формат макрокоманды: .EXIT

**Пример:**

**.TITLE EXIT.MAC**

; в этом примере, после завершения программы, клавиатурно-  
; му монитору передается командная строка на печать спра-  
; вочника устройства DK:

```
    .MCALL .EXIT
CHNIF$ =4000
JSW     =44
START:  MOV     #510,R0
        MOV     #CMDSTR,R1
        MOV     #START,SP
1$:     MOVB    (R1)+,(R0)+
        CMP     R1,#CMDEND
        BLO    1$
        BIS     #CHNIF$,@ #JSW      ; установить
                                     11-й разряд
        CLR     R0                  ; в JSW
                                     ; в R0 должен
                                     ; быть 0
        .EXIT
CMDSTR: .WORD   CMDEND—CMDSTR     ; длина команд-
                                     ; ной строки
        .ASCIZ  «DIRECT/FULL * .MAC» ; командная
                                     ; строка
CMDEND:
        .EVEN
        .END     START
```

**5.3.13. .FETCH, .RELEASES.** Запрос **.FETCH** загружает драйвер устройства из системного устройства в оперативную память. Формат макрокоманды:

**.FETCH ADDR,DEVNAM**

где **ADDR** — адрес загрузки драйвера;

**DEVNAM** — адрес имени устройства в кодах **RADIX-50**.

После выполнения **.FETCH**, **R0** содержит адрес первой, расположенной выше драйвера, ячейки оперативной памяти. Если драйвер уже загружен в память, **R0** будет содержать адрес загрузки **ADDR**.

Для выполнения запросов

**.CLOSE .READC .READ .SFDAT**

**.LOOKUP .WRITE .WRITC .FPROT**

**.ENTER .READW .SPFUN**

**.RENAME .WRITW .DELETE**

необходимо наличие в памяти драйверов соответствующих устройств.

В режиме FB, драйверы для основной программы или системного задания должны загружаться по команде LOAD перед их выполнением.

В режиме XM, драйверы устройств должны загружаться в любую область памяти (фоновую или основную) по команде LOAD перед загрузкой программ.

Пример использования .FETCH см. в описании запроса .CSISPC.

Запрос .RELEASE удаляет драйвер указанного устройства из памяти. Запрос .RELEASE игнорируется, если драйвер:

- 1) является частью RMON (т. е. является драйвером системного устройства);
- 2) отсутствует в памяти;
- 3) был загружен по команде LOAD.

Запрос .RELEASE в основной программе или в системном задании под управлением монитора FB (XM) всегда игнорируется, т. к. мониторы FB или XM могут использовать только те драйверы, которые были загружены по команде LOAD.

Формат макрокоманды:

.RELEASE DEVNAM

где DEVNAM — адрес имени устройства в кодах RADIX-50.

Пример:

.TITLE RELEASE.MAC

; в этом примере драйвер PC: загружается в память, а затем ; по запросу .RELEASE удаляется из памяти.

```
.MCALL .FETCH, .RELEASE, .EXIT, .PRINT
START: .FETCH #DSPACE, #DEVNAM; загрузить драйвер
        BCS      FERR
        .RELEASE #DEVNAM          ; удалить драйвер
        BR      START
FERR:   .PRINT  #NODEV
        .EXIT
DEVNAM: .RAD50  /PC/
NODEV:  .ASCIZ  «Устройство отсутствует!»
        .EVEN
DSPACE:
        .END    START
```

5.3.14. Запрос .FORK используется в драйверах для синхронизации работы системы. Запросу .FORK должен предшествовать запрос .INTEN; .FORK требует для своего выполнения блок памяти из 4 слов. Содержимое регистров R4 и R5 после выполнения запроса сохраняется. .FORK используется в том случае, когда должен быть упорядочен доступ к системным ресурсам или должна быть выполнена не критическая

по времени, но большая по размеру, часть программы. После обработки монитором запроса .FORK прерывание от устройства будет запрещено и драйвер будет работать на нулевом приоритете.

Формат макрокоманды:

.FORK FKBLK

где FKBLK — адрес блока из 4 слов (должен быть в драйвере устройства).

**5.3.15. С помощью запроса .FPROT устанавливается или отменяется защита файла (от удаления).**

На защищенный файл не распространяется действия запросов .CLOSE, .DELETE, .ENTER и .RENAME. Однако данные, хранящиеся в защищенном файле, можно модифицировать. Например, использование запроса .LOOKUP для защищенного файла за которым следует .WRITE, использующий тот же канал, допускается.

Защита файла от удаления реализуется установкой 15-го разряда в слове состояния справочника устройства для указанного файла.

Формат макрокоманды:

.FPROT AREA, CCHAN, DBLK, PROT

где AREA — адрес блока (из 4 слов) аргументов EMT;

CHAN — номер канала (от 0 до 376);

DBLK — адрес блока (из 4 слов), содержащего спецификацию файла в кодах RADIX-50;

PROT — константа, равная 1 (защита устанавливается) или 0 (защита отменяется).

Формат блока аргументов:

AREA: .BYTE CHAN,43  
.WORD DBLK  
.WORD PROT

**Пример:**

.TITLE FPROT.MAC

; в примере используются запросы .FPROT и .SFDAT для установки защиты указанного файла и изменения даты создания файла на текущую дату.

; командная строка вводится с терминала.

```
.MCALL .FPROT, .FETCH, .CSISPC, .DSTATUS, .SFDAT
.MCALL .PRINT, .EXIT
START: .CSISPC #OUTSP, #DEFEXT ;ввод командной строки
       .DSTAT #STAT, #INSPEC ;проверить наличие
                               ;устройства
       TST STAT+4 ;драйвер загружен?
       BNE 1$ ;да
       .FETCH #DRVL0D, #INSPEC ;нет—загрузить драйвер
       BCC 1$
```

```

        .PRINT #LOFAIL
        BR      START
1$:     .FPROT  #EMTBLK,#0,#INSPEC,#1 ;установить защиту
        ;файла
        BCC    2$
        .PRINT #PRFAIL
        BR      START
2$:     .SFDAT  #EMTBLK,#0,#INSPEC,#0 ; изменить дату на
        ;текущую
        BCC    3$
        .PRINT #SDFAIL
        BR      START
3$:     .EXIT
EMTBLK: .BLKW  4
DEFEXT: .WORD  0,0,0
STAT:   .BLKW  4 ;блок состояния устройства
LOFAIL: .ASCIZ  /ошибка запроса .FETCH/
PRFAIL: .ASCIZ  /ошибка запроса .FPROT/
SDFAIL: .ASCIZ  /ошибка запроса .SFDAT/
        .EVEN
OUTSP:  .BLKW  5*3
INSPEC: .BLKW  4*6
DRVLOD: .BLKW  1
        .END   START

```

**5.3.16. Запрос .GTIM** позволяет программе определить текущее время суток. Величина времени определяется в тиках (1 тик = 1/50 с).

Формат макрокоманды:

```
.GTIM AREA,ADDR
```

где AREA — адрес блока из 2 слов аргументов EMT;  
 ADDR — адрес блока из 2 слов, содержащего значение времени во внутреннем формате; первое слово содержит старшие разряды, второе слово — младшие.

Формат блока аргументов:

```
AREA:  .BYTE  0,21
        .WORD  ADDR
```

Пользователь должен предусмотреть в своей программе операцию преобразования тиков в часы-минуты-секунды. В режиме FB величина времени автоматически восстанавливается после 24:00., в режиме SJ операция восстановления выполняется, если при генерации системы была определена поддержка таймера.

**Пример:**

```
.TITLE .GTIM.MAC
```

; в этом примере пользователь получает текущее время в тиках.

```

        .MCALL .GTIM, .EXIT
START:  .GTIM  #AREA, #TICKS ;определить текущее время

```

```

.EXIT
TICKS: .WORD 0,0
AREA: .BLKW 2
.END START

```

5.3.17. Запрос **.GTJB** позволяет программе определить номер выполняемого задания, нижнюю границу памяти и другие параметры задания.

Формат макрокоманды:

```
.GTJB AREA,ADDR[,JOBBLK]
```

где AREA — адрес блока из 2 слов аргументов EMT;

ADDR — адрес блока из 8 или 12 слов для параметров задания (см. табл. 13);

JOBBLK — адрес блока из 3 слов, содержащего логическое имя задания в кодах КОИ-7.

Формат блока аргументов:

```

AREA: .BYTE 0,20
      .WORD ADDR
      .WORD JOBBLK

```

Таблица 13

Слово	Содержимое
1	2
1	Номер задания; в режимах FB, XM без поддержки системных заданий: фоновое — 0, основное — 2; в режимах FB, XM с поддержкой системных заданий: фоновое — 0, системные — 2, 4, 6, 10, 12, 14, основное — 16
2	Верхняя граница области задания
3	Нижняя граница области задания
4	Начало области каналов ввода-вывода
5	Адрес смешанной области заданий для мониторов FB и XM
6	Младший байт — номер системного терминала для систем с мультитерминальной поддержкой
7	В режиме XM — верхний виртуальный адрес для виртуального задания
10—12	В режимах FB, XM с поддержкой системных заданий — логическое имя задания в кодах КОИ-7

**ПРИМЕЧАНИЕ.** Информацию, необходимую для управления выполнением основного задания, монитор хранит в, так называемой, смешанной области, которая создается во время загрузки основной программы в младших адресах основной области памяти.

Слово 4 содержит адрес, с которого начинается область каналов ввода-вывода. Обычно этот адрес находится в пределах RMON. Однако, после выполнения .CDFN, область каналов ввода-вывода заменяется на область, адрес которой указывается пользователем.

**Пример:**

```
.TITLE GTJB.MAC
; в данном примере с помощью запроса .GTJB определяется
; загружено ли основное задание.
; программа может выполняться как под управлением мони-
; тора FB, имеющего поддержку системных заданий, так и под
; управлением монитора FB, который не имеет поддержки
; системных заданий.
.MCALL .GVAL, .GTJB, .PRINT, .EXIT
SYSGEN = 372 ; слово параметров генерации
SYSJOB = 40000 ; разряд системного задания
START: MOV #2, R1
.GVAL #LIST, #SYSGEN ;получить в R0 содержимое
;ячейки монитора
BIT #SYSJOB,R0 ;есть поддержка системного
;задания
BEQ 1$
MOV #16, R1 ;номер основного задания=16
1$: .GTJB #LIST,#JOBARG,R1 ;получить параметры
BCS 2$
.PRINT #FGLOAD
.EXIT
2$: .PRINT #NOFG
.EXIT
LIST: .BLKW 3
JOBARG: .BLKW 12 ;блок параметров задания
FGLOAD .ASCIZ /программа в основной области памяти/
NOFG: .ASCIZ /нет программы в основной области/
.EVEN
.END START
```

**5.3.18. Запрос .GTLIN** используется для получения от пользователя командной информации, отличной от командных строк CSI и позволяет программе работать с косвенными файлами.

Запрос .GTLIN требует наличия \_USR в памяти и принимает входную строку с системного терминала или из косвенного файла, в зависимости от того была ли вызвана программа с терминала или командой из косвенного файла.

Запрос .GTLIN игнорирует аргумент PSTR, если ввод осуществляется из косвенного файла и, если ранее была выполнена команда SET TT QUIT. Однако, если ввод осуществляется с терминала или из косвенного файла, и ранее не была выполнена команда SET TT QUIT, запрос .GTLIN будет

использовать аргумент PSTR для печати побуждающей строки символов перед выборкой входной строки.

Формат макрокоманды:

`.GTLIN SPTR[,PSTR] [,TYPE]`

где SPTR — адрес буфера для записи входной строки; длина строки не должна превышать 81 байт; строка будет ограничена нулевым байтом вместо <ВК> <ПС>;

PSTR — адрес побуждающей строки, которая будет печататься на терминале; PSTR имеет тот же формат, что и аргумент запроса .PRINT.

TYPE — необязательный аргумент, позволяющий осуществить ввод с терминала при выполнении косвенного файла.

ПРИМЕЧАНИЕ. Единственными запросами, которые могут осуществлять ввод из косвенного файла, являются запросы .CSIGEN, .CSISPC и .GTLIN.

Пример:

`.TITLE GTLIN.MAC`

; в этом примере печатается строка, введенная с терминала.

`.MCALL .GTLIN, .PRINT, .EXIT`

**START:** `.GTLIN #BUFF, #PROMT`

`TSTB BUFF`

`BEQ 1$`

`.PRINT #BUFF`

`CLRB BUFF`

`BR START`

`1$: .EXIT`

`BUFF: .BLKW 41.`

`PROMT: .ASCII /введите строку — /<200>`

`.END START`

**5.3.19. .GVAL, .PVAL.** Запрос .GVAL передает в R0 содержимое указанной ячейки монитора.

Формат макрокоманды:

`.GVAL AREA, OFFSET`

где AREA — адрес блока из 2 слов аргументов EMT;

OFFSET — значение смещения относительно начала RMON.

Формат блока аргументов:

`AREA: .BYTE 0,34`

`.WORD OFFSET`

Пример использования .GVAL см. в описании запроса GTJB.

Запрос .PVAL изменяет содержимое указанной ячейки монитора. Старое содержимое записывается в регистр R0.

В режиме XM .GVAL и .PVAL должны использоваться для изменения содержимого ячеек монитора для совместимости с другими мониторами ФОДОС-2.

Формат макрокоманды:  
**.PVAL AREA,OFFSET,VALUE**  
 где AREA — адрес блока из 2 слов аргументов EMT;  
 OFFSET — значение смещения относительно начала RMON;  
 VALUE — новое содержимое ячейки монитора.

Формат блока аргументов:  
**AREA: .BYTE 2,34**  
**.WORD OFFSET**  
**.WORD VALUE**

**5.3.20. Запросы .HERR, .SERR** позволяют задать определенный режим работы монитора при обнаружении серьезных ошибок. Во время выполнения программы могут возникнуть определенные ошибки, вызывающие прерывание программы (см. табл. 14). Обычно эти ошибки вызывают выход из программы и печать монитором сообщения об ошибке. Однако в некоторых случаях невозможно прервать выполнение программы из-за возникших ошибок. Например, многопользовательская программа должна сохранять управление и прервать работу только с тем пользователем, который сделал ошибку. В этом случае полезно использовать запрос **.SERR**, который запрещает монитору удаление из памяти программы, в которой обнаружена ошибка. Вместо этого, при возникновении ошибки после выполнения запроса **.SERR**, устанавливается C-разряд и код ошибки в 52-ом байте. В некоторых случаях запрос **.SERR** оставляет каналы открытыми, например, если использовались запросы **.LOOKUP** и **.ENTER**.

Запрос **.HERR** отменяет действие запроса **.SERR**. В этом случае печатается сообщение об ошибке и программа удаляется из памяти.

Формат макрокоманды:  
**.HERR**  
**.SERR**

Таблица 14

Код	Причина ошибки
1	2
—1	Вызов USR из подпрограммы завершения
—2	В памяти нет драйвера устройства
—3	Ошибка ввода-вывода справочника устройства
—4	Ошибка <b>.FETCH</b> . Ошибка при чтении драйвера, или попытка загрузить его выше USR или RMON
—5	Ошибка чтения оверлейного сегмента

1	2
-6	В справочнике устройства нет свободного места для записи файла
-7	Недопустимый адрес (только в режиме FB) попытка выполнить операцию вне границ задания
-10	Несуществующий номер канала
-11	Недопустимая команда EMT; определен недопустимый функциональный код
-14	Недопустимый справочник
-15	Отсутствует драйвер устройства (в режиме XM)

**Пример:**

.TITLE HERR.MAC

;использование в этом примере запросов .HERR, .SERR  
;позволяет программе пользователя самой обрабатывать  
;серьезные ошибки.

```

.MCALL .HERR, .SERR, .LOOKUP, .PURGE
.MCALL .EXIT, .PRINT, .CSISPC
START: .SERR ;запретить прерывание
;программы по ошибке
;ввод командной строки
.CSISPC #OUTSP, #DEFEXT
.PURGE #0
.LOOKUP #AREA, #0, #OUTSP + 36
BCS ERROR
.HERR ;разрешить прерывание
;по ошибке
.PRINT #LUPOK
.EXIT
ERROR: MOVB @ #52, R0 ;серьезная ошибка?
BMI FTLERR ;да
.PRINT #NOFIL
BR START
FTLERR: NEG R0
DEC R0
ASL R0
MOV TBL(R0), R0
.PRINT
BR START
TBL. C2
C3
C7
C10
C11
C14
C2. .ASCIZ /драйвер устройства не загружен/
C3. .ASCIZ /ошибка чтения справочника/
C7. .ASCIZ /ошибка адресации/
C10. .ASCIZ /недопустимый номер канала/
C11. .ASCIZ /недопустимый код EMT/

```

C14:	.ASCIZ	/недопустимый справочник/
NOFIL:	.ASCIZ	/файл не найден/
LUPOK:	.ASCIZ	/конец программы/
	.EVEN	
AREA:	.BLKW	4
DEFEXT:	.WORD	0,0,0,0
OUTSP:	.BLKW	5*3
INSREC:	.BLKW	4*6
	.END	START

**5.3.21. Запрос .HRESET** прекращает выполнение операций ввода-вывода для вызывающей программы и выполняет функции запроса .SRESET. В режиме SJ запрос .HRESET используется для прекращения всех операций ввода-вывода для задания, в котором используется .HRESET. Все другие операции ввода-вывода продолжаются.

Формат макрокоманды:

**.HRESET**

Пример использования .HRESET см. в описании запроса .CDFN.

**5.3.22. Запрос .INTEN** используется в подпрограмме обработки прерывания программы пользователя и выполняет следующие функции:

- 1) сообщает монитору о том, что произошло прерывание;
- 2) устанавливает приоритет процессора на соответствующий уровень.

Все внешние прерывания вызывают работу процессора по 7-му приоритету. Запрос .INTEN используется для понижения приоритета до уровня, на котором будет работать соответствующее устройство.

Формат макрокоманды:

**.INTEN PRIOR[,PIC]**

где PRIOR — приоритет процессора, который пользователь устанавливает для своей подпрограммы обработки прерывания; обычно тот приоритет, по которому устройство запрашивает прерывание;

PIC — необязательный аргумент; должен быть указан, если подпрограмма обработки прерывания написана в позиционно-независимых кодах; любая подпрограмма обработки прерывания (драйвер устройства), должна быть написана в этих кодах.

**5.3.23. .LOCK, .UNLOCK**

Запрос .LOCK блокирует USR в памяти для выполнения ряда операций. Если существуют условия, вызывающие процесс свопинга, то программа пользователя (та ее часть, которую USR перекрывает в памяти) записывается в блоки диска для временного хранения данных и USR загружается в па-

мять. В противном случае используется `USR`, находящаяся в памяти и свопинга не происходит. Запрос `.LOCK` всегда загружает `USR` в память, если там ее нет. `USR` будет заблокирована в памяти до тех пор, пока не будет сделан запрос `.UNLOCK`. Заметим, что в режиме `FB`, вызов `CSI` может также выполнить действие аналогичное `.UNLOCK`.

В режиме `FB` запрос `.LOCK` запрещает другим заданиям использовать `USR`. После запроса `.LOCK` должен быть выполнен запрос `.UNLOCK` для освобождения `USR`.

Запросы `.LOCK` и `.UNLOCK` дополняют друг друга и, поэтому, должны быть согласованы (т. е., если выполнено три запроса `.LOCK`, то должно быть выполнено, по крайней мере, три `.UNLOCK`, в противном случае `USR` не будет удалена из памяти). Если количество `.UNLOCK` больше чем `.LOCK`, то ошибки не происходит.

Формат макрокоманды: `.LOCK`

#### ПРИМЕЧАНИЯ.

1. Важно, чтобы программа, использующая `.LOCK`, не находилась в области памяти, куда будет загружаться `USR`. Если это произойдет, то после выполнения `.LOCK` управление будет передано не программе пользователя, а `USR`.

2. После выполнения `.LOCK` не рекомендуется использовать область памяти, которую занимает `USR`, даже если дальнейшее использование `USR` не требуется. Это может вызвать непредвиденные результаты при выполнении `.UNLOCK`.

3. Если основное задание выполняет `.LOCK`, в то время как фоновое задание использует `USR`, то выполнение основного задания задерживается.

Запрос `.UNLOCK` удаляет `USR` из памяти, если она была загружена по запросу `.LOCK`. Если при выполнении `.LOCK` использовался свопинг, то `.UNLOCK` загружает программу пользователя обратно в память.

Формат макрокоманды:

`.UNLOCK`

**ПРИМЕЧАНИЕ.** При выполнении двух заданий в режиме `FB` пары `.LOCK/.UNLOCK` должны использоваться только по необходимости, т. к. когда задание блокирует `USR` в памяти, другое задание не может использовать `USR` до тех пор, пока не будет выполнен `.UNLOCK` в первом задании.

**Пример:**

`.TITLE LOCK.MAC`

; пример использования запросов `.LOCK`, `.UNLOCK`

`.MCALL .LOCK, .UNLOCK, .LOOKUP, .SETUP, .PRINT, .EXIT`  
`SYSPTR=54 ;указатель на начало RMON`

```

START:  .SETOP @ #SYSPTR           ;запросить всю свободную
                                           ;память
        .LOCK                      ;вызов USR в память
        .LOOKUP #AREA,#0,#FILF1 ;открыть файл на 0-ом канале
        BCC 1$
2$:     .PRINT #LMSG
        .EXIT
1$:     .PRINT #F1FND
        MOV #AREA,R0
        INC @ R0                   ;номер канала = 1
        MOV #FILE2,2(R0)
        .LOOKUP                    ;открыть файл на 1-ом канале
        BCS 2$
        .PRINT #F2FND
        .UNLOCK                    ;удалить USR из памяти
        .EXIT

AREA:   .BLKW 3
FILE1:  .RAD50 /DK /
        .RAD50 /PIP /
        .RAD50 /SAV/
FILE2:  .RAD50 /DK /
        .RAD50 /DIR /
        .RAD50 /SAV/
LMSG:   .ASCIZ /ошибка .LOOKUP/
F1FND:  .ASCIZ /файл PIP.SAV открыт на 0-ом канале/
F2FND:  .ASCIZ /файл DIR.SAV открыт на 1-ом канале/
        .EVEN
        .END START

```

**5.3.24. Запросы .MFPS и .MTPS** обеспечивают доступ к ССП. Запрос .MFPS используется для выборки ССП.

Формат макрокоманды:

**.MFPS ADDR**

где ADDR — адрес, по которому записывается содержимое ССП; если адрес не указан, содержимое ССП пересылается для установки ССП.

Формат макрокоманды:

**.MTPS ADDR**

где ADDR — адрес слова, которое будет записано в ССП; если адрес не указан, то значение для ССП берется из стека.

**Пример:**

**.TITLE MFPS.MAC**

; в данном примере, с использованием запросов .MFPS,  
; .MTPS, осуществляется обращение к подпрограмме, которая  
; определяет наличие свободного элемента очереди.

```

        .MCALL .MFPS, .MTPS, .EXIT, .PRINT, .TTNR
        JSW = 44                      ;слово состояния задания
        TTSPC$ = 10000                ;разряд специального режима вывода
START:
                                           ;главная программа

```

```

BIS      #TTSPC$, @ #JSW ;установить 12 разряд в JSW
:
CALL     GETQUE           ;вывоз подпрограммы
BCC      1$
.PRINT   #NOELEM
BIC      #TTSPC$, @ #JSW
.EXIT
1$:      NOP               ;продолжение главной
        NOP               ;программы
        .PRINT #GOT1
2$:      .TTINR
        BCS      2$
        BR       START
GETQUE:  MOV      #QHEAD, R4
        TST     @ R4
        BEQ     11$
        .MFPS   ;сохранить значение в стеке
        .MTPS   #340    ;установить приоритет = 7
        MOV     @ R4, R5 ;указатель на следующий
                        ;элемент
        MOV     @ R5, @ R4
        .MTPS   ;восстановить разряды
                        ;приоритета
11$:     TST     (PC) +
        SEC
        RETURN
QHEAD:   .WORD   Q1
Q1:      .WORD   Q2, 0, 0
Q2:      .WORD   Q3, 0, 0
Q3:      .WORD   0, 0, 0
NOELEM:  .ASCIZ  /нет в наличии элемента очереди/
GOT1:    .ASCII  /есть свободный элемент — нажмите любую /
        .ASCIZ  /клавишу для продолжения/
        .EVEN
        .END     START

```

**5.3.25. Запрос .MRKT** планирует запуск подпрограммы завершения через определенный интервал времени.

Запрос .MRKT используют элементы очереди из списка элементов очереди ввода-вывода. Элемент очереди используется до тех пор, пока не будет введена подпрограмма завершения или не будет выполнен запрос .CMKT.

Формат макрокоманды:

```
.MRKT AREA, TIME, CRTN, ID
```

где AREA — адрес блока из 4 слов аргументов EMT;

TIME — адрес блока из 2 слов, содержащего величину интервала времени;

CRTN — точка входа подпрограммы завершения;

ID — число, используемое для идентификации запроса с программой завершения и запросами .CMKT; это число не должно быть в пределах от 177400 до 177777; несколько запро-

сов .MRKT могут иметь один и тот же ID; на входе в подпрограмму завершения R0 будет содержать число ID.

Формат блока аргументов:

```
AREA: .BYTE 0,22
       .WORD TIME
       .WORD CRTN
       .WORD ID
```

Пример:

```
.TITLE TREAD.MAC
; пример использования запросов .MRKT, .CMKT
.MCALL .MRKT, .TTINR, .EXIT, .PRINT, .TTYOUT, .CMKT, .TWAIT
LF = 12
JSW = 44
TCBIT$ = 100
TTSPC$ = 10000
START: .QSET #XQUE,#1 ;добавить элемент очереди
1$: .MOV #PROMT,R0
     .MOV #BUFF,R1
     .CALL TREAD$ ;вызов подпрограммы
     .BCS 2$
     .PRINT #LINE
     .BR 1$
2$: .PRINT #TIMOUT
     .EXIT
TREAD$: .TST R0
        .BEQ 1$
1$: .PRINT
     .CLR TBYT
     .MRKT #TAREA,#TIME,#TOUT,#1 ;запустить подпрограмму
                                   ;завершения по истечении
                                   ;времени
        .BIS #TCBIT$,@ #JSW
        .CLRB @ R1
TTIN: .TWAIT #AREA
      .TTINR
      .BIT #1,(PC) +
TBYT: .WORD 0
      .BNE 2$
      .BCS TTIN
      .MOVB R0,(R1) +
      .CMKT #TAREA,#0 ;отменить .MRKT
2$: .BIS #TTSPC$,@ #JSW
3$: .TTINR
     .MOVB R0,(R1) +
     .BCC 3$
     .CLRB -(R1)
     .BIC #TCBIT$!TTSPC$,@ #JSW
     .ROR TBYT
TOUT: .RETURN
      .INC TBYT
      .RETURN
XQUE: .BLKW 10.
AREA: .WORD 0,WAIT
```

```

TAREA: .BLKW 4
TIME: .WORD 0,500.
WAIT: .WORD 0,1
LINE: .ASCII /обработка данных /
BUFF: .BLKB 81.
PROMT: .ASCIZ /введите данные : / <200>
TIMOUT: .ASCIZ /время на ввод данных истекло/
.END START

```

5.3.26. Запрос **.MTATCH** логически связывает задание с определенным терминалом. Этот запрос должен быть выполнен перед тем как задание начнет использовать терминал в мультитерминальном режиме.

Формат макрокоманды:

```
.MTATCH AREA,ADDR,UNIT
```

где AREA — адрес блока из 3 слов аргументов EMT;

ADDR — адрес слова асинхронного состояния терминала или 0;

UNIT — логический номер терминала.

Формат блока аргументов:

```

AREA: .BYTE 5,37
      .WORD ADDR
      .BYTE UNIT,0

```

5.3.27. Запрос **.MTDTCH** прерывает логическую связь указанного терминала с заданием и делает его доступным для других заданий. Попытка прервать связь с терминалом, логически связанным с другим заданием, приводит к ошибке.

Формат макрокоманды:

```
.MTDTCH AREA,UNIT
```

где AREA — адрес блока из 3 слов аргументов EMT;

UNIT — логический номер терминала.

Формат блока аргументов:

```

AREA: .BYTE 6,37
      .WORD <НЕ ИСПОЛЬЗУЕТСЯ>
      .BYTE UNIT

```

5.3.28. Запрос **.MTSET** позволяет установить характеристики терминала. Он также определяет режим ввода-вывода для указанного терминала.

Формат макрокоманды:

```
.MTSET AREA,ADDR,UNIT
```

где AREA — адрес блока из 3 слов аргументов EMT;

ADDR — адрес блока из 4 слов состояния указанного терминала;

UNIT — логический номер терминала.

Формат блока аргументов:

AREA: .BYTE 0,37  
 .WORD ADDR  
 .BYTE UNIT

На рис. 6 показана структура блока состояния терминала, который организуется пользователем. Описание слов блока состояния приведено в табл. 15.

Структура блока состояния терминала

M.TSTS	
M.TST2	
M.TST3	
M.TSTW	M.TWID

Рис. 6.

Таблица 15

Слово блока состояния терминала	Описание
1	2
M.TSTS	Слово конфигурации терминала
M.TSTS2	Резервное
M.TSTS3	Резервное
M.TWID	Длина строки
M.TSTW	Байт состояния терминала

Значения разрядов слова конфигурации терминала (M.TSTS) даны в табл. 16.

Таблица 16

Номер разряда	Содержимое	Значение
1	2	3
0	1	Аппаратная табуляция
1	2	Выполнение возврата каретки и перевода строки, если действительная длина строки больше установленной в M.TWID
2	4	Аппаратный перевод формата

1	2	3
3	10	Выполнение команд СУ/F и СУ/B
4,5	—	Резервные
6	100	Запрещение режима ожидания для терминала (эквивалентен 6-ому разряду в ССЗ); разряд ТСВIT
7—11	—	Резервные
12	10000	Режим посимвольного ввода (эквивалентен разряду специального режима в ССЗ); разряд TTSPC
13	—	Резервный
14	40000	Запрещение преобразования кодов символов нижнего регистра в коды символов верхнего регистра

В табл. 17 даны значения разрядов для M.TSTW.

Таблица 17

Номер разряда	Содержимое	Значение
1	2	3
10	2000	Терминал используется основным и фоновым заданиями
11	4000	Терминал не включен
14	40000	Дважды подана команда СУ/С
15	100000	Системный терминал

5.3.29. Запрос .MTGET позволяет получить информацию о состоянии указанного терминала. После выполнения запроса блок состояния будет содержать информацию, описанную в табл. 18.

Таблица 18

Относительный номер байта в блоке	Содержимое
1	2
0	Слово конфигурации терминала. Значения разрядов те же, что и в .MTSET
2—4	Резервные
6	Длина строки
7	Текущее положение строки

Формат макрокоманды:  
.MTGET AREA,ADDR,UNIT  
где AREA — адрес блока из 3 слов аргументов EMT;  
ADDR — адрес блока состояния терминала из 4 слов;  
UNIT — логический номер терминала, информацию о состоянии которого требуется получить.

Формат блока аргументов:  
AREA: .BYTE 1,37  
.WORD ADDR  
.BYTE UNIT

**5.3.30. Запрос .MTSTAT** позволяет получить информацию о состоянии мультитерминальной системы.

Формат макрокоманды:  
.MTSTAT AREA,ADDR  
где AREA — адрес 3 словного блока аргументов;  
ADDR — адрес 8 словного блока, содержащего информацию о состоянии мультитерминальной системы (см. запрос .MTGET).

Формат блока аргументов:  
AREA: .BYTE 10,37  
.WORD ADDR  
.WORD 0

**5.3.31. Запрос .MTRCTD** является мультитерминальной формой запроса .RCTRLD. Он отменяет команду CY/D и разрешает вывод на указанный терминал.

Формат макрокоманды:  
.MTRCTD AREA,UNIT, где AREA — адрес блока из 3 слов аргументов EMT;  
UNIT — логический номер терминала.

Формат блока аргументов:  
AREA: .BYTE 4,37  
.WORD. <НЕ ИСПОЛЬЗУЕТСЯ>  
.BYTE UNIT

**5.3.32. Запрос .MWAIT (режим FB, XM)** переводит программу в состояние ожидания до тех пор, пока все сообщения не будут переданы или получены. С помощью этого запроса пользователь может удостовериться в правильности выполнения операции. Запрос .MWAIT используется в основном совместно с запросами .RCVD или .SDAT, которые не предусматривают специальных действий после получения или передачи сообщений (данных).

Формат макрокоманды:  
.MWAIT

### Пример:

; в приведенном примере используется запрос **.MWAIT**.  
; пример состоит из двух программ: фоновой, которая посылает сообщение, и основной программы, которое это сообщение получает.

**.TITLE MWAITF.MAC**

; основная программа ...

```
START:      .MCALL      .MWAIT, .RCVD, .PRINT, .EXIT
            .RCVD      #AREA, #MBUFF, #40. ;получить сообщение
            ;
            ;
            ;
            .PRINT     #FGJOB
            ;
            ;
            ;
            .MWAIT     ;ожидать приема
            TST        MBUFF+2 ;сообщения нет?
            BEQ        FEXIT   ;если нет — выход
                                ;из программы
            .PRINT     #FMSG
            .PRINT     #MBUFF+2 ;печатать полученного сообщения
            BR         START
FEXIT:      .EXIT
AREA:       .BLKW      5
MBUFF:      .BLKW      41.
            .WORD      0
FGJOB:      .ASCIZ     /основная программа готова к приему сообщений/
FMSG:       .ASCIZ     /передано сообщение:/
            .EVEN
            .END START
```

; фоновая программа ...

```
.TITLE      MWAITB.MAC
START:      .MCALL      .MWAIT, .SDAT, .GTIN, .PRINT, .EXIT
            CLR        BUFF
            .GTIN     #BUFF, #PROMPT ;вести сообщение
            .SDAT     #AREA, #BUFF, #40. ;передать сообщение
            BCS       1$
            .MWAIT
            TST       BUFF
            BNE       START
            .EXIT
1$:         .PRINT     #NOFG
            .EXIT
AREA:       .BLKW      5
BUFF:       .BLKW      40.
PROMPT:     .ASCII     /введите сообщение для основной программы/
            .BYTE     15,12,200
NOFG:       .ASCIZ     /основной программы нет/
            .EVEN
            .END      START
```

5.3.33. **.PEEK, .POKE.** Запрос **.PEEK** передает в регистр R0 содержимое указанной ячейки памяти (ниже 28K слов) или содержимое ячейки страницы ввода-вывода.

Запрос **.POKE** изменяет содержимое ячейки памяти. При этом старое содержимое записывается в регистр R0.

Формат макрокоманды:

**.PEEK AREA,ADDR**

где **AREA** — адрес блока из 2 слов аргументов **EMT**;

**ADDR** — адрес ячейки памяти.

Формат блока аргументов:

**AREA: .BYTE 1,34**

**.WORD ADDR**

Формат макрокоманды:

**.POKE AREA,ADDR,VALUE**

где **AREA** — адрес блока из 3 слов аргументов **EMT**;

**ADDR** — адрес ячейки памяти;

**VALUE** — новое содержимое ячейки памяти.

Формат блока аргументов:

**AREA: .BYTE 3,34**

**.WORD ADDR**

**.WORD VALUE**

**Пример:**

; в данном примере запросы **.PEEK, .POKE** используются для чтения и установки размера файла, открытого ранее по запросу **.ENIER**.

```

                .MCALL      .PEEK, .POKE, .EXIT
                RMON       = 54
                MAXBLK    = 314
START:         .PEEK      #EMTBLK, #RMON      ;установить адрес начала
                                                ;RMON
                ADD       #MAXBLK, R0        ;получить размер файла
                MOV       R0, R1
                .POKE     #EMTBLK, R1, #, NEWSIZ;установить новый
                                                ;размер файла
                MOV       R0, OLDSIZ
                .EXIT
EMTBLK:       .BLKW      3
NEWSIZ:       .WORD     100.
OLDSIZ:       .WORD     0                    ;старый размер файла
                .END      START

```

### 5.3.34. **.PROTECT, .UNPROTECT (режим FB, XM)**

Запрос **.PROTECT** используется для получения исключительного права пользования вектором прерывания в диапазоне 0—476 и устанавливает защиту вектора.

Если запрос **.PROTECT** завершен успешно, то это указывает на то, что ячейки вектора не используются другим заданием или монитором и в этом случае текущее задание может

записать адрес подпрограммы обработки прерывания и значения уровня приоритета в эти ячейки.

Формат макрокоманды:

`.PROTECT AREA,ADDR`

где AREA — адрес блока из 3 слов аргументов EMT;

ADDR — адрес вектора, который будет использоваться данным заданием; ADDR должен быть кратен четырем и меньше 476.

Формат блока аргументов:

AREA: .BYTE 0,31  
.WORD ADDR

Запрос `.UNPROTECT` отменяет результат действия запроса `.PROTECT`.

Формат макрокоманды:

`.UNPROTECT AREA,ADDR`

где AREA — адрес блока из 2 слов аргументов EMT;

ADDR — адрес вектора, который освобождается от защиты.

Формат блока аргументов:

AREA: .BYTE 1,31  
.WORD ADDR

**Пример:**

`.TITLE PROTECT.MAC`

; В данном примере показано как с помощью запросов `.PROTECT` и `.UNPROTECT` установить защиту вектора устройства, а затем отменить защиту вектора.

```
START: .MCALL .PROTECT, .UNPROTECT, .PRINT, .EXIT
        .PROTECT #AREA, #300
        BCS 1$
        .UNPROTECT #AREA, #300
        .EXIT
1$: .PRINT #NOVEC
    .EXIT
AREA: .BLKW 3
NOVEC: .ASCIZ /вектор уже защищен !/
        .END START
```

### 5.3.35. QSET

Все операции ввода-вывода в системе ФОДОС осуществляются через централизованную систему управления очередью запросов. Каждый запрос несинхронной передачи (например, `.WRITE`) требует для своего выполнения наличие элемента очереди. Если для очередного запроса ввода-вывода нет свободного элемента очереди, то выполнение программы пользователя будет блокировано до тех пор, пока элемент очереди не освободится. В режиме FB, одно из заданий может выполняться, пока другое задание ожидает элемент очереди.

Запрос `.QSET` используется для того, чтобы увеличить число элементов очереди ввода-вывода. **Общее правило заключается** в том, чтобы в каждой программе количество элементов очереди было на единицу больше, чем общее количество запросов ввода-вывода, которые будут выполняться одновременно на различных каналах.

Запросы `.MRKT`, `.TWAIT` и `.SDAT` также используют элементы очереди и должны приниматься во внимание при определении количества элементов очереди для программы. Каждый раз, когда выполняется `.QSET`, непрерывная область памяти, которая отводится под элементы очереди делится на сегменты, содержащие по 7 слов (по 10 слов для монитора ХМ) и связывается с уже существующей очередью ввода-вывода для этого задания.

Запрос `.QSET` может быть использован неоднократно в программе пользователя. При этом, области памяти для новых элементов очереди, отводимые различными `.QSET`, обязательно должны быть соприкасающимися. Область памяти для дополнительных элементов очереди ввода-вывода должна быть определена внутри программы пользователя. Запросы `.SRESET` и `.HRESET` аннулируют результат действия `.QSET`.

Необходимо соблюдать осторожность при распределении памяти для элементов очереди. Очередь ввода-вывода изменяется монитором асинхронно. Если область, отведенная пользователем для элементов очереди оказалась недостаточной, это приведет к нарушению выполнения программы. Элементы для очереди ввода-вывода нельзя располагать в оверлейной области, а также в области памяти, предназначенной для свопинга `USR`.

Следующие запросы требуют для своего выполнения элемент очереди:

```
.TWAIT .READW .WRITE .SDAT .SDATW  
.MRKT .RCVD .WRITC .READC .SDATC  
.READ .RCVDC .WRITW .RCVDW
```

Формат макрокоманды:

```
.QSET ADDR,LEN
```

где `ADDR` — адрес начала области для новых элементов очереди ввода-вывода;

`LEN` — число новых элементов очереди ввода-вывода.

Пример использования `.QSET` см. в описании запроса `.TWAIT`.

**5.3.36. Запрос `.RCTRLD` обеспечивает готовность системного терминала к печати. Команда `CU/D` запрещает вывод на**

терминал до тех пор, пока не будет повторно подана команда `CV/D` или в программе не будет выполнен запрос `.RCTRLD`, т. е. программа, в которой предусмотрена печать сообщений на терминале, должна предусматривать возможность подачи пользователем команды `CV/D`.

Формат макрокоманды:

`.RCTRLD`

Пример:

```
.TITLE RCTRLD.MAC
.MCALL .RCTRLD, .CSIGEN, .EXIT
START: .RCTRLD
       .CSIGEN #DSPACE, #DEXT, #0
;
;
;
       JMP     START
DEXT:  .WORD  0,0,0,0
DSPACE = .
       .END   START
```

### 5.3.37. `.SPND`, `.RSUM` (режим FB, XM)

Запрос `.SPND` временно приостанавливает выполнение программы пользователя и допускает работу только подпрограмм завершения (для операций ввода-вывода и запросов `.MRKT`).

Формат макрокоманды:

`.SPND`

Запрос `.RSUM` из подпрограммы завершения вновь передает управление программе пользователя.

Формат макрокоманды:

`.RSUM`

#### ПРИМЕЧАНИЯ:

1. Монитор содержит так называемый счетчик приостановки для каждого задания. Этот счетчик уменьшается по `.SPND` и наращивается по `.RSUM`. Задание будет фактически приостановленным, если счетчик будет содержать отрицательное значение. Поэтому, если `.RSUM` будет выполняться перед `.SPND`, последний запрос будет игнорироваться.

2. Программа должна использовать равное количество `.SPND` и `.RSUM`.

3. Запрос `.SPND` из подпрограммы завершения уменьшает счетчик приостановки, но не приостанавливает выполнение программы. Если в подпрограмме завершения выполняется `.SPND`, выполнение программы продолжается до тех пор пока она также не выполнит `.SPND`, после чего выполнение про-

граммы будет приостановлено и для возобновления ее работы потребуется уже два запроса .RSUM.

4. .SPND и .RSUM подобно другим запросам, могут быть выполнены в подпрограммах обработки прерываний, которые написаны пользователем, если ранее были выполнены .INTEN и .SYNCH. Все замечания, относящиеся к .SPND и .RSUM из подпрограмм завершения, имеют силу и для этого случая.

**5.3.38. Запрос .SCCA** выполняет следующие функции:

— блокирует выполнение команды CY/C;

— сообщает о том, что с терминала было подано две команды CY/C>

По запросу .SCCA команда CY/C воспринимается программой как обычная управляющая команда без специальных функций.

Формат макрокоманды:

.SCCA AREA, ADDR

где AREA — адрес блока из 2 слов аргументов EMT;

ADDR — адрес слова состояния терминала.

Слово состояния терминала предназначено для фиксирования подачи двух команд CY/C. Если в нем установлен 15-й разряд, то это значит, что с терминала были последовательно поданы две команды CY/C. Этот разряд должен очищаться программой пользователя.

Если аргумент ADDR равен нулю в запросе .SCCA, монитор восстанавливает системные функции команды CY/C.

Формат блока аргументов:

AREA: .BYTE 0,35  
.WORD ADDR

**Пример:**

```
.TITLE SCCA.MAC
.MCALL .SCCA, .TTYIN, .TTYOUT, .PRINT
JSW = 44
TTSPC$ = 10000
START: MOV #SCCA,R1
.SCCA #AREA,R1
BIS #TTSPC$,C #JSW
CLR (R1)
.PRINT #MSG1
1$: TST (R1)
BNE 2$
.TTYIN
.TTYOUT
BR 1$
2$: .PRINT #MSG2
.SCCA #AREA,#0
BR .
SCCA: .BLKW 1
AREA: .BLKW 2
```

```

MSG1.  ..ASCII  /пример использования запроса .SCCA/<15><12>
        .ASCIIZ \ введите двойное СУ/С \
MSG2:  .ASCII  \ СУ/С введена \ <15><12>
        .ASCIIZ \ для выхода из программы введите двойное СУ/С \
        .EVEN
        .END    START

```

**5.3.39. Запрос .SDTTM** позволяет программе пользователя устанавливать системную дату и время.

Формат макрокоманды:

```
.SDTTM AREA,ADDR
```

где AREA — адрес блока аргументов EMT из 2 слов;

ADDR — адрес блока из 3 слов, содержащего новую дату и время.

Формат блока аргументов:

```

AREA:  .BYTE    0,40
        .WORD    ADDR

```

Первое слово трехсловного блока содержит новую дату во внутреннем формате (см. запрос .DATE). Если первое слово содержит минус 1, то монитор игнорирует его. Второе и третье слова содержат новое значение времени (старшего и младшего порядка соответственно). Если второе слово содержит отрицательную величину, то монитор не изменяет значение времени.

Изменения системной даты или времени не влияет на выполнение запросов, использующих таймер.

**Пример:**

```
.TITLE SDTTM.MAC
```

; в этом примере приведена подпрограмма коррекции времени; ни (на 1 час) с учетом декретного времени.

```

        .MCALL   .SDTTM, .PRINT, .GTIM, .EXIT
        .GLOBL  STD, DALITE
STD.    COM      HR
        NEG      HR+2
DALITE: .GTIM    #AREA, #TIME      ;получить текущее время
        CALL    JADD
        .SDTTM  #AREA, #NEWDT     ;установить новое время
        .GTIM   #AREA, #TIME     ;получить новое время
        RETURN
NEWDT:  .WORD    —1              ;дату не изменять
TIME:   .WORD    0,0
HR      .WORD    2              ;1 час во внутреннем
        .WORD    137440         ;формате
AREA:   .WORD    0,0
JADD:   MOV      #HR, R4
        MOV     #AREA, R3
        MOV     #HR, R1
        MOV     —(R4), R2
        ADD    —(R3), R2
        MOV    —(R4), R5

```

ADC	R5
ADD	—(R3),R5
MOV	R2,—(R1)
MOV	R5,—(R1)
RETURN	
END	

**5.3.40. Запрос .SETTOP** позволяет программе пользователя установить новый адрес в качестве верхней границы программы. Монитор определяет, является ли этот адрес допустимым и нужен ли свопинг `USR`. Например, если запрос определяет адрес ниже начального адреса `USR`, то свопинг `USR` не требуется. Если `.SETTOP` из фонового задания определяет адрес выше начального адреса `USR` и ранее не была выполнена команда `SET USR NOSWAP`, то необходим свопинг `USR`.

После выполнения `.SETTOP`, `R0` и ячейка 50 будут содержать адрес памяти, предоставленный монитором программе пользователя в качестве верхнего адреса программы. Если программа запрашивает по `.SETTOP` недопустимый адрес для данного задания, монитор не удовлетворит это требование, и установит для программы фактически допустимую верхнюю границу памяти. Формат макрокоманды:

`.SETTOP ADDR`

где `ADDR` — запрашиваемый адрес верхней границы программы.

Пример использования `.SETTOP` см. в описании запроса `.LOCK`.

**5.3.41. Запрос .SPCPS (режим FB, XM)** используется в подпрограммах завершения для сохранения содержимого регистров `PC` и `PS` и замену содержимого `PC` на новое значение.

Формат макрокоманды:

`.SPCPS AREA,ADDR`

где `AREA` — адрес блока аргументов `EMT` из 2 слов;  
`ADDR` — адрес блока из 3 слов, содержащего новое значение `PC`, старое значение `PC` и старое значение `PS` соответственно в 1-м, 2-м и 3-м словах.

Монитор сохраняет старые значения `PS` и `PC` и передает управление по адресу, который является содержимым 1-го слова. Формат блока аргументов:

`AREA: .BYTE 0,41`  
`.WORD ADDR`

**Пример:**

`.TITLE SPCPS.MAC`

; пример использования запроса `.SPCPS`.

```

.MCALL .READC, .WRITC, .CSIGEN, .PRINT, .SPCPS
.MCALL .CLOSE, .WAIT, .SRESET, .EXIT
ERRBYT = 52
LSB
ENABL
START: .CSIGEN #DSPACE, #DEFEXT ;ввод командной строки
CALL IOXFER
.PRINT #MES
1$: DEC R5
BR 1$
FINI: .CLOSE #0 ;>0— конец файла
MOV #DONE, R0
BR EXIT
WERR: MOV #WRERR, R0
BR EXIT
RERR: MOV #RDERR, R0
EXIT: .PRINT
.SRESET ;удалить загруженные
;драйверы

WRDONE: .EXIT
.WAIT #0
BCS 3$
IOXFER .READC #AREA, #3, , #6$ ;читать блок
BCC 5$
TSTB @ #ERRBYT ;конец файла?
BEQ 4$ ;да
2$: MOV #RERR, SBLOK
BR 4$
3$: MOV #WERR, SBLOK
4$: TSTB SPCALL
BNE 5$
.SPCPS #AREA, #SBLOK
INCB SPCALL
BCS 7$
5$: RETURN
6$: .WAIT #3
BCS 2$
.WRITC #AREA, #0, , #WRDONE;записать блок
BCS 3$
INC BLOK
RETURN
7$: .PRINT #SPERR
RETURN
AREA: .WORD 0 ;блок аргументов EMT
BLOK: .WORD 0 ;номер блока
.WORD BUFF ;адрес буфера
.WORD 256. ;счетчик слов
.WORD 0 ;адрес подпрограммы завершения
SBLOK: .WORD FINI, 0, 0
BUFF: .BLKW 256.
DEFEXT: .WORD 0, 0, 0, 0
SPCALL: .BYTE 0
.NLIST BEX
DONE: .ASCIZ /конец программы/
MES: .ASCIZ /выполнение программы продолжается/
WRERR: .ASCIZ /ошибка .WRITC/

```

```

RDERR:   .ASCIZ   /ошибка READC/
SPERR    .ASCIZ   /ошибка .SPCPS/
          .EVEN
DSPACE   =.
          .END     START

```

**5.3.42. Запрос .SFDAT** позволяет программе пользователя установить или изменить дату создания указанного файла (как защищенного так и не защищенного).

Формат макрокоманды:

```
.SFDAT AREA,CHAN,DBLK,DATE
```

где AREA — адрес блока аргументов EMT из 3-х слов;

CHAN — номер канала;

DBLK — адрес блока из 4 слов, содержащего спецификацию файла;

DATE — адрес ячейки, содержащей новое значение даты (в формате ФОДОС); если аргумент равен нулю, то используется системная дата.

Формат блока аргументов:

```

AREA:    .BYTE    CHAN,42
          .WORD    DBLK
          .WORD    DATE

```

Пример использования .SFDAT см. в описании запроса .FPROT.

**5.3.43. Запрос .SFPA** позволяет пользователю, работающему с процессором плавающей запятой, установить адрес подпрограммы обработки прерывания от ППЗ. Если адрес подпрограммы обработки прерывания не определен и происходит прерывание от ППЗ, монитор печатает сообщение об ошибке и удаляет программу пользователя из памяти.

Формат макрокоманды:

```
.SFPA AREA,ADDR
```

где AREA — адрес блока из 2 слов аргументов EMT;

ADDR — адрес подпрограммы обработки прерывания от ППЗ.

По окончании работы подпрограммы обработки прерывания от ППЗ, необходимо повторить запрос .SFPA, т. к. монитор запрещает обработку прерываний после обслуживания одного из них. Это делается для того, чтобы исключить возможность закливания программы при многократном повторении прерывания от ППЗ.

При использовании ППЗ, монитор выполняет команду STST—(SP). Поэтому, перед командой RTI, подпрограмма пользователя должна увеличить указатель стека на два.

Формат блока аргументов:

```

AREA:    .BYTE    0,30
          .WORD    ADDR

```

**5.3.44. Запрос .SRESET** выполняет следующие функции:

1) удаляет из памяти драйверы устройств, которые были загружены по запросу .FETCH; драйверы, которые были загружены по команде LOAD, остаются резидентными, так же как и драйвер системного устройства;

2) стирает открытые в настоящее время файлы (файлы, открытые по .ENTER, никогда не станут постоянными);

3) восстанавливает число каналов ввода-вывода (16 (десятичное)); каналы, определенные по .CDFN, аннулируются;

4) восстанавливает очередь ввода-вывода (первоначально — один элемент); QSET должен выполняться повторно для определения дополнительных элементов очереди;

5) очищает очередь подпрограмм завершения.

Формат макрокоманды:

**.SRESET**

Пример использования .SRESET см. в описании запроса .READ.

**5.3.45. Запрос .SYNCH** позволяет в подпрограмме обработки прерывания использовать программные запросы. Без .SYNCH запросы из подпрограммы обработки прерывания не будут обрабатываться монитором.

Формат макрокоманды:

**.SYNCH AREA[.PIC]**

где AREA — адрес блока из 7-ми слов; этот блок необходим при использовании .SYNCH; содержимое блока формируется пользователем (см. табл. 19); PIC — необязательный аргумент, который дает макрокоманде .SYNCH возможность образования PIC — кодов для использования драйверами устройств.

Таблица 19

Слово	Содержимое
1	2
1	Используется только системой; его содержимое не должно изменяться пользователем
2	Номер текущего задания (0 или 2); может быть получен с помощью запроса .GTJB
3,4	Не используются
5	Аргумент R0; при успешном выполнении .SYNCH регистр R0 будет содержать этот аргумент
6	—1
7	0

**ПРИМЕЧАНИЕ.** Между выполнением запросов .INTEN и .SYNCH нельзя выполнять операции со стеком.

**5.3.46. Запрос .TLOCK** может использоваться программами, которые выполняются в режиме FB, для того чтобы сделать USR резидентной, если только USR не занята обслуживанием другого задания. Если USR используется другим заданием, то при выполнении .TLOCK будет установлен С-разряд, что указывает на то, что запрос не выполнен.

Формат макрокоманды:

.TLOCK

**Пример:**

.TITLE TLOCK.MAC

; пример использования запроса .TLOCK.

```

START: .MCALL      .TLOCK, .UNLOCK, .LOOKUP, .PRINT, .EXIT
        TLOCK                      ;USR занята?
        BCS        SUSPND           ;да
        .LOOKUP   #AREA,#4,#FILE   ;открыть файл
        BCS        LKERR
        PRINT     #J1MSG
        .UNLOCK                      ;освободить USR
        TSTB     J2SW
        BNE      1$
        CALL     JOB2
1$      .EXIT
SUSPND: TSTB     J2SW
        BNE      START
        JSR     PC,JOB2
        INC     J2SW
        BR      START
AREA:   .BLKW   5
FILE:   .RAD50 /DK /
        .RAD50 /FILE /
        .RAD50 /TMP/
LKERR:  .PRINT  #LKMSG
        .EXIT
LKMSG:  .ASCIZ  /файл не найден/
J1MSG:  .ASCIZ  /выполняется задание 1/
J2MSG:  .ASCIZ  /выполняется задание 2/
J2SW:   .BYTE   0
        .EVEN
JOB2:   .PRINT  #J2MSG
        RTS    PC
        .END   START

```

**5.3.47. Запрос .TRPSET** позволяет установить адрес подпрограммы обработки прерывания по 4-му и 10-му векторам. Значение С-разряда на входе в подпрограмму указывает на то, какое прерывание произошло: очищенный С-разряд — прерывание по 4 вектору; установленный С-разряд — по 10 вектору. Выход из подпрограммы обработки прерывания пользователя осуществляется по команде RTI.

Если при переполнении стека произошло прерывание по 4-му вектору, то оно не обрабатывается по .TRPSET и на терминале печатается сообщение ?MON—F—STACK OVERFLOW монитором SJ или ?MON—F—TRAP TO 4 монитором FB и XM.

Формат макрокоманды:

TRPSET AREA,ADDR

где AREA — адрес блока из 2 слов аргументов EMT;  
ADDR — адрес подпрограммы обработки прерывания; если ADDR=0, то предыдущий .TRPSET отменяется.

Формат блока аргументов:

AREA: .BYTE 0,3  
.WORD ADDR

Пример:

```
.TITLE TRPSET.MAC
.MCALL .TRPSET, .PRINT, .EXIT
DIVZ =67
START: .TRPSET #AREA, #TRPLOC ; запретить удаление
                                           ; программы
                                           ; недопустимая
                                           ; команда
                                           ; несуществующий
                                           ; адрес
        DIVZ
        TST    C #166666
        .EXIT
TRPLOC:                                     ; подпрограмма об-
                                           ; работки прерыва-
                                           ; ний
        BCS    1$
        .PRINT #TRP4
        BR     2$
1$:     .PRINT #TRP10
        .TRPSET #AREA, #TRPLOC
2$:     RTI
AREA    .WORD 0,0
TRP4:   .ASCIZ /прерывание по 4 вектору/
TRP10:  .ASCIZ /прерывание по 10 вектору/
        .EVEN
        .END START
```

**5.3.48. Запрос .TWAIT** приостанавливает выполнение программы пользователя на указанный интервал времени. Запрос .TWAIT требует для своего выполнения элемент очереди.

Формат макрокоманды:

.TWAIT AREA, TIME

где AREA — адрес блока из 2 слов аргументов EMT;  
 TIME — адрес блока из 2 слов, которые содержат величину интервала времени (первое слово — нижнюю границу интервала, второе слово — верхнюю границу интервала).

Формат блока аргументов:

```
AREA:  .BYTE    0,24
        .WORD    TIME
```

**Пример:**

```

        .TITLE   TWAIT.MAC
        .MCALL   .TWAIT, .QSET, .PRINT, .EXIT
START:  CALL     TASK
1$:     .TWAIT   #AREA, #TIME
        BCS     NOQ
        CALL    TASK
        DEC     COUNT
        BNE    1$
        .PRINT  #EXIT
        .EXIT

TASK:   INC     TCNT
        BIT     #1, TCNT
        BEQ    1$
        .PRINT  #TIK
        RETURN

1$:     .PRINT  #TAK
        RETURN

NOQ:   .PRINT  #QERR
        .EXIT

AREA:   .WORD   0,0
TIME:   .WORD   0,50.*2
COUNT: .WORD   7
TCNT:   .WORD   0
TIK:    .ASCII  /тик -- /<200>
TAK:    .ASCIZ  /так/
EXIT:   .ASCIZ  /конец программы/
QERR:   .ASCIZ  /нет элемента очереди/
        .EVEN
        .END    START
```

**.WAIT**

**5.3.49. Запрос .WAIT** приостанавливает выполнение программы до тех пор, пока не закончится операция ввода-вывода на указанном канале. По этому запросу можно получить

информацию о правильности выполнения операции: если канал не открыт или произошла аппаратная ошибка, устанавливается C-разряд.

В режиме FB(XM) запрос .WAIT, если операция ввода-вывода не завершена, вызывает задержку выполнения одного из заданий и выполнение другого, если оно существует.

Формат макрокоманды:

.WAIT CHAN

Пример использования .WAIT см. в описании запроса .READ.

## ПРИЛОЖЕНИЕ 1

### КОДЫ ОШИБОК ПРОГРАММНЫХ ЗАПРОСОВ

В таблице приведены коды ошибок программных запросов и причины, вызывающие эти ошибки.

Таблица

Запрос	Код ошибки	Причина ошибки
.CDFN	0	Попытка определить меньшее число каналов, чем есть в наличии
.CHCOPY	0	Другое задание не существует или канал (OCHAN) не открыт
CLOSE	1	Канал (CHAN) занят
CMKT	3	Защищенный файл с тем же именем уже существует
CMKT	0	Запрос с указанным ID (ID=0) не существует
CNTXSW	0	В списке адресов указан недопустимый адрес
CSIGEN	0	Недопустимая команда
	1	Указанного устройства нет в таблицах системы
	2	Попытка открыть защищенный файл, имеющий то же имя
	3	Попытка открыть файл по .ENTER неудачна из-за переполнения справочника устройства
	4	Входной файл не найден при выполнении LOOKUP
CSISPC	0	Недопустимая команда
	1	Недопустимое устройство
CSTAT	0	Канал не открыт
DELETE	0	Указанный канал занят
	1	Файл не найден в справочнике устройства

1	2	3
	2	Недопустимая операция
	3	Защищенный файл
.DSTATUS	0	Устройство не найдено в таблицах системы
.ENTER	0	Указанный канал занят
	1	На устройстве не найдена область большая или равная M (LEN=M); или устройство или справочник устройства переполнен
	3	Защищенный файл с тем же именем уже существует
.FETCH	0	Недопустимое имя устройства или в системе отсутствует драйвер указанного устройства
.RELEASES	0	Недопустимое имя устройства
.FPROT	0	Указанный канал занят
	1	Файл не найден
	2	Недопустимая операция
	3	Недопустимое значение аргумента «PROT»
.GTJB	0	Нет задания
.GVAL	0	Указанное значение смещения находится вне области RMON
.PVAL	0	Указанный канал занят. Нет задания
.LOOKUP	0	Нет свободного элемента очереди
.MRKT	0	Несуществующий логический номер терминала (ЛНТ)
.MTATCH	2	Недопустимый функциональный код
	3	Терминал логически связан с другим заданием
	4	Недействительный адрес слова асинхронного состояния терминала (в режиме XM)
.MTDTCH	5	Недопустимый логический номер терминала; терминал не связан с заданием
	1	Несуществующий ЛНТ
	2	Недопустимый функциональный код
.MTGET	3	Недопустимый номер терминала; терминал не связан с заданием
	1	Несуществующий ЛНТ
	2	Недопустимый функциональный код
	3	Терминал связан с другим заданием
	4	Недействительный адрес буфера асинхронного состояния терминала
.MTIN	5	Нет символов в буфере ввода
	0	Недопустимый номер терминала; терминал не связан с заданием
	1	Несуществующий ЛНТ
	2	Недопустимый функциональный код
	3	Недействительный адрес буфера (в режиме XM)
.MTOUT	5	Нет свободного места в буфере вывода
	0	Недопустимый номер терминала, терминал не связан с заданием
	1	

1	2	3
.MTPRNT	2 3 5 1	Несуществующий ЛНТ Недопустимый функциональный код Недействительный адрес буфера Недопустимый адрес терминала; терминал не связан с заданием
.MTRCTD	2 3 5 1	Несуществующий ЛНТ Недопустимый функциональный код Недействительный адрес буфера строки Недопустимый номер терминала; терминал не связан с заданием
.MTSET	2 3 1	Несуществующий ЛНТ Недопустимый функциональный код Недопустимый номер терминала; терминал не связан с заданием
.MTSTAT	2	Несуществующий ЛНТ
.PROTECT	3 5 3 5	Недопустимый функциональный код Недействительный адрес буфера Недопустимый функциональный код Недействительный адрес буфера
.UNPROTECT	0	Указанный вектор уже используется
.RCVD	1	Адрес вектора больше 474 или не кратен четырем
.RCVDC	1	Адрес вектора больше 474 или не кратен четырем
.RCVDW	0	Попытка обращения к несуществующему заданию
.READ	0	Попытка чтения после конца файла
.READC	0	Попытка чтения после конца файла
.READW	0	Попытка чтения после конца файла
.RENAME	1 2 0 1 2 3	Аппаратная ошибка на канале Указанный канал не открыт Указанный канал занят Файл не найден Недопустимая операция Попытка переименования защищенного файла
.REOPEN	0	Указанный канал занят
.SAVESTATUS	0	Указанный канал не открыт
.SDAT	1	Файл открыт по .ENTER или находится на магнитной ленте, поэтому .SAVESTATUS для него недопустим
.SDATC	0	Попытка обращения к несуществующему заданию
.SDATW	0	Попытка обращения к несуществующему заданию
.SFDAT	0	Канал занят, Файл не найден, устройство нефайловой структуры

1	2	3
.SPCPS	0	Попытка использовать .SPCPS в главной программе (а не в подпрограмме завершения)
.SPFUN	1	Предыдущий .SPCPS не выполнен
	0	Попытка чтения или записи после конца файла
	1	Аппаратная ошибка на канале
.TLOCK	2	Канал не открыт
	0	USR уже используется другим заданием
.TTYIN	0	Нет символа в буфере ввода
.TTINR		
.TTYOUT	0	Буфер вывода переполнен
.TOUTR		
.TWAIT	0	Нет свободного элемента очереди
.WAIT	0	Указанный канал не открыт
	1	Аппаратная ошибка на канале во время выполнения предыдущей операции ввода-вывода
.WRITE	0	Попытка записи после конца файла
.WRITC	1	Аппаратная ошибка на канале
	2	Канал не открыт

#### ПРИМЕЧАНИЯ.

1. Для запроса .SYNCH возврат по ошибке происходит в следующих случаях:

- не закончена макрокоманда .SYNCH, использующая тот же блок аргументов EMT;
- указан недопустимый номер задания;
- если произошло удаление или останов программы.

При этом управление передается на следующую за .SYNCH ячейку. Если макрокоманда .SYNCH выполнена успешно, то возврат происходит в следующую за возвратом по ошибке ячейку.

2. При выполнении запросов, не указанных в таблице, ошибок не происходит.

## ПРИЛОЖЕНИЕ 2 СТРУКТУРА БЛОКА SVLK

Таблица

Слово	Разряд	Значение	Описание
1 (Слово состояния канала)	0	1	Аппаратная ошибка на канале N — индекс устройства в таблицах монитора (идентифицирует физическое устройство, которое связано с используемым каналом) На канале выполняется запрос .RENAME На канале выполнен .ENTER; справочник устройства должен модифицироваться по запросу .CLOSE Номер сегмента справочника (1—37 (восьмеричное)), в котором находится запись о текущем файле Обнаружена метка конца файла (EOF) Не используется Канал занят Номер начального блока файла на томе; значение слова равно нулю, если устройство последовательного доступа Длина файла в блоках Не используется Четный байт — число незавершенных операций ввода-вывода в канале; нечетный байт — номер привода устройства, связанного с каналом (от 0 до 7)
	1—5	N	
	6	1	
	7	1	
	8—12		
	13	1	
2	14	—	
	15	1	
3			
4			
5			

### ПЕРЕЧЕНЬ ССЫЛОЧНЫХ ДОКУМЕНТОВ

1. Операционная система ФОДОС-2  
Программирование периферийных устройств  
Руководство системного программиста
2. Операционная система ФОДОС-2  
Монитор расширенной памяти  
Руководство программиста
3. Операционная система ФОДОС-2  
Сообщения системы
4. Операционная система ФОДОС-2  
Описание применения

## **БИБЛИОТЕКА СИСТЕМНЫХ ПОДПРОГРАММ РУКОВОДСТВО ПРОГРАММИСТА**

### **1. НАЗНАЧЕНИЕ И УСЛОВИЯ ПРИМЕНЕНИЯ БИБЛИОТЕКИ СИСТЕМНЫХ ПОДПРОГРАММ**

#### **1.1. Назначение**

Библиотека системных подпрограмм (файл SYSLIB.OBJ) предназначена для выполнения различных системных операций в режиме любого монитора (SJ, FB и XM) операционной системы ФОДОС-2 и обеспечивает:

- выполнение операций ввода-вывода в режимах: синхронном, модифицированном синхронном и асинхронном;
- выполнение операций ввода-вывода в мультитерминальном режиме;
- планирование во времени выполнения асинхронных подпрограмм завершения;
- взаимодействие основного и фонового заданий (только для мониторов FB и XM);
- обслуживание прерываний;
- обслуживание таймера;
- управление файлами (открытие, закрытие, переименование, создание и стирание файлов);
- определение параметров выполняемой программы, а также информации об устройствах и каналах ввода-вывода;
- прием и анализ командных строк;
- выполнение операций с текстовыми данными;
- выполнение операций с целыми числами двойной точности.

Список системных подпрограмм приведен в приложении 1.

#### **1.2. Условия применения**

##### **1.2.1. Требования к техническим средствам**

Для выполнения системных подпрограмм необходима следующая минимальная конфигурация технических средств:

- центральный процессор с блоком плавающей запятой;
- оперативное запоминающее устройство емкостью 16К слов

для работы под управлением мониторов SJ и FB, 32К слов и диспетчер памяти для работы под управлением монитора XM;

— терминал (дисплей типа 15ИЭ-00-013);

— таймер;

— устройство ввода-вывода информации на гибких магнитных дисках типа «МС 5301» или «МС 5305».

### 1.2.2. Требования к программным средствам

Для работы с библиотекой системных подпрограмм необходимы следующие программные средства:

1) компоненты операционной системы ФОДОС-2:

FMON<YY>.SYS — монитор системы;

SWAP.SYS — файл свопинга;

<ZZ>.SYS — драйвер диска;

TT.SYS — драйвер терминала;

LINK.SAV — программа «Редактор связей»;

LIBK.SAV — программа «Библиотекарь»;

MACRO.SAV — транслятор с языка АССЕМБЛЕР;

EDIT.SAV — программа «Редактор текста»;

K13.SAV — программа «Экранный редактор»;

где YY — SJ, FB или XM;

ZZ — RK, DX, DY, MX, MY или RKX, DXX, DYX

2) компоненты системы ФОРТРАН/ФОДОС-2:

FORTRA.SAV — транслятор с языка ФОРТРАН;

FORLIB.OBJ — библиотека ФОРТРАНа.

### 1.2.3. Порядок использования библиотеки системных подпрограмм

Для выполнения системных подпрограмм необходимо предварительно выполнить следующие действия:

1) составить программу на языке ФОРТРАН или языке АССЕМБЛЕР с командами обращения к библиотеке;

2) протранслировать программу и получить объектный модуль программы;

3) связать объектный модуль программы с программами библиотеки и библиотеки ФОРТРАНа и получить загрузочный модуль;

4) запустить полученный загрузочный модуль.

#### ПРИМЕЧАНИЯ.

1. Для использования по умолчанию библиотеки ФОРТРАНа FORLIB.OBJ и оверлейных драйверов их следует включить в библиотеку SYSLIB.OBJ.

Включение модулей FORLIB в библиотеку SYSLIB.OBJ выполняется по команде монитора

LIBRARY/INSERT/REMOVE SYSLIB FORLIB <BK>

?GLOBAL? \$OYRH <BK>  
 ?GLOBAL? <BK>

2. Если в программе, вызывающей подпрограммы библиотеки, используется объявление IMPLICIT, изменяющее тип внешних функций, необходимо явно указать тип функций библиотеки, которые выдают результаты типа целый и вещественный.

#### 1.2.4. Требования к оперативной памяти для ФОРТРАН-программ в основном режиме

Для выполнения в основном режиме ФОРТРАН-программы, вызывающей системные подпрограммы, объем оперативной памяти рассчитывается по формуле:

$$1/2(504 + (35 * N) + (R - 136) + (A * 512)) + Z$$

где N и R — аргументы переключателей /N и /R транслятора с языка ФОРТРАН (по умолчанию N=6, R=136);

A — число одновременно открытых файлов. При использовании двойной буферизации значение A должно быть умножено на 2 (по умолчанию A=1);

Z — дополнительный объем памяти, требуемый для выполнения подпрограмм IQSET, ICDFN, INTSET и системных подпрограмм, вызывающих подпрограммы завершения (см. табл. 1).

Таблица 1

Подпрограмма	Z
1	2
IQSET ICDFN INTSET	10*Q 6*NM 25*IN
Системная подпрограмма, вызывающая подпрограмму завершения	64+R/2

Q — число дополнительных элементов очереди;

NM — число дополнительных каналов;

IN — число обращений к подпрограмме INTSET;

R — аргумент переключателя /R транслятора с языка ФОРТРАН.

Список системных подпрограмм, которые вызывают подпрограммы завершения, приведен в приложении 2.

Если для выполнения ФОРТРАН-программы в основном режиме недостаточно оперативной памяти, печатается сообщение:

PERR 0 NON—FORTRAN ERROR CALL  
(см. [3]).

### 1.2.5. Особенности использования **USR**

Программы на языке АССЕМБЛЕРА, которые обращаются к системным подпрограммам (и подпрограммам библиотеки ФОРТРАНА FORLIB), использующим **USR**, вызывают свопинг **USR**, если она не резидентна. **USR** помещается в область памяти, которая задается указателем стека и занимает 2К слов (10000 (восьмеричное) ячеек).

Необходимо следить за тем, чтобы FORLIB, программы на языке АССЕМБЛЕРА, а также подпрограммы завершения и обслуживания прерываний (и их данные) не располагались в области свопинга **USR**. Выполнение этого требования обеспечивается одним из следующих действий:

- сделать **USR** резидентной;
- задать переключатель /NO SWAP в командной строке транслятора с языка ФОРТРАН;
- запретить свопинг **USR** по команде монитора SET **USR NOSWAP**.

При вызове ФОРТРАН-программой системных подпрограмм, использующих **USR** (если она не резидентна), размер секции OTS\$I должен быть не менее 2К слов, в противном случае необходимо расширить OTS\$I до необходимой длины. Информация о распределении программных секций приведена в документе [2].

### 1.2.6. Требования к подпрограммам завершения

Подпрограммы завершения создаются пользователями на языке ФОРТРАН или языке АССЕМБЛЕРА и не должны иметь более двух параметров. Например, на языке ФОРТРАН заголовок подпрограммы имеет вид:

```
SUBROUTINE CR[ (A1[,A2]) ]
```

где CR — имя подпрограммы завершения;

A1 — слово состояния канала;

A2 — номер используемого канала.

Для подпрограмм завершения на языке АССЕМБЛЕРА A1 принимает значение содержимого R0, а A2 — содержимого R1.

При создании подпрограмм завершения необходимо учитывать следующее:

1) каналы, предназначенные для использования подпрограммами завершения, должны назначаться по подпрограмме IGETC перед обращением к системной подпрограмме, освобождаться по IFREEC и объявляться в COMMON;

2) первое обращение к системной подпрограмме, исполь-

зующей подпрограмму завершения, должно осуществляться из основного задания;

3) файлы, обрабатываемые подпрограммой завершения, должны открываться в основном задании;

4) подпрограммы завершения не должны использовать системные подпрограммы, вызывающие `USR` (см. приложение 2);

5) имена подпрограмм завершения и обслуживания прерываний, вызываемых системными подпрограммами `ISCHED`, `ITIMER`, `IREADF` и другие, должны быть объявлены в `ФОРТРАН`-программе с помощью `EXTERNAL`;

6) выход из подпрограммы завершения на `ФОРТРАНе` должен осуществляться по оператору `RETURN`;

7) выход из подпрограммы завершения на языке `АССЕМБЛЕР` должен осуществляться по команде `RTS PC`;

8) подпрограммы завершения могут быть использованы повторно, но не могут вводиться многократно, то есть основная программа будет работать неправильно, если она будет прервана на уровне завершения;

9) в режиме `SJ` в каждый момент времени должна выполняться только одна подпрограмма завершения.

Управление подпрограммами завершения осуществляется из системных подпрограмм (в которых предусмотрен их вызов) после окончания операции ввода-вывода данных или через заданный интервал времени.

Управление подпрограммами завершения различно в разных режимах. Подпрограммы завершения в режиме `SJ` могут прерывать друг друга, так как они выполняются с тем же приоритетом, что и устройство, прерывание от которого запланировало их. Например, подпрограмма завершения, вызываемая системной подпрограммой `IWRITC` для обслуживания устройства `TT`: выполняется с приоритетом 4. Подпрограмма завершения, вызываемая для обслуживания таймера, выполняется с тем же приоритетом, что и часы системы.

В режиме `FB` и `XM` подпрограммы завершения выполняются с нулевым приоритетом. Они не прерывают друг друга, а устанавливаются в очередь для последовательной обработки.

## 2. ОБРАЩЕНИЕ К ПОДПРОГРАММАМ.

Входные и выходные данные

Обращение к системным подпрограммам осуществляется на языке `ФОРТРАН` и на языке `АССЕМБЛЕР`.

## 2.1. Обращение на ФОРТРАНе

Обращение к системным подпрограммам на языке ФОРТРАН осуществляется как к внешней функции, если необходимо получить значение результата, или как к внешней подпрограмме, если значение результата не требуется.

Формат обращения:

$I = F[(A1, A2, \dots, AN)]$

или

CALL S[(A1, A2, \dots, AN)]

где F — символическое имя функции;

S — символическое имя системной подпрограммы;

AI — фактические параметры.

Например, подпрограмма LOCK может быть вызвана как CALL LOCK

или

I=LOCK

Существует ряд подпрограмм, которые не возвращают результат выполнения в качестве значения функции, поэтому обращение к таким подпрограммам как к внешним функциям не имеет смысла.

Обращение на языке ФОРТРАН осуществляется ко всем системным подпрограммам библиотеки SYSLIB.OBJ.

## 2.2. Обращение на АССЕМБЛЕРе

Обращение к системным подпрограммам на языке АССЕМБЛЕР осуществляется по команде:

JSR PC,S

где PC — символическое имя счетчика команд;

S — символическое имя системной подпрограммы.

До обращения к системной подпрограмме символическое имя этой подпрограммы должно быть объявлено как глобальное, в регистре R5 указан адрес списка параметров подпрограммы, первый элемент которого задает число параметров, второй — адрес первого параметра, третий — адрес второго параметра и т. д.

Программы на языке АССЕМБЛЕР должны запоминать содержимое используемых регистров общего назначения до обращения к системным подпрограммам и восстановить их после выхода из подпрограммы.

Системные функции помещают результат в регистры общего назначения, в зависимости от типа данных результата (см. табл. 2).

Таблица 2

Тип результата	Регистры
1	2
Целый, логический вещественный	R0 R0 — R1: R0 — старшие разряды результата R1 — младшие разряды результата
Двойной точности	R0 — R3: R0 — R2 — старшие разряды результата R3 — младшие разряды результата
Комплексный	R0 — R3: R0, R1 — старшие и младшие разряды действительной части результата R2, R3 — старшие и младшие разряды мнимой части результата

Обращение на языке АССЕМБЛЕР осуществляется ко всем системным подпрограммам библиотеки за исключением подпрограмм, перечисленных в приложении 2.

Пример обращения на языке АССЕМБЛЕР:

```

.GLOBL JMUL
:
:
MOV #LIST,R5
JSR PC,JMUL
CMP #—2,R0
BEQ OVRFL
:
:
LIST: .WORD 3
      .WORD OPR1
      .WORD OPR2
      .WORD RESULT
OPR1: .WORD 100
      .WORD 0
OPR2: .WORD 10
      .WORD 10
RESULT: .BLKW 2
.END

```

2.3. Входными и выходными данными для системных подпрограмм являются данные типа целый, вещественный, двойной точности, логический и текстовый. Запись данных и представление их в машине приведено в документах [2], [4].

### 3. ХАРАКТЕРИСТИКИ ПОДПРОГРАММ

В данном разделе описаны характеристики системных подпрограмм, особенности их выполнения, приведены примеры.

Обращение к системным подпрограммам приведено на языке ФОРТРАН. Число параметров и порядок их следования сохраняется при обращении к подпрограммам на языке АССЕМБЛЕР.

Системные подпрограммы осуществляют:

- 1) операции с файлами;
- 2) передачу данных;
- 3) операции с каналами;
- 4) описание устройств и файлов;
- 5) работу с таймером;
- 6) системные операции;
- 7) операции с данными INTEGER\*4;
- 8) операции с текстовыми данными;
- 9) преобразования RADIX-50;
- 10) работу в мультитерминальном режиме;
- 11) вспомогательные операции.

В табл. 3 приведены условные обозначения часто используемых параметров подпрограмм.

Таблица 3

Обозначение	Описание параметра
1	2
C	Целое или имя переменной типа целый, определяющей номер канала
D	Имя четырехсловного массива типа целый, содержащего спецификацию файла (уст: имфайл. тип) в коде RADIX — 50 (каждый элемент массива занимает одно слово памяти)
B	Целое или имя переменной типа целый, определяющей номер блока в файле, в который записываются или из которого считываются данные
BF	Имя массива типа целый, определяющего область памяти, используемую как буфер ввода-вывода
CR	Имя подпрограммы завершения на языке АССЕМБЛЕР или на языке ФОРТРАН

1	2
AR	Имя четырехсловного массива типа целый, используемого подпрограммой завершения для связи
W	Целое или имя переменной типа целый, определяющей число слов, передаваемых во время операций ввода-вывода
UN	Целое или имя переменной типа целый, определяющей логический номер терминала
LE	Целое или имя переменной типа целый, определяющей максимальное число символов в текстовом данном
ER	Имя переменной типа целый, определяющей признак ошибки
I	Имя переменной типа целый, которой присваивается значение функции
H	Целое или имя переменной типа целый, определяющей время в часах
M	Целое или имя переменной типа целый, определяющей время в минутах (от 00 до 59)
S	Целое или имя переменной типа целый, определяющей время в секундах (от 00 до 59)
T	Целое или имя переменной типа целый, определяющей время в долях секунды (доля секунды — 1/50 секунды для 50-герцового таймера).

### ПРИМЕЧАНИЯ.

1. Под данными типа целый понимаются данные INTEGER\*2.

2. Параметры, указанные в табл. 3, далее в подпрограммах не описываются.

#### 3.1. Операции с файлами

Во время выполнения операций с файлами драйвер соответствующего устройства должен быть резидентным.

##### 3.1.1. CLOSEC/ICLOSE

Назначение: закрытие канала и разрешение его использования в другой операции.

Обращение:

CALL CLOSEC(C[,I])

I=CLOSEC(C)

CALL ICLOSE(C[,I])

I=ICLOSE(C)

Значение функции:

I=0 подпрограмма выполнена;

I=-4 файл с указанным именем существует и защищен.

На устройстве записан второй файл с тем же именем.

Ограничения:

1. CLOSEC нельзя вызывать из подпрограммы завершения или обслуживания прерываний.

2. CLOSEC (или PURGE) может быть выполнена для любого канала, назначенного для ввода или вывода данных. Вызов CLOSEC для неназначенного канала игнорируется.

**ПРИМЕЧАНИЕ.** CLOSEC, которая выполняется для канала, связанного с файлом, открытым по подпрограмме IENTER, вызывает корректировку справочника устройства и файл становится постоянным. Если устройство уже содержит файл с тем же именем и типом файла, старый файл стирается, когда новый становится постоянным. Применение CLOSEC к файлу, открытому с помощью LOOKUP, не изменяет справочника устройства.

**Пример:**

```
REAL*4 DBLK(2)
DATA DBLK/6RSY0NEW,6RFILDAT/
DATA ISIZE/56/
```

```

.
.
.
ICHAIN=IGETC()
IF(ICHAN.LT.0) GOTO 100
IERR=IENTER(ICHAN,DBLK,ISIZE)
IF(IERR.LT.0) GOTO 20
20 GOTO(100,120,130) ABS(IER)
CALL ICLOSE (ICHAN,I)
IF(IEQ.-4) GOTO 20
CALL IFREEC(ICHAN)
CALL EXIT
100 STOP 'NO AVAILABLE CHANNELS'
110 STOP 'CHANNEL ALREADY IN USE'
120 STOP 'NOT ENOUGH ROOM ON DEVICE'
130 STOP 'DEVICE IN USE'
200 STOP 'PROTECTION ERROR'
END
```

### 3.1.2. IDELET

Назначение: стирание файла с заданного устройства.

Обращение: I=IDELET(C,D[,SN])

Описание параметров:

SN — целое или имя переменной типа целый, определяющей номер файла для магнитной ленты кассетного типа или способ поиска файла для магнитной ленты;

-1 — поиск файла по его имени с текущего положения магнитной ленты;

0 — перемотка магнитной ленты и поиск файла по его имени;

N — поиск файла с последовательным номером N (N-го файла от начала ленты).

Значение функции:

I=0 подпрограмма выполнена;

I=1 указанный канал занят;

I=2 указанный файл не найден на устройстве;

I=3 устройство занято;

I=4 файл с указанным именем существует и защищен.

Ограничения: IDELET не может быть вызвана из подпрограммы завершения или обслуживания прерываний.

ПРИМЕЧАНИЕ. После выполнения IDELET указанный канал освобождается.

**Пример:**

```
REAL*4 FILNAM(2)
```

```
DATA FILNAM/6RSY0FTN,6R5 DAT/
```

.

.

```
I=IGETC()
```

```
IF(I.LT.0) STOP 'NO CHANNEL'
```

```
CALL IDELET(I,FILNAM)
```

```
CALL IFREEC(I)
```

### 3.1.3. IENTER

Назначение: открытие канала и резервирование области на томе для записи временного файла с заданным именем.

Обращение: I=IENTER(C,D,L[,SN])

Описание параметров:

L — целое или имя переменной типа целый, определяющей размер резервируемой области на томе:

0 — половина наибольшей свободной области, либо вторая по величине свободная область в зависимости от того, которая больше;

N — число блоков;

I — наибольшая свободная область;

SN — то же, что в п. 3.1.2 и SN=-2:

Значение функции:

I=-1 указанный канал занят;

I=-2 не найдена область большая либо равная N блоков при значении L=N;

I=-3 устройство занято;

I=-4 файл с указанным именем существует и защищен;

I=-5 файл с данным последовательным номером не найден;

I=N подпрограмма выполнена. N — число блоков, зарезервированных для файла (N=0 для устройств нефайловой структуры).

Ограничения: IENTER не может быть вызвана из подпрограммы завершения или обслуживания прерываний.

**Пример:**

```
REAL*4 DBLK(2)
DATA DBLK/6RSYOTEM,6RP TMP/
ICHAN=IGETC()
IF(ICHAN.LT.0) STOP 'NO AVAILABLE CHANNEL'
```

```
IF(IENTER(ICHAN,DBLK,20).LT.0) STOP 'ENTER FAILURE'
```

```
CALL PURGE(ICHAN)
CALL IFREEC(ICHAN)
```

#### 3.1.4. IFPROT

Назначение: установление или отмена защиты файла от стирания.

Обращение: I=IFPROT(C,F,P)

Описание параметров:

C — целое или имя переменной типа целый, определяющей номер канала, на котором будет осуществлена защита;

F — буквенно-цифровой литерал, содержащий спецификацию защищаемого файла в виде уст:имфайл.тип;

P — целое, принимающее значение 1 или 0:

1 — установить защиту файла;

0 — отменить защиту файла.

Значение функции:

I=0 подпрограмма выполнена;

I=1 канал занят;

I=2 файл не найден;

I=3 недопустимая операция;

I=4 недопустимое значение параметра P.

**Пример:**

В данном примере устанавливается защита файла SY:RITAFB.SYS от стирания.

```
ICHAN=IGETC()
IF(ICHAN.LT.0) STOP 'CANNOT ALLOCATE CHANNEL'
I=IFPROT(ICHAN,'SY:RITAFB.SYS',1)
```

```
END
```

### 3.1.5. IRENAM

Назначение: переименование указанного файла на заданном устройстве.

Обращение: I=IRENAM(C,D)

Описание параметров:

D — имя восьмисловного массива, содержащего спецификации файлов в коде RADIX-50:

D(1)—D(4) — старая спецификация файла;

D(5)—D(8) — новая спецификация файла;

Значение функции:

I=0 подпрограмма выполнена;

I=1 указанный канал занят;

I=2 указанный файл не найден на устройстве;

I=3 файл с указанным именем существует и защищен.

ПРИМЕЧАНИЕ. Если файл с указанным именем и типом уже существует на устройстве, то он стирается.

**Пример:**

```
REAL*8 NAME(2)
```

```
DATA NAME/12RDKOFTN2 DAT,12RDKOFTN2 OLD/
```

```
·  
·  
·
```

```
ICHAN=IGETC( )
```

```
IF(ICHAN.LT.0) STOP 'NO CHANNEL'
```

```
CALL IRENAM(ICHAN,NAME)
```

```
CALL IFREEC(ICHAN)
```

### 3.1.6. ISFDAT

Назначение: изменение даты создания файла на устройстве файловой структуры.

Обращение: I=ISFDAT(C,D,IDAT)

Описание параметров:

C — целое или имя переменной типа целый, определяющей номер используемого канала;

D — имя массива типа целый, определяющего спецификацию файла в коде RADIX-50, дата создания которого изменяется;

IDAT — целое, определяющее дату в формате ФОДОС-2.

Значение функции:

I=0 подпрограмма выполнена;

I=1 канал занят;

I=2 файл не найден;

I=3 недопустимая операция.

**Пример:**

В данном примере изменяется дата создания файла MY1:OLD23.DAT на 4 июля 1976 года.

```

REAL*4 FILNAM(2)
DATA FILNAM /6RMY1OLD,6R23 DAT/
IDATE=7*1024+4*32+(1976—1972)
ICHAN=IGETC( )
I=ISFDAT(ICHAN,FILNAM,IDATE)
IF (I.NE.0) STOP 'ERROR DURING ISFDAT CALL'

```

END

### 3.1.7. LOOKUP

Назначение: установление связи указанного канала с устройством и/или существующим файлом для выполнения операций ввода-вывода.

Обращение:

```

I=LOOKUP(C,D[,CT,SN,])
I=LOOKUP(C,JB)

```

Описание параметров:

CT — необязательный параметр, используемый для драйвера магнитной ленты кассетного типа. Целое или имя переменной типа целый. По умолчанию CT=0.

SN — то же, что в п. 3.1.2.

JB — четырехсловный описатель задания, которому передаются или от которого принимаются сообщения. Первое слово может содержать имя устройства в коде RADIX-50, второе, третье и четвертое слово — логическое имя задания, состоящее из 6 символов в коде КОИ-7.

Значение функции:

I=N подпрограмма выполнена. N — число блоков в файле;  
 I=0 выполнен нефайловый LOOKUP на устройстве произвольного доступа или на устройстве нефайловой структуры, или выполнен файловый LOOKUP на магнитной ленте;  
 I=-1 указанный канал занят;  
 I=-2 указанный файл не найден на устройстве;  
 I=-3 устройство занято;  
 I=-4 драйвер магнитной ленты нерезидентен;  
 I=-5 недопустимый параметр для устройства файловой структуры;  
 I=-6 недопустимый параметр для устройства нефайловой структуры.

### ПРИМЕЧАНИЯ.

1. Используемый канал будет занят до выполнения одной из подпрограмм:

CLOSEC или ICLOSE

## ISAVES

### PURGE

2. Если имя файла не задано (второе слово в параметре D равно 0) и устройство имеет файловую структуру, то началом «файла» считается абсолютный нулевой блок устройства. В этом случае говорят о нефайловом LOOKUP. Таким образом можно осуществить ввод-вывод на любой физический блок устройства. Если имя файла определено для устройства нефайловой структуры (например, LP:FILE.TYP), то имя файла игнорируется.

3. Если логическое имя задания в параметре JB не задано, то канал может быть открыт только для ввода данных (по запросу .READ/.READC/.READW) или приема сообщения от другого задания.

#### Примеры:

1. INTEGER\*2 DBLK(4)  
DATA DBLK/3RDK0,3RFTN,3R44 ,3RDAT/  
ICHAN=IGETC()  
IF (ICHAN.LT.0) STOP 'NO CHANNEL'  
IF (IFETCH(DBLK).NE.0) STOP 'BAD FETCH'  
IF (LOOKUP(ICHAN,DBLK).LT.0) STOP 'BAD LOOKUP'  
CALL ICLOSE(ICHAN,I)  
I=ICLOSE()  
CALL IFREEC(ICHAN)

## 2. Использование LOOKUP в системном задании:

```
LOGICAL*1 JNAM(6)
DIMENSION JBLK(4)
EQUIVALENCE (JNAM(1),JBLK(2))
DATA JNAM /'Q','U','E','U','E',0/
DATA JBLK(1) /3RMD /
OPEN A MESSAGE CHANNEL TO 'QUEUE'
ICHN=IGETC()
IF(LOOKUP(ICHN,JBLK).LT.0) STOP 'QUEUE IS NOT RUNNING'
```

### 3.2. Передача данных

#### 3.2.1. IABTIO

Назначение: прерывание ввода-вывода на указанном канале.

Обращение: CALL IABTIO(C)

Описание параметров:

C — целое или имя переменной типа целый, определяющей номер канала, на котором прерывается ввод-вывод.

#### 3.2.2. GTLIN

Назначение: ввод текстового данного с системного терми-

нала или косвенного файла.

Обращение: CALL GTLIN(R[,PR])

Описание параметров:

R — имя массива типа LOGICAL\*1 для приема текстового данного. Число элементов массива R должно быть на единицу больше числа символов текстового данного;

PR — имя массива типа LOGICAL\*1, содержащего побуждающий символ (или последовательность символов).

**ПРИМЕЧАНИЯ:**

1. Размещение текстового данного в массиве приведено в п. 3.8.

2. Число символов вводимого текстового данного должно быть не более 80.

3. Побуждающий символ печатается на терминале подпрограммой для указания начала ввода текстового данного.

**Пример:**

```
LOGICAL*1 INP(80),PROMT(6)  
DATA PROMT/'N', 'A', 'M', 'E', '?', «200/
```

·  
·  
·

```
CALL GTLIN(INP,PROMT)
```

·  
·  
·

### 3.2.3. IRCVD/IRCVDC/IRCVDF/IRCVDW (режим FB)

Подпрограммы группы IRCVD используются в режиме FB с подпрограммами группы ISDAT (п. 3.2.5) для обмена данными (или сообщениями) между основным и фоновым заданиями.

#### 3.2.3.1. IRCVD

Назначение: прием данных от другого задания в синхронном режиме ввода-вывода.

Обращение: I=IRCVD(BF,W)

Описание параметров:

BF — имя массива типа целый, в который записываются данные, передаваемые другим заданием. Число элементов массива должно быть на единицу больше числа передаваемых данных W. Первый элемент массива BF содержит число фактически принятых слов.

Значение функции:

I=0 подпрограмма выполнена;

I=1 попытка обращения к несуществующему заданию.



### 3.2.4. IREAD/IREADC/IREADF/IREADW

Подпрограммы группы IREAD используются с подпрограммами группы IWRITE (п. 3.2.9).

#### 3.2.4.1. IREAD

Назначение: прием данных в буфер памяти по указанному каналу в синхронном режиме ввода-вывода.

Обращение: I=IREAD(W,BF,B,C)

Описание параметров:

BF — имя массива типа целый, предназначенного для записи передаваемых данных. Число элементов массива должно быть не менее W.

Значение функции:

I=N подпрограмма выполнена: N — число принятых блоков (0 — для устройств нефайловой структуры);

I=-1 попытка чтения после признака конца файла;

I=-2 аппаратная ошибка;

I=-3 указанный канал не открыт.

Пример:

```
INTEGER*2 BUFFER(256),RCODE,BLK
```

```
·  
·  
·
```

```
RCODE=IREAD(256,BUFFER,BLK,ICHAN)
```

```
IF(RCODE+1) 1010,1000,10
```

```
IF(IWAIT(ICHAN).NE.0) GOTO 1010
```

```
IF NO ERROR, START HERE
```

```
C  
10
```

```
·  
·  
·
```

```
1000 CONTINUE
```

```
·  
·  
·
```

```
CALL EXIT
```

```
1010 STOP 'FATAL READ'
```

```
END
```

#### 3.2.4.2. IREADC

Назначение: прием данных в буфер памяти по указанному каналу в асинхронном режиме ввода-вывода с использованием подпрограммы завершения на языке АССЕМБЛЕР.

Обращение: I=IREADC(W,BF,B,C,CR)

Описание параметров: BF — то же, что в п. 3.2.4.1.

Значение функции: то же, что в п. 3.2.4.1.

Пример:

INTEGER\*2 IBUF(256),RCODE,IBLK  
EXTERNAL RDCMP

RCODE=IREADC(256,IBUF,IBLK,ICHAN,RDCMP)

### 3.2.4.3. IREADF

Назначение: прием данных в буфер памяти по указанному каналу в асинхронном режиме ввода-вывода с использованием подпрограммы завершения на языке ФОРТРАН.

Обращение: I=IREADF(W,BF,B,C,AR,CR)

Описание параметров: BF — то же, что в п. 3.2.4.1.

Значение функции: то же, что в п. 3.2.4.1.

**Пример:**

```
INTEGER*2 DBLK(4),BUFFER(256),BLKNO
DATA DBLK/3RMX0,3RINF,3RUT ,3RDAT/,BLKNO/0/
EXTERNAL RCMPLT
ICHAN=IGETC()
IF(ICHAN.LT.0) STOP 'NO CHANNEL AVAILABLE'
IF(IFETCH(DBLK).NE.0) STOP 'BAD FETCH'
IF(LOOKUP(ICHAN,DBLK).LT.0) STOP 'BAD LOOKUP'
.
.
20 IF (IREADF(256,BUFFER,BLKNO,ICHAN,DBLK,RCMPLT).LT.0)
+GOTO 100
CALL IWAIT(ICHAN)
BLKNO=BLKNO+1
GOTO 20
.
.
100 CALL ICLOSE(ICHAN,I)
I=ICLOSE()
CALL IFREEC(ICHAN)
.
.
CALL EXIT
END
SUBROUTINE RCMPLT(I,J)
.
.
RETURN
END
```

### 3.2.4.4. IREADW

Назначение: прием данных в буфер памяти по указанному каналу в модифицированном синхронном режиме ввода-вывода.

Обращение: I=IREADW(W,BF,B,C)

Описание параметров: BF — то же, что в п. 3.2.4.1.

Значение функции: то же, что в п. 3.2.4.1.

**Пример:**

```
INTEGER*2 IBUF(1024)
ICODE=IREADW(1024,IBUF,IBLK,ICHAN)
IF (ICODE.EQ.-1) GOTO 100
IF (ICODE.LT.-1) GOTO 200
```

```
ICODE=IWRITW(1024,IBUF,IBLK,ICHAN)
```

### 3.2.5. ISDAT/ISDATC/ISDATF/ISDATW (режим FB)

Подпрограммы группы ISDAT используются в режиме FB с подпрограммами группы IRCVD (см. п. 3.2.3) для обмена данными (или сообщениями). Буфер, содержащий данные, не должен модифицироваться или повторно использоваться до тех пор, пока данные не будут приняты другим заданием.

#### 3.2.5.1. ISDAT

Назначение: передача данных другому заданию в синхронном режиме ввода-вывода.

Обращение: I=ISDAT(BF,W)

Описание параметров:

BF — имя массива типа целый, содержащего данные, которые необходимо передать.

Значение функции:

I=0 подпрограмма выполнена;

I=1 попытка обращения к несуществующему заданию.

**Пример:**

```
INTEGER*2 MSG(40)
```

```
·
·
·
```

```
CALL ISDAT(MSG,40)
```

```
·
·
·
```

```
CALL MWAIT
```

#### 3.2.5.2. ISDATC

Назначение: передача данных другому заданию в асинхронном режиме ввода-вывода с использованием подпрограммы завершения на языке АССЕМБЛЕР.

Обращение: I=ISDATC(BF,W,CR)

Описание параметров: BF — то же, что в п. 3.2.5.1.

Значение функции: то же, что в п. 3.2.5.1.

**Пример:**

```
INTEGER*2 MSG(40)
EXTERNAL RTN
```

.

.

```
CALL ISDATC(MSG,40,RTN)
```

### 3.2.5.3. ISDATF

Назначение: передача данных другому заданию в асинхронном режиме ввода-вывода с использованием подпрограммы завершения на языке ФОРТРАН.

Обращение: I=ISDATF(BF,W,AR,CR)

Описание параметров: BF — то же, что в п. 3.2.5.1.

Значение функции: то же, что в п. 3.2.5.1.

**Пример:**

```
INTEGER*2 MSG(40),SPOT(4)
EXTERNAL RTN
```

.

.

```
CALL ISDATF(MSG,40,SPOT,RTN)
```

### 3.2.5.4. ISDATW

Назначение: передача данных другому заданию в модифицированном синхронном режиме ввода-вывода.

Обращение: I=ISDATW(BF,W)

Описание параметров: BF — то же, что в п. 3.2.5.1.

Значение функции: то же, что в п. 3.2.5.1.

**Пример:**

```
INTEGER*2 MSG(40)
```

.

```
IF (ISDATW(MSG,40).NE.0) STOP'FOREGROUND JOB NOT
+RUNNING'
```

### 3.2.6. ITTINR

Назначение: ввод символа с системного терминала.

Обращение: I=ITTINR

Значение функции:

I>0 подпрограмма выполнена;

I<0 нет символа в буфере ввода (пользователь его не напечатал).

В ФОДОС-2 различают два режима ввода-вывода с терминала — общий и специальный. Режим ввода-вывода определяется значением 12-го разряда слова состояния задания (ССЗ, ячейка 000044). Формат ССЗ приведен в документе

[1]. Если значение 12-го разряда ноль, то говорят об общем режиме ввода-вывода, в противном случае — специальном.

В общем режиме ввода-вывода выполняются следующие действия:

1) монитор отвечает «эхо» — печатью каждого вводимого с терминала символа; коды символов нижнего регистра автоматически преобразуются в коды символов верхнего регистра, если не установлен разряд 14 ССЗ;

2) стирание символа и последовательности символов осуществляется с помощью команд ЗБ и СУ/У соответственно;

3) прежде, чем символ или последовательность символов станут доступными для программы пользователя, должен быть напечатан один из ограничителей: <ВК>, <ПС>, команда СУ/З или СУ/С. Возврат каретки и перевод строки также передаются программе пользователя;

4) восьмеричные коды 175 и 176 преобразуются в восьмеричный код 33.

В специальном режиме ввода-вывода выполняются следующие действия:

1) монитор не отвечает «эхо» — печатью каждого вводимого с терминала символа, кроме СУ/С и СУ/Д;

2) СУ/У и ЗБ не выполняют своих функций;

3) вводимые символы сразу становятся доступными программе пользователя;

4) нет преобразования кодов 175 и 176 в код 33.

#### ПРИМЕЧАНИЯ.

1. В специальном режиме пользователь должен организовать в программе «эхо» — печать вводимых символов, если необходимо проконтролировать ввод. Команды СУ/С и СУ/Д выполняют свои обычные функции. Разряды 12 и 14 ССЗ должны устанавливаться программой пользователя, если требуется специальный режим и символы нижнего регистра. Эти разряды очищаются системой, когда управление передается монитору.

2. В режиме FB, если разряд 6 ССЗ не установлен, программа ITINR приостанавливается в ожидании символа (результат  $I < 0$  не выдается). Если разряд 6 ССЗ установлен, программа не ждет вводимого символа и продолжает выполняться; при этом  $I < 0$ .

3. Установить и/или очистить разряды ССЗ можно по подпрограмме ИРОКЕ (п. 3.11.6).

4. В режиме FB команды СУ/Ф и СУ/В выполняются независимо от состояния разряда 12 ССЗ.

### 3.2.7. ITTOUR

Назначение: вывод символа на системный терминал.

Обращение: I=ITTOUR(CH)

Описание параметров: CH — выводимый символ.

Значение функции:

I=0 подпрограмма выполнена;

I=1 буфер терминала переполнен.

**ПРИМЕЧАНИЯ.**

1. Значение функции I=1 означает, что буфер терминала переполнен и символ не может быть выведен на терминал.

2. В режиме FB I не принимает значение равное 1, если разряд 6 ССЗ не установлен. В этом случае выполнение задания приостанавливается до тех пор, пока не освободится место в буфере терминала. Если разряд 6 ССЗ установлен, задание не ждет освобождения места в буфере, продолжает выполняться, но I в этом случае принимает значение 1.

### 3.2.8. IWAIT

Назначение: ожидание завершения передачи данных по указанному каналу.

Обращение: I=IWAIT(C)

Значение функции:

I=0 подпрограмма выполнена;

I=1 указанный канал не открыт;

I=2 аппаратная ошибка во время выполнения предыдущей операции ввода-вывода.

**ПРИМЕЧАНИЕ.** IWAIT используется обычно с подпрограммами IREAD, IWRITE, ISPFN и не влияет на выполнение подпрограмм завершения.

### 3.2.9. IWRITE/IWRITC/IWRITF/IWRITW

Подпрограммы группы IWRITE используются с подпрограммами группы IREAD (см. п. 3.2.4).

#### 3.2.9.1. IWRITE

Назначение: передача данных из памяти в устройство по указанному каналу в синхронном режиме ввода-вывода.

Обращение: I=IWRITE(W,BF,B,C)

Описание параметров:

BF — имя массива типа целый, содержащего передаваемые данные.

Значение функции:

I=N подпрограмма выполнена: N — число переданных блоков (0 — для устройств нефайловой структуры);

I=-1 попытка записать данные после признака конца файла;

I=-2 аппаратная ошибка;

I=-3 указанный канал не открыт.

### 3.2.9.2. IWRTIC

Назначение: передача данных из памяти в устройство по указанному каналу в асинхронном режиме ввода-вывода с использованием подпрограммы завершения на языке АССЕМБЛЕР.

Обращение:  $I = IWRTIC(W, BF, B, C, CR)$

Описание параметров: BF — то же, что в п. 3.2.9.1.

Значение функции: то же, что в п. 3.2.9.1.

**Пример:**

```
INTEGER*2 IBUF(256)
EXTERNAL(CRTN)
```

.

.

.

```
ICODE = IWRTIC(256, IBUF, IBLK, ICHAN, CRTN)
```

### 3.2.9.3. IWRITF

Назначение: передача данных из памяти в устройство по указанному каналу в асинхронном режиме ввода-вывода с использованием подпрограммы завершения на языке ФОРТРАН.

Обращение:  $I = IWRITF(W, BF, B, C, AR, CR)$

Описание параметров: BF — то же, что в п. 3.2.9.1.

Значение функции: то же, что в п. 3.2.9.1.

**Пример:**

```
INTEGER*2 DBLK(4), BUFFER(256), BLKNO
DATA DBLK/3RMX0, 3RINP, 3RUT, 3RDAT/, BLKNO/0/
EXTERNAL RCMPLT
```

.

.

```
ICODE = IWRITF(256, BUFFER, BLKNO, ICHAN, DBLK, RCMPLT)
```

.

.

```
END
```

```
SUBROUTINE RCMPLT(I, J)
```

.

.

```
RETURN
```

```
END
```

### 3.2.9.4. IWRITW

Назначение: передача данных из памяти в устройство по каналу в модифицированном синхронном режиме ввода-вывода.

Обращение:  $I = IWRITW(W, BF, B, C)$

Описание параметров: BF — то же, что в п. 3.2.9.1.

Значение функции: то же, что в п. 3.2.9.1.



вого задания. В связи с этим подпрограмму PRINT следует использовать для вывода экстренных сообщений.

4. Подпрограмма PRINT может быть использована для печати сообщений подпрограммами завершения. Управление возвращается программе пользователя после того, как все символы будут помещены в выходной буфер терминала.

**Пример:**

```
CALL PRINT('THE COFFEE IS READY')
```

### 3.3. Операции с каналами

Для ввода-вывода данных в системе ФОРТРАН/ФОДОС-2<sup>\*</sup> используется 16 каналов с номерами от 0 до 17(8). Число каналов можно увеличить до 254(10) по подпрограмме ICDFN.

Каналы, используемые для выполнения операций ввода-вывода, должны резервироваться в основной программе или ее подпрограммах, но не в подпрограмме завершения.

Каналы назначаются по подпрограмме IGETC (п. 3.3.5) и возвращаются по IFREEC (п. 3.3.4).

#### 3.3.1. ICDFN

Назначение: увеличение числа доступных каналов ввода-вывода.

Обращение:  $I=ICDFN(NM[,AR])$

Описание параметров:

NM — целое или имя переменной типа **целый**, определяющей число каналов ввода-вывода в десятичной системе счисления,  $16 < NM <= 254$ ;

AR — имя массива типа **целый**, определяющего область памяти для новых каналов.

Значение функции:

$I=0$  подпрограмма выполнена;

$I=1$  попытка определить меньше 16 каналов;

$I=2$  для новых каналов недостаточно зарезервированной области памяти.

Ограничения: ICDFN не может быть вызвана из подпрограммы завершения или обслуживания прерываний.

#### ПРИМЕЧАНИЯ.

1. Подробная информация по определению области памяти для каналов приведена в документе [1].

2. ICDFN рекомендуется использовать в начале основной программы перед выполнением операций ввода-вывода.

3. Каждый раз ICDFN формирует новый набор каналов. Первоначально определенные каналы переопределяются.

4. При использовании ICDFN в основном задании необходима дополнительная память (см. п. 1.2.4).

**Пример:**

**IF(ICDFN(24).EQ.2) STOP 'NOT ENOUGH MEMORY'**

**3.3.2. ICNCRU (режим FB)**

Назначение: открытие канала и установление логической связи с каналом другого задания.

Обращение: **I=ICNCRU(C,OC)**

Описание параметров:

**C** — целое или имя переменной типа целый, определяющей номер канала текущего задания, который используется заданием для приема данных;

**OC** — целое или имя переменной типа целый, определяющей номер канала другого задания, файл которого должен быть связан с каналом **C**.

Значение функции:

**I=0** подпрограмма выполнена;

**I=1** нет другого задания или канал его не открыт;

**I=2** канал **C** занят.

Ограничения: файл, открытый на устройстве последовательного доступа, не может быть скопирован.

**ПРИМЕЧАНИЯ.**

1. Если файл другого задания был открыт для формирования по подпрограмме **IENTER** (см. п. 3.1.3) или программному запросу **.ENTER**, то канал текущего задания указывает на файл, длина которого соответствует длине файла, сформированного к моменту выполнения подпрограммы **ICNCRU**.

2. Программа текущего задания может производить запись в файл, который формируется другим заданием, по каналу копирования так, как если бы текущее задание формировало файл. Если при этом канал другого задания закрыт, то файл не формируется.

**3.3.3. ICSTAT (режим FB)**

Назначение: получение информации о состоянии канала ввода-вывода.

Обращение: **I=ICSTAT(C,AD)**

Описание параметров:

**AD** — имя шестисловного массива типа целый, в который записывается информация о состоянии указанного канала:

**AD(1)** — слово состояния канала;

**AD(2)** — номер начального блока файла;

**AD(3)** — длина файла в блоках;

**AD(4)** — номер последнего блока, записанного в файл с момента открытия файла;

**AD(5)** — номер привода устройства, с которым связан канал;

AD(6) — имя устройства в коде RADIX-50, с которым связан канал.

Формат слова состояния канала приведен в приложении 3.

Значение функции:

I=0 подпрограмма выполнена;

I=1 канал не открыт.

**Пример:**

```
INTEGER*2 AREA(6)
```

```
I=7
```

```
IF (ICSTAT(I,AREA).NE.0) TYPE 99,I
```

```
99 FORMAT(1X,'CHANNEL',14,'IS NOT OPEN')
```

### 3.3.4. IFREEC

Назначение: возвращение номера указанного канала ввода-вывода в список доступных номеров каналов.

Обращение: I=IFREEC(C)

Значение функции:

I=0 подпрограмма выполнена;

I=1 номер канала не принадлежит списку каналов, назначенных по подпрограмме IGETC (п. 3.3.5).

Ограничения: подпрограмма IFREEC не может быть вызвана из подпрограммы завершения или обслуживания прерываний.

**ПРИМЕЧАНИЯ.**

1. Перед вызовом IFREEC канал должен быть закрыт или освобожден с помощью подпрограммы CLOSEC (см. п. 3.1.1) или PURGE (п. 3.3.9).

2. Подпрограмма IFREEC возвращает номера тех каналов, которые были назначены по подпрограмме IGETC.

**Пример:**

```
ICHAN=IGETC()
```

```
IF (ICHAN.LT.0) STOP'CANNOT ALLOCATE CHANNEL'
```

```
·
```

```
·
```

```
·
```

```
CALL IFREEC
```

### 3.3.5. IGETC

Назначение: назначение номера канала (от 0 до 17) для выполнения операций ввода-вывода.

Обращение: I=IGETC ( )

Значение функции:

I=-1 нет свободных каналов;

I=N назначен канал N.

Ограничения: подпрограмма IGETC не может быть вызвана из подпрограммы завершения или обслуживания прерываний.

**Пример:**  
 ICHAN=IGETC()  
 IF (ICHAN.LT.0) STOP'CANNOT ALLOCATE CHANNEL  
 .  
 .  
 .  
 CALL IFREEC(ICHAN)  
 .  
 .  
 .  
 END

### 3.3.6. ILUN

Назначение: определение номера канала, с которым связано логическое устройство.

Обращение: I=ILUN(LN)

Описание параметров:

LN — целое или имя переменной типа целый, определяющей номер логического устройства (от 1 до 99).

Значение функции:

I=1 логическое устройство не открыто;

I=-2 логическим устройством является терминал;

I=N N — номер канала, с которым связано данное логическое устройство.

**Пример:**

```
PRINT 99
99  FORMAT('PRINT LOGICAL UNIT 6, WHICH FURTHER DERAULTS
      +TO LP:')
      LUNRT=ILUN(6)
```

### 3.3.7. IREOPN

Назначение: повторное установление связи указанного канала с файлом, для которого была выполнена подпрограмма ISAVES.

Обращение: I=IREOPN(C,CB)

Описание параметров:

C — целое или имя переменной типа целый, определяющей номер канала, который будет связан с повторно открытым файлом;

CB — имя пятисловного массива, содержащего информацию о состоянии канала:

CB(1) — слово состояния канала;

CB(2) — номер начального блока файла (0 — для устройств последовательного доступа);

CB(3) — длина файла (в блоках по 256 слов);

CB(4) — не используется;

CB(5) — четный байт — число незавершенных операций на заданном канале, нечетный байт — номер привода устройства (0—7), связанного с каналом.

Формат слова состояния канала приведен в приложении 3.

Значение функции:

I=0 подпрограмма выполнена;

I=1 указанный канал занят.

**ПРИМЕЧАНИЯ.**

1. Сочетание подпрограмм ISAVES/IREOPN необходимо применять при одновременной работе с большим числом файлов. Необходимое число файлов может быть открыто по подпрограмме LOOKUP (см. п. 3.1.7), а их состояние сохранено по подпрограмме ISAVES (в этом случае USR вызывается один раз для открытия всех файлов). Если из файла потребуются данные, то IREOPN позволяет программе считать их.

2. Если при выполнении подпрограмм IREAD (см. п. 3.2.4.1) или IWRITE (см. п. 3.2.9.1) драйвер соответствующего устройства не был резидентным, при выполнении IREOPN может произойти ошибка.

**Пример:**

```
INTEGER*2 SAVES(5,10)
```

```
DATA ISVPRT/1/
```

```
:
```

```
:
```

```
:
```

```
CALL ISAVES(ICHAN,SAVES(1,ISVPTR))
```

```
:
```

```
:
```

```
CALL IREOPN(ICHAN,SAVES(1,ISVPTR))
```

### 3.3.8. ISAVES

Назначение: получение информации о состоянии канала. Эта информация дает полное определение файла в системе ФОДОС-2.

Обращение: I=ISAVES(C,CB)

Описание параметров:

CB — имя пятисловного массива типа целый, в который записывается информация о состоянии канала (см. п. 3.3.7).

Значение функции:

I=0 подпрограмма выполнена;

I=1 указанный канал занят;

I=2 вызов подпрограммы недопустим (файл был открыт по подпрограмме IENTER).

Ограничения:

1. ISAVES может использоваться только в том случае, если файл был открыт по LOOKUP (см. п. 3.1.7). Если была использована подпрограмма IENTER (п. 3.1.3), то обращение к ISAVES игнорируется и возникает ошибка.

2. ISAVES не применима к файлам, находящимся на магнитной ленте или магнитной ленте кассетного типа.

**ПРИМЕЧАНИЯ.**

1. После выполнения ISAVES информация о состоянии канала запоминается и указанный канал закрывается. Если требуются сохраненные данные о канале, то используется подпрограмма IREOPEN.

2. Если ISAVES выполнена для файла, который затем стирается, область памяти, занимаемая этим файлом, становится свободной и может быть использована для формирования нового файла по подпрограмме IENTER. В этом случае содержимое файла, состояние которого считалось сохраненным, изменится.

**Пример:**

INTEGER\*2 BLK(5)

IF (ISAVES (ICHAN,BLK).NE.0) STOP'ISAVES ERROR'

### 3.3.9. PURGE

Назначение: освобождение канала для использования в другой операции без выполнения действий, аналогичных действиям в подпрограммах ISAVES и CLOSEC.

Обращение: CALL PURGE (C)

**ПРИМЕЧАНИЕ.** Временный файл, связанный в данный момент с каналом, после выполнения PURGE не становится постоянным. Чтобы файл стал постоянным, необходимо выполнить подпрограмму CLOSEC/ICLOSEC.

## 3.4. Описание устройств и файлов

### 3.4.1. IASIGN

Назначение: запись информации в таблицу логического устройства и отмена значений, назначаемых по умолчанию. Информация об устройстве используется библиотекой ФОРТРАНа при открытии логического устройства.

Обращение: I=IASIGN (LN,IV[,IF[,IS[,IT]]])

Описание параметров:

LN — целое или имя переменной типа целый, определяющей номер логического устройства;

IV — имя однословного массива или имя элемента массива типа целый, содержащего имя устройства в коде RADIX-50; IF — имя трехсловного массива типа целый, содержащего имя и тип файла в коде RADIX-50;

IS — целое или имя переменной типа целый, определяющей число блоков для размещения выходного файла. Если IS=0, резервируется либо половина наибольшей свободной области, либо вторая по величине свободная область. Если IS=-1, резервируется наибольшая свободная область;

IT — целое или имя переменной типа целый, определяющей дополнительные признаки, присваиваемые файлу. Это значение получается сложением величин, которые соответствуют требуемым операциям:

1 — использовать два буфера для вывода;

2 — открыть файл как временный;

4 — открыть файл для ввода-вывода. Первая операция ввода-вывода определяет использование файла. Если файл открыт для ввода, он рассматривается как новый файл, если открыт для вывода — как старый;

8 — использовать информацию об управлении кареткой;

16 — не использовать информацию об управлении кареткой;

32 — файл только для считывания.

Значение функции:

I=0 подпрограмма выполнена;

I < > 0 логическое устройство занято или нет места для размещения другого логического устройства (знак «< >» означает «не равно»).

#### ПРИМЕЧАНИЯ.

1. В качестве параметров IV, IF, IS могут использоваться соответствующие элементы массива F подпрограммы ICSI (п. 3.4.2).

2. Информация об управлении кареткой используется только для форматных записей и означает, что при выводе на логическое устройство первый символ выводимой записи используется для управления кареткой. Первый символ не выводится, по нему только выполняется соответствующее действие.

3. Если информация об управлении кареткой не используется (IT=16), то первый символ записи не анализируется. В этом случае библиотека ФОРТРАНа выводит символ перевода строки, за которым следует запись и возврат каретки.

4. Если параметр IT не задан, библиотека ФОРТРАНа анализирует первый символ записи для управления кареткой

только на системном терминале и построчно-печатающем устройстве.

5. IASIGN должна быть вызвана до того, как будет открыто логическое устройство, т. е. до выполнения одного из операторов READ, WRITE, PRINT, TYPE, ACCEPT, которые требуют обращения к логическому устройству.

6. Подпрограмму IASIGN можно использовать с подпрограммой ICSI, чтобы программа пользователя могла принять стандартное описание ввода интерпретатора командной строки CSI.

**Пример:**

```
INTEGER*2 SPEC(39)
REAL*4 EXT(2)
DATA EXT/6RDATDAT,6RDATDAT/
```

```
10 IF (ICSI(SPEC,EXT,,0).NE.0) GOTO 10
```

```
CALL IASIGN(3,SPEC(1),SPEC(2),SPEC(5),1)
CALL IASIGN(4,SPEC(16),SPEC(17),0,32)
```

### 3.4.2. ICSI

Назначение: синтаксический анализ командной строки в специальном режиме работы интерпретатора командной строки и прием спецификаций файлов и переключателей.

Обращение: I=ICSI(F,DT,[CS],[OP],X)

Описание параметров:

F — имя 39-словного массива типа целый, предназначенного для приема спецификаций файлов:

F(1) — имя устройства для выходного файла 1;

F(2—4) — имя и тип выходного файла 1;

F(5) — число блоков выходного файла 1;

F(6) — имя устройства для выходного файла 2;

F(7—9) — имя и тип выходного блока 2;

F(10) — число блоков выходного блока 2;

F(11) — имя устройства для выходного файла 3;

F(12—1) — имя и тип выходного файла 3;

F(15) — число блоков выходного файла 3;

F(16) — имя устройства для входного файла 1;

F(17—19) — имя и тип входного файла 3;

F(20) — имя устройства для входного файла 2;

F(21—23) — имя и тип входного файла 2;

F(24) — имя устройства для входного файла 3;  
F(25—27) — имя и тип входного файла 3;  
F(28) — имя устройства для входного файла 4;  
F(29—31) — имя и тип входного файла 4;  
F(32) — имя устройства для входного файла 5;  
F(33—35) — имя и тип входного файла 5;  
F(36) — имя устройства для входного файла 6;  
F(37—39) — имя и тип входного файла 6;

DT — имя четырехсловного массива типа целый, содержащего типы файлов в коде RADIX-50, назначаемые по умолчанию, если в командной строке не указаны типы файлов:

DT(1) — тип для всех входных файлов;  
DT(2) — тип выходного файла 1;  
DT(3) — тип выходного файла 2;  
DT(4) — тип выходного файла 3;

CS — имя массива типа LOGICAL\*1, содержащего командную строку в коде КОИ-7. Строка должна заканчиваться нулевым байтом. Если параметр не задан, печатается звездочка (\*) и осуществляется прием командной строки с терминала.

OP — имя массива типа целый со списком границ (4,N), предназначенного для приема переключателей, где N — число используемых переключателей. Элементы массива, соответствующие переключателю K ( $1 \leq K \leq N$ ):

OP(1,K) — имя переключателя (один символ КОИ-7);

OP(2,K) — устанавливается подпрограммой:

0 — переключатель не используется;

1 — переключатель не имеет параметра;

2 — переключатель имеет параметр;

OP(3,K) — устанавливается подпрограммой в значение, равное номеру файла, для которого задан переключатель;

OP(4,K) — устанавливается подпрограммой на определенное значение, если элемент массива OP(2,K) равен 2.

Параметр OP используется, если параметр X не равен нулю.

X — число переключателей, описанных в массиве OP.

Значение функции:

I=0 подпрограмма выполнена;

I=1 недопустимая командная строка;

I=2 недопустимое устройство в командной строке;

I=3 недопустимый переключатель или использование переключателя в командной строке большее число раз, чем определено в массиве переключателей.

Ограничение: ICSI нельзя вызывать из подпрограммы завершения или обслуживания прерываний.

#### ПРИМЕЧАНИЯ.

1. В специальном режиме CSI не осуществляет загрузки драйвера и действий, аналогичных действию программных запросов .CLOSE, .ENTER, .LOOKUP.

2. Массив OP должен содержать имена всех используемых переключателей. Например, для задания имен пяти переключателей используются следующие операторы:

```
INTEGER*2 SW(4,5)
DATA SW(1,1)'/S',SW(1,2)'/M',SW(1,3)'/I'
DATA SW(1,4)'/L',SW(1,5)'/E'
```

Пример:

```
INTEGER*2 SPEC(39)
REAL*4 EXT(2)
DATA EXT/6RDATDAT,6RDATDAT/
10 TYPE 99
99 FORMAT('ENTER VALID CSI STRING WITH NO OPTIONS')
IF (ICSI(SPEC,EXT,,0).NE.0) GOTO 10
```

### 3.5. Работа с таймером

#### 3.5.1. CVTTIM

Назначение: преобразование времени из внутреннего представления в часы, минуты, секунды и доли секунды.

Обращение: CALL CVTTIM(TM,H,M,S,T)

Описание параметров:

TM — имя переменной типа INTEGER\*4 или имя двухсловного массива типа INTEGER\*2, определяющего время во внутреннем представлении.

#### ПРИМЕЧАНИЯ.

1. Время во внутреннем представлении задается в десятичной системе счисления в долях секунды как данное типа INTEGER\*4. Однако, в отличие от представления данных INTEGER\*4 (п. 3.7), ячейка с младшим адресом должна содержать старшие разряды данного, ячейка со старшим адресом — младшие разряды данного.

2. Если параметр TM задан в виде массива INTEGER\*2, то первый элемент массива определяет старшие разряды данного, второй — младшие разряды данного.

Пример:

```
INTEGER*4 ITIME
```

```
CALL GTIM(ITIME)
CALL CVTTIM(ITIME,IHRS,IMIN,ISEC,ITCK)
IF(IHRS.GE.12.AND.IHRS.LT.13) GOTO 100
```

### 3.5.2. GTIM

Назначение: определение текущего времени во внутреннем представлении. Текущее время определяется с начала суток.

Обращение: CALL GTIM(IE)

Описание параметров:

IE — имя переменной типа INTEGER\*4 или двухсловного массива типа INTEGER\*2 для приема результата.

ПРИМЕЧАНИЯ.

1. Если в системе нет таймера, то IE принимает значение равное нулю.

2. Если время не установлено по команде монитора TIME, то IE определяет время с момента загрузки системы.

3. Для преобразования времени в часы, минуты, секунды и доли секунды может быть использована подпрограмма CVTTIM.

4. В режиме FB значение времени автоматически восстанавливается после 24:00; в режиме SJ восстановление отсутствует.

Пример:

INTEGER\*4 JTIME

.

.

CALL GTIM(JTIME)

### 3.5.3. ICMKT

Назначение: отмена выполнения подпрограммы завершения, запланированной по подпрограмме ISCHED/TIMER/MRKT.

Обращение: I=ICMKT(ID,MT)

Описание параметров:

ID — целое или имя переменной типа целый, определяющей идентификатор отменяемой подпрограммы завершения. Если ID=0, отменяются все запланированные подпрограммы завершения.

MT — имя переменной типа INTEGER\*4 или имя двухсловного массива типа INTEGER\*2, в который записывается значение времени во внутреннем представлении, оставшегося до выполнения подпрограммы завершения.

Значение функции:

I=0 подпрограмма выполнена;

I=1 подпрограмма завершения с указанным ID (не равным 0) не существует.

**Пример:**  
 INTEGER\*4 J  
 .  
 .  
 .  
 CALL ICMKT(0,J)  
 .  
 .  
 .  
 END

#### 3.5.4. ISCHED (режим FB)

Назначение: планирование выполнения подпрограммы завершения на языке ФОРТРАН в заданное время суток.

Обращение: I=ISCHED(H,M,S,T,AR,ID,CR)

Описание параметров:

ID — целое или имя переменной типа целый, определяющей идентификатор подпрограммы завершения. Значение параметра ID передается в качестве фактического параметра подпрограмме CR;

CR — имя подпрограммы завершения на языке ФОРТРАН, которая должна выполняться в указанное время суток.

Значение функции:

I=0 подпрограмма выполнена;

I=1 подпрограмма не выполнена (нет свободного элемента очереди).

#### ПРИМЕЧАНИЯ.

1. Подпрограмма завершения

```
SUBROUTINE CR(ID)
  INTEGER ID
```

должна иметь один параметр, который получает значение, равное значению идентификатора в соответств. вызове ISCHED.

2. Выполнение планируемой подпрограммы завершения может быть отменено по подпрограмме ICMKT (см. п. 3.5.3).

3. Если система занята, подпрограмма завершения может быть выполнена позднее запрошенного времени.

4. Подпрограммы на ФОРТРАНе могут быть перепланированы по времени выполнения подпрограммами ISCHED или ITIMER.

5. ISCHED использует элемент очереди, что необходимо учитывать при обращении к подпрограмме IQSET (п. 3.6.6).

#### Пример:

```
INTEGER*2 LINK(4)
EXTERNAL NOON
```

I=ISCHED(12,0,0,0,LINK,0,NOON)

END  
SUBROUTINE NOON(ID)

STOP 'ABORT JOB LUNCHTIME'

### 3.5.5. ISDTTM

Назначение: установка системной даты и времени.

Обращение: CALL ISDTTM(DT,HT,LT)

Описание параметров:

DT — целое, определяющее новое значение системной даты в формате даты ФОДОС;

HT — целое, определяющее старшие разряды времени в долях секунды;

LT — целое, определяющее младшие разряды времени в долях секунды.

Значение любого параметра равно «-1» оставляет соответствующее ему значение даты или времени неизменным.

**Пример:**

IDATE=IMONTH\*1024+IDAY\*32+(IYEAR-1972)

CALL ISDTTM(IDATE,-1,-1)

### 3.5.6. ISLEEP (режим SJ)

Назначение: задержка выполнения основного задания на заданный промежуток времени. Время — в часах, минутах, секундах и долях секунды.

Обращение: I=ISLEEP(H,M,S,T)

Значение функции:

I=0 подпрограмма выполнена;

I=1 подпрограмма не выполнена (нет свободного элемента очереди).

**ПРИМЕЧАНИЯ.**

1. ISLEEP использует элемент очереди, что необходимо учитывать при обращении к IQSET (п. 3.6.6).

2. Если система занята, основное задание может быть выполнено позднее запрошенного времени.

3. ISLEEP не влияет на выполнение подпрограмм завершения.

**Пример:**

·  
·  
·  
CALL IQSET(2)  
·  
·  
·

CALL ISLEEP(0,0,0,4)

### 3.5.7. ITIMER

Назначение: планирование выполнения подпрограммы завершения на языке ФОРТРАН по истечении заданного промежутка времени.

Обращение: I=ITIMER(H,M,S,T,AR,ID,CR)

Описание параметров:

ID — целое или имя переменной типа целый, определяющей идентификатор подпрограммы завершения. Значение параметра ID передается в качестве фактического параметра подпрограмме CR;

CR — имя подпрограммы завершения на языке ФОРТРАН, которая будет выполняться по истечении заданного промежутка времени.

Значение функции:

I=0 подпрограмма выполнена;

I=1 подпрограмма не выполнена (нет свободного элемента очереди).

**ПРИМЕЧАНИЯ.**

1. Подпрограмма завершения:

SUBROUTINE CR(ID)

INTEGER ID

должна иметь один параметр, который получает значение, равное значению идентификатора в соответствующем вызове ITIMER.

2. Выполнение планируемой подпрограммы завершения может быть отменено по подпрограмме ICMKT.

3. Если система занята, подпрограмма завершения может быть выполнена позднее запрошенного времени.

4. Подпрограммы на ФОРТРАНе могут быть перепланированы по времени выполнения подпрограммами ISCHED и IQSET.

5. ITIMER использует элемент очереди, что необходимо учитывать при обращении к подпрограмме IQSET.

```

Пример:
INTEGER*2 AREA(4)
EXTERNAL WATCHD
.
.
CALL ITIMER(0,12,0,0,AREA,3,WATCHD)
.
.
CALL ICMKT(3,AREA)
.
.
END
SUBROUTINE WATCHD(ID)
.
.
RETURN
END

```

### 3.5.8. ITWAIT (вариант SJ)

Назначение: задержка выполнения основного задания на заданный промежуток времени. Время — во внутреннем представлении.

Обращение: I=ITWAIT(IM)

Описание параметров:

IM — имя переменной типа INTEGER\*4 или имя двухсловного массива типа INTEGER\*2, определяющего значение промежутка времени во внутреннем представлении.

Значение функции:

I=0 подпрограмма выполнена;

I=1 подпрограмма не выполнена (нет свободного элемента очереди).

**ПРИМЕЧАНИЯ.**

1. ITWAIT использует элемент очереди, что необходимо учитывать при обращении к IQSET (п. 3.6.6).

2. Если система занята, задание может быть выполнено позднее запрошенного времени.

3. ITWAIT не влияет на выполнение подпрограмм завершения.

**Пример:**  
 INTEGER\*2 TIME(2)

CALL ITWAIT(TIME)

### 3.5.9. IUNTIL (вариант SJ)

Назначение: задержка выполнения основного задания до заданного времени суток.

Обращение: I=IUNTIL(H,M,S,T)

Значение функции:

I=0 подпрограмма выполнена;

I=1 подпрограмма не выполнена (нет свободного элемента очереди).

#### ПРИМЕЧАНИЯ.

1. IUNTIL использует элемент очереди, что необходимо учитывать при обращении к подпрограмме IQSET (п. 3.6.6).

2. Если система занята, основное задание может быть выполнено позднее запрошенного времени.

3. IUNTIL не влияет на выполнение подпрограмм завершения.

#### Пример:

CALL IUNTIL(13,0,0,0)

### 3.5.10. JTIME

Назначение: преобразование времени во внутреннее представление.

Обращение: CALL JTIME(H,M,S,T,MT)

Описание параметров:

MT — имя переменной типа INTEGER\*4 или имя двухсловного массива типа INTEGER\*2, в который записывается время во внутреннем представлении.

#### Пример:

INTEGER\*4 J1

CALL JTIME(3,7,23,0,J1)

CALL JCVT(J1)

### 3.5.11. MRKT (вариант SJ)

Назначение: планирование выполнения подпрограммы завершения на языке АССЕМБЛЕРА по истечении заданного промежутка времени.

Обращение: I=MRKT(ID,CR,MT)

Описание параметров:

ID — целое или имя переменной типа целый, определяющей идентификатор подпрограммы завершения. Значение параметра ID передается в качестве фактического параметра подпрограмме CR;

CR — имя подпрограммы завершения на языке АССЕМБЛЕРА,

которая будет выполняться по истечении заданного промежутка времени;

MT — имя переменной типа INTEGER\*4 или имя двухсловного массива типа INTEGER\*2, определяющего время во внутреннем представлении.

Значение функции:

I=0 подпрограмма выполнена;

I=1 подпрограмма не выполнена (нет свободного элемента очереди).

· ПРИМЕЧАНИЯ.

1. MRKT использует элемент очереди, что необходимо учитывать при обращении к подпрограмме IQSET (п. 3.6.6).

2. Если система занята, подпрограмма завершения может быть выполнена позднее запрошенного времени.

Пример:

```
INTEGER*2 TINT(2)
EXTERNAL ARTN
```

·  
·

```
CALL MRKT(4,ARTN,TINT)
```

### 3.5.12. SECNDS

Назначение: определение времени, прошедшего с заданного момента до текущего.

Обращение: A=SECNDS(AT)

Описание параметров:

AT — константа, выражение или имя переменной типа REAL\*4, значение которой определяет начало отсчета времени.

Значение функции: результатом выполнения подпрограммы является значение A, определяющее время в долях секунды.

Пример:

```
T1=SECNDS(0.)
```

·  
·

```
DELTA=SECNDS(T1)
```

### 3.5.13. TIMASC

Назначение: преобразование времени из внутреннего представления в часы, минуты, секунды.

Обращение: CALL TIMASC(IE,SG)

Описание параметров:

IE — время во внутреннем представлении;

SG — имя восьмисловного массива для записи значения времени в виде HH:MM:SS, где HH — часы, MM — минуты, SS — секунды.

**Пример:**

```
INTEGER*4 J1,J2,J3
LOGICAL*1 STRNG(8)
```

.

.

```
CALL JTIME(17,0,0,J1)
CALL GTIM(J2)
CALL JJCVT(J1)
CALL JJCVT(J2)
CALL JSUB(J1,J2,J3)
CALL JJCVT(J3)
```

.

.

```
CALL TIMASC(J3,STRNG)
TYPE 99,(STRNG(I),I=1,8)
```

99 FORMAT(' IT IS',8A1,'TILL 5 P .M.')

### 3.5.14. TIME

Назначение: определение текущего времени в часах, минутах, секундах.

Обращение: CALL TIME(SG)

Описание параметров:

SG — имя восьмисловного массива типа LOGICAL\*1 для записи значения времени в виде HH:MM:SS, где HH — часы, MM — минуты, SS — секунды.

**Пример:**

```
LOGICAL*1 STRNG(8)
CALL TIME(STRNG)
TYPE 99,(STRNG(I),I=1,8)
```

99 FORMAT(' IT IS NOW ',8A1)

### 3.6. Системные операции

Подпрограммы данной группы выполняют системные операции аналогично системным запросам (см. [1]).

#### 3.6.1. CHAIN

Назначение: передача управления другому фоновому заданию без вмешательства оператора.

Обращение: CALL CHAIN(D,V,W)

Описание параметров:

V — имя первой переменной последовательности передаваемых переменных;

W — целое или имя переменной типа целый, определяющей число слов (до 60), передаваемых фоновому заданию.

**Ограничения:**

1. CHAIN нельзя вызвать из подпрограммы завершения или обслуживания прерываний.

2. CHAIN не закрывает файл на логическом устройстве. При передаче управления от одного фонового задания другому необходимо закрыть файл на логическом устройстве с помощью подпрограммы CLOSE (см. [2]).

**ПРИМЕЧАНИЯ.**

1. Последовательность передаваемых переменных может быть задана как одномерный массив, блок COMMON или часть блока COMMON.

2. Последовательность передаваемых переменных занимает ячейки памяти с абсолютными адресами 510—700 в возрастающем порядке.

3. Если передаваемых переменных нет, параметр W=0.

**Примеры:**

В следующем примере передается управление программе PROG.SAV :

```
DIMENSION SPEC(2)
INTEGER*2 DATA(10)
DATA SPEC/6RDT0PRO,6RG SAV/
```

·  
·  
·

```
CALL CHAIN(SPEC,DATA,10)
```

### 3.6.2. DEVICE (режим FB)

Назначение: запись данных по указанным адресам или задание дополнительных действий, которые должен выполнить монитор после выполнения программы или выхода из нее по СУ/С.

Обращение: CALL DEVICE(IL[,LK])

Описание параметров:

IL — имя массива типа целый, содержащего список пар «адрес/данное»;

LK — целая константа, указатель связанного формата списка адресов и данных.

Список адресов и данных может иметь два формата — связанный и несвязанный. Оба формата должны оканчиваться нулевым словом. Связанный формат используется в случае неоднократного обращения к DEVICE и отличается от несвязанного формата тем, что содержит ноль в первом сло-

ве списка. Формат списка адресов и данных описан в документе [1].

Ограничения:

1. В каждый момент времени может быть использован только один список адресов и данных; если задано несколько обращений к подпрограмме DEVICE, выполняется только последнее.

2. Список адресов и данных не должен изменяться программой пользователя после вызова DEVICE и не должен размещаться в области свопинга USR.

ПРИМЕЧАНИЯ.

1. DEVICE используется, в основном, для записи в регистры устройств необходимых данных.

2. После окончания выполнения программы данные, указанные в списке, засылаются по указанным адресам.

Пример:

```
INTEGER*2 IDR11(3)
DATA IDR11(1)/«167770/
DATA IDR11(2)/0/
DATA IDR11(3)/0/
CALL DEVICE(IDR11)
```

### 3.6.3. GTJB/IGTJB

Назначение: получение информации о выполняемом задании (номер задания, нижняя и верхняя граница памяти и другая информация).

Обращение:

```
CALL GTJB(DR,[JK],[I])
I=GTJB(DR,[JK])
CALL IGTJB(DR,[JK],[I])
I=IGTJB(DR,[JK])
```

Описание параметров:

DR — имя восьми- или двенадцатисловного массива типа целый для получения информации:

DR(1) — номер задания;

для FB и XM мониторов значение DR(1) равно:

0 — фоновое задание;

2 — основное задание;

для SJ монитора значение DR(1) равно 0;

для мониторов, поддерживающих системные задания, значение DR(1) равно:

0 — фоновое задание;

2,4,6,8,10,12,14 — системные задания;

16 — основное задание;

DR(2) — адрес верхней границы памяти разделенного задания (адрес последней ячейки плюс два);  
 DR(3) — адрес нижней границы памяти разделенного задания (адрес первой ячейки);  
 DR(4) — указатель области каналов ввода-вывода;  
 DR(5) — адрес смешанной области заданий для FB и XM мониторов (0 для SJ монитора);  
 DR(6) — младший байт: номер системного терминала задания, если используется мультитерминальный переключатель, или ноль в других случаях. Старший байт не используется;  
 DR(7) — верхняя виртуальная граница для задания, созданного редактором связей по переключателю /V в режиме XM; параметр равен нулю в режиме SJ и FB или если переключатель /V не используется;  
 DR(8)—DR(9) — не используются;  
 DR(10)—DR(12) — имя задания в коде КОИ-7 (только для мониторов, поддерживающих системные задания).  
 JK — данного типа текстовый, определяющее спецификацию файла задания, о котором требуются данные.

Значение функции:

I=0 подпрограмма выполнена;

I=-1 указанное задание в текущий момент не выполняется.

Информация о задании записывается в восьмисловный массив, если в обращении к подпрограмме задан один параметр.

Например,

```
INTEGER IJPARM(8)
CALL GTJB(IJPARM)
I=GTJB(IJPARM)
```

Для записи информации в двенадцатисловный массив в обращении к подпрограмме необходимо после первого параметра указать запятую или задать все параметры. Например,

```
INTEGER IJPARM(12)
CALL GTJB(IJPARM,)
I=GTJB(IJPARM,)
```

Пример:

```
DIMENSION JDATA(12)
I=GTJB(JDATA,16)
IF (I.EQ.0) GOTO 20
TYPE 10
10 FORMAT('NO FG JOB!')
STOP
```

### 3.6.4. IDSTAT

Назначение: получение информации о состоянии устройства.

Обращение: I=IDSTAT(DN,US)

Описание параметров:

DN — имя переменной типа REAL\*4, определяющей имя устройства в коде RADIX-50;

US — имя четырехсловного массива типа целый для приема информации о состоянии заданного устройства;

US(1) — слово состояния устройства;

US(2) — длина драйвера устройства в байтах;

US(3) — адрес загрузки драйвера;

US(4) — объем тома устройства: в блоках по 256 слов — для устройств с произвольным доступом к данным, 0 — для устройств с последовательным доступом.

Значение функции:

I=0 подпрограмма выполнена;

I=1 устройство не описано в таблицах монитора.

Ограничение: IDSTAT не может быть вызвана из подпрограммы завершения или обслуживания прерываний.

**ПРИМЕЧАНИЯ.**

1. Формат слова состояния устройства приведен в документе [1].

2. US(3)=0 означает, что драйвер нерезидентен.

**Пример:**

В данном примере определяется наличие в оперативной памяти драйвера построчно-печатающего устройства. Если его нет в памяти, программа печатает сообщение:

```
INTEGER IDNAM
```

```
INTEGER*2 CBLK(4)
```

```
DATA IDNAM/3RLP /
```

```
DATA CBLK/4*0/
```

```
CALL IDSTAT(IDNAM,CBLK)
```

```
IF (CBLK(3).EQ.0) STOP 'LOAD THE LP HANDLER AND  
RERUN'
```

### 3.6.5. IFETCH

Назначение: загрузка в оперативную память драйвера устройства.

Обращение: I=IFETCH(DN)

Описание параметров:

DN — имя переменной типа REAL\*4, определяющей имя устройства в коде RADIX-50, драйвер которого следует загрузить.

Значение функции:

- I=0 подпрограмма выполнена;
- I=1 недопустимое имя устройства;
- I=2 недостаточно оперативной памяти;
- I=3 отсутствует драйвер указанного устройства.

Ограничения: IFETCH не может быть вызвана из подпрограммы завершения или обслуживания прерываний.

**ПРИМЕЧАНИЕ.** В качестве параметра DN может использоваться соответствующий элемент массива F подпрограммы ICSI (см. п. 3.4.2).

**Пример:**

В данном примере загружается драйвер устройства MX:  
REAL\*4 IDNAM  
DATA IDNAM/3RMX /  
IF (IFETCH(IDNAM).NE.0) STOP'FATAL ERROR FETCHING  
+HANDLER'

### 3.6.6. IQSET

Назначение: увеличение числа элементов очереди ввода-вывода.

Обращение: I=IQSET(GL)

Описание параметров:

GL — целое или имя переменной типа целый, определяющей число элементов, добавляемых к очереди.

Значение функции:

- I=0 подпрограмма выполнена;
- I=1 недостаточно оперативной памяти для размещения добавляемых элементов очереди.

Все операции ввода-вывода в системе ФОДОС-2 выполняются через централизованную систему управления очередью. Число элементов очереди должно быть на единицу больше общего числа запросов ввода-вывода и таймера, которые будут выполняться одновременно. Если для запросов на ввод-вывод нет свободных элементов очереди, то в режиме SJ выполнение программы приостанавливается до их появления. В режиме FB в то время, как подпрограмма ожидает элемент очереди, выполняется другое задание.

Ограничения: IQSET не может быть вызвана из подпрограммы завершения или обслуживания прерываний.

**ПРИМЕЧАНИЯ.**

1. Если IQSET используется в основном задании, необходимо назначить дополнительную память для элементов очереди (см. п. 1.2.4).

2. Если выполняется ввод-вывод по подпрограммам IREADW/IWRITW (п.п. 3.2.4.4, 3.2.9.4), то нет необходимости

в увеличении числа элементов очереди. В этом случае определены четыре элемента очереди.

3. Список системных подпрограмм, использующих элемент очереди, приведен в приложении 2.

**Пример:**

```
IF (IQSET(5).NE.0) STOP'NOTE NOUGH FREE FOR'  
+QUEUEELEMENTS'
```

### 3.6.7. ISPFN/ISPFNC/ISPFNF/ISPFNW

Подпрограммы ISPFN, ISPFNC, ISPFNF, ISPFNW используются драйверами устройств (накопитель на гибких дисках, накопитель на магнитной ленте, накопитель на магнитной ленте кассетного типа) для выполнения специальных операций.

При использовании программ данной группы необходимо, чтобы драйвер соответствующего устройства находился в памяти и канал был связан с файлом по подпрограмме LOOKUP (см. п. 3.1.7).

Данные подпрограммы используют элемент очереди; это необходимо учитывать при обращении к IQSET.

#### 3.6.7.1. ISPFN

Назначение: выполнение специальной операции в синхронном режиме ввода-вывода.

Обращение: I=ISPFN(CD,C[,W,BF,B])

Описание параметров:

CD — целое или имя переменной типа целый, определяющей код специальной операции (табл. 4);

W — целое или имя переменной типа целый, определяющей число слов, занятых в операциях ввода-вывода (чтение, запись — для гибких дисков, запись с расширенным межзонным промежутком для магнитной ленты). По умолчанию W=0. При записи на магнитную ленту параметр W=0 указывает число записей, на которое осуществляется пропуск вперед или назад: при пропуске назад осуществляется продвижение ленты до маркера ленты или в точку загрузки (BOT); при пропуске вперед осуществляется перемотка до маркера ленты или до метки EOT.

BF — имя массива типа целый, используемого как буфер данных. По умолчанию BF=0.

Значение функции:

I=0 подпрограмма выполнена;

I=1 попытка чтения или записи после признака конца файла;

I=2 аппаратная ошибка на канале;

I=3 канал закрыт.

Таблица 4

## КОДЫ СПЕЦИАЛЬНЫХ ОПЕРАЦИЙ

Операция	Код операции					
	MT	CT	DX	DY	LD	MU
1	2					
Чтение	371		377	377		
Запись	370		376	376		
Запись со стиранием данных			375	375		
Пропуск вперед к последнему файлу		377				
Пропуск вперед к последнему блоку		376				
Пропуск вперед к следующему файлу		375				
Пропуск вперед к следующему блоку		374				
Перемотка	373	373				
Запись интервала между файлами		372				
Запись метки «EOF»	377					
Пропуск вперед	376					
Пропуск назад	375					
Запись с расширенным межзонным промежутком	374					
Автономная перемотка	372					
Получение размера тома				373	373	373
Чтение/запись таблицы трансляции					372	

## ПРИМЕЧАНИЯ:

1. В табл. 4 использованы следующие обозначения устройств: MT — накопитель на магнитной ленте, CT — накопитель на магнитной ленте кассетного типа, DX — накопитель на гибком диске, DY, MU — накопители на гибком диске с двойной плотностью, LD — драйвер логического диска.

2. Вызов подпрограмм для других устройств игнорируется.

## Пример:

```
INTEGER*2 ERRADR,ERRBLK(4)
DATA ERRBLK/0,0,0,0/
```

```
ERRADR = IADDR(ERRBLK)
```

ICODE=ISPFNC(CODE,ICHAN,WDCT,BUF,ERRADR)

### 3.6.7.2. ISPFNC

Назначение: выполнение специальной операции в асинхронном режиме с использованием подпрограммы завершения на языке АССЕМБЛЕР.

Обращение: I=ISPFNC(CD,C,W,BF,B,CR)

Описание параметров: CD,W,BF — то же, что в п. 3.6.7.1.

Значение функции: то же, что в п. 3.6.7.1.

#### Пример:

```
INTEGER*2 ERRADR,ERRBLK(4)
DATA ERRBLK(0,0,0,0)
```

ERRADR=IADDR(ERRBLK)

ICODE=ISPENC(CODE,ICHAN,WDCT,BUF,ERRADR)

### 3.6.7.3. ISPFNF

Назначение: выполнение специальной операции в асинхронном режиме ввода-вывода с использованием подпрограммы завершения на языке ФОРТРАН.

Обращение: I=ISPFNF(CD,C,W,BF,B,AR,CR)

Описание параметров: CD,W,BF — то же, что в п. 3.6.7.1.

Значение функции: то же, что в п. 3.6.7.1.

#### Примеры:

```
1) INTEGER*2 ERRADR ERRBLK(4)
   DATA ERRBLK/0,0,0,0/
```

ERRADR=IADDR(ERRBLK)

ICODE=ISPFNF(CODE,ICHAN,WDCT,BUF,ERRADR)

```
2) REAL*4 MTNAME(2),AREA(2)
   DATA MTNAME/3RMT0,0./
   EXTERNAL DONSUB
```

I=IGETC()

CALL IFETCH(MTNAME)

CALL LOOKUP(I,MTNAME)

IERR=ISPFNF(«373,I,0,0,0,AREA,DONSUB)

END  
SUBROUTINE DONSUB

END

#### 3.6.7.4. ISPFNW

Назначение: выполнение специальной операции в модифицированном синхронном режиме.

Обращение: I=ISPFNW (CD,C,[W,BF,B])

Описание параметров: CD,W,BF — то же, что в п. 3.6.7.1.

Значение функции: то же, что в п. 3.6.7.1.

**Примеры:**

- 1) INTEGER\*2 ERRADR,ERRBLK(4)  
DATA ERRBLK/0,0,0,0/

ERRADR=IADDR(ERRBLK)

ICODE=ISPFNF(CODE,ICHAN,WDCT,BUF,ERRADR)

- 2) INTEGER\*2 BUF(65),TRACK,SECTOR,DBLK(4)  
DATA DBLK/3RMX0,0,0,0,/

ICHAN=IGETC( )

IF(ICHAN.LT.0) STOP 'NO CHANNEL AVAILABLE'

IF(LOOKUP(ICHAN,DBLK).LT.0) STOP 'BAD LOOKUP'

ICODE=ISPFNW(«377,ICHAN,TRACK,BUF,SECTOR)

#### 3.6.8. ITLOCK (режим FB/XM)

Назначение: блокирование USR в памяти, если она не используется другим заданием.

Обращение: I=ITLOCK( )

Значение функции:

I=0 подпрограмма выполнена;

I=1 USR используется другим заданием.

Ограничения: ITLOCK не может быть вызвана из подпрограммы завершения или обслуживания прерываний.

**ПРИМЕЧАНИЕ.** Отличие ITLOCK от LOCK состоит в том, что, если USR используется другим заданием, в случае ITLOCK USR не блокируется в памяти и управление возвра-

щается запрашивающему заданию, в случае LOCK ожидается освобождение USR и осуществляется блокирование USR в памяти.

**Пример:**

```
IF (ITLOCK().NE.0) GOTO 100
```

**3.6.9. LOCK**

Назначение: блокирование USR в памяти.

Обращение: CALL LOCK

Подпрограмма LOCK блокирует USR в памяти во время выполнения ряда операций. Программа пользователя, которая часто обращается к USR, может блокировать USR в памяти по LOCK, выполнить все необходимые действия и затем освободить USR по UNLOCK, при этом не тратится время на ненужный свопинг USR. В режиме FB вызов интерпретатора командной строки (CSI) может выполнить действие аналогичное UNLOCK. В режиме FB LOCK запрещает другому заданию использовать USR. Таким образом, USR может быть заблокирована на столько, на сколько это необходимо.

Высокоприоритетные задания выполняют LOCK в тех случаях, когда им требуется USR. При этом USR не будет доступна для других заданий в течение значительного промежутка времени. USR повторно не загружается и не может быть назначена для совместного использования с несколькими заданиями. В указанное время только одно задание может использовать USR, а все другие задания, сделавшие запрос USR, должны стать в очередь. Эти условия необходимо учитывать для систем, требующих одновременного использования высокоприоритетного и фоновых заданий, а также при работе с магнитными лентами MT и CT.

**Ограничения:**

1. Программа, вызывающая LOCK, не должна находиться в области свопинга USR. В противном случае после выполнения LOCK управление будет передано не программе пользователя, а USR. Если USR заблокирована в памяти, переменные и массивы, находящиеся в области свопирования USR, не должны использоваться.

2. После выполнения LOCK не рекомендуется использовать область памяти, которую занимает USR, даже в том случае, если не предполагается дальнейшее использование USR. Это может вызвать сбой при выполнении UNLOCK.

3. LOCK не может быть вызвана из подпрограммы завершения или обслуживания прерываний.

**ПРИМЕЧАНИЯ.**

1. Если при выполнении задания возникают условия, тре-

бующие свопинга **USR**, то программа пользователя (та ее часть, которую **USR** перекрывает в памяти) записывается на системное устройство и на ее место загружается **USR**. Если **USR** резидентна, свопинга не происходит.

2. **LOCK** всегда загружает **USR** в память, если **USR** не резидентна.

3. Подпрограммы **LOCK/UNLOCK** должны быть согласованы по числу обращений. Лишние **UNLOCK** игнорируются.

4. Если была подана команда **SET USR NOSWAP**, подпрограммы **LOCK** и **UNLOCK** не вызывают свопинга **USR**, но в режиме **FB** подпрограмма **LOCK** запрещает другому заданию использовать **USR**, а **UNLOCK** разрешает другому заданию доступ к **USR**.

### **3.6.10. RCHAIN**

Назначение: получение доступа к переменным, переданным по **CHAIN**. Обращение: **CALL RCHAIN(FL,V,W)**

Описание параметров:

**FL** — имя переменной типа целый, значение которой устанавливается подпрограммой **RCHAIN**:

**V** — имя первой переменной последовательности переданных переменных (см. п. 3.6.1);

—1 — программе передано управление;

0 — программе не передано управление;

**W** — число слов, переданных вызываемой программе.

Ограничения: **RCHAIN** не может быть вызвана из подпрограммы завершения или обслуживания прерываний.

**ПРИМЕЧАНИЕ.** Оператор **CALL**, по которому осуществляется обращение к **RCHAIN**, должен быть первым выполняемым оператором **ФОРТРАН**-программы.

**Пример:**

```
INTEGER*2 PARM5(50)
CALL RCHAIN(IFLAG, PARM5, 50)
IF (IFLAG) GOTO 10
```

·  
·  
·

### **3.6.11. RCTRL0**

Назначение: отмена команды **CV/D** (разрешение вывода на терминал). Обращение: **CALL RCTRL0**

**ПРИМЕЧАНИЕ.** Вывод на терминал осуществляется до тех пор, пока будет подана другая команда **CV/D**.

**Пример:**

```
CALL RCTRL0
CALL PRINT('PRINT UNTIL ANOTHER CV/D TYPED')
```

### 3.6.12. RESUME (режим FB)

Назначение: продолжение выполнения задания, приостановленного по SUSPND.

Обращение: CALL RESUME

ПРИМЕЧАНИЯ.

1. Подпрограмма RESUME используется обычно с подпрограммой SUSPND (п. 3.6.15).

2. Обращение к RESUME осуществляется обычно из подпрограммы завершения.

Пример использования RESUME приведен в п. 3.6.15.

### 3.6.13. SCCA

Назначение: запрещение выполнения команды СУ/С при вызове подпрограммы без параметра и определение подачи команды СУ/С при вызове подпрограммы с параметром.

Обращение: CALL SCCA[IG]

Описание параметров:

IG — имя переменной типа целый, значение которой устанавливается по подпрограмме; IG не равно нулю, если с терминала последовательно поданы две команды СУ/С.

ПРИМЕЧАНИЯ.

1. Подпрограмма пользователя должна проверять и очищать IG перед вызовом SCCA.

2. Подпрограмма SCCA без параметра запрещает выполнение команды СУ/С и помещает код команды в буфер, откуда она может быть считана как символ.

Пример:

```
PROGRAM SCCA
CALL PRINT('PROGRAM HAS STARTED,TYPE')
IFLAG=0
CALL SCCA(IFLAG)
10 I=ITTINR()
CALL PRINT('A СУ/С WAS TYPED')
IF (IFLAG.EQ.0) GOTO 10
.
.
CALL PRINT('A DOUBLE СУ/С WAS TYPED')
TYPE 19,IFLAG
19 FORMAT('IFLAG=','06,/')
CALL SCCA
CALL PRINT('TYPE A СУ/С TO EXIT')
20 GOTO 20
END
```

### 3.6.14. SETCMD

Назначение: передача командной строки клавиатурному монитору.

Обращение: CALL SETCMD(KS)

Описание параметров:

KS — текстовая константа (без ограничивающих символов <BK> и <PC>) или переменная типа текстовый, определяющая строку для передачи клавиатурному монитору.

Командная строка будет выполнена после завершения программы пользователя.

**ПРИМЕЧАНИЯ.**

1. Вызов подпрограммы SETCMD должен осуществляться до завершения программы пользователя. Командная строка должна быть короткой во избежание размещения ее в области стека. Если необходимо выполнение нескольких командных строк, следует использовать косвенный файл.

2. При использовании SETCMD

— запрещены команды клавиатурного монитора REENTER, START, CLOSE

— перед выполнением команды COMPILE необходимо запретить свопинг USR.

3. Командная строка передается в область связей. (500—777(8)) и записывается, начиная с ячейки 512.

**Пример:**

```
LOGICAL *1 INPUT(134),PROMPT(8)
DATA PROMPT/'P','R','O','M','P','T','>',«200/
CALL GTLIN(INPUT,PROMPT)
CALL SETCMD(INPUT)
END
```

### 3.6.15. SUSPND (режим FB)

Назначение: задержка выполнения задания и разрешение выполнения подпрограммы завершения.

Обращение: CALL SUSPND

**ПРИМЕЧАНИЯ.**

1. Монитор имеет счетчик задержки для каждого задания. Этот счетчик уменьшается подпрограммой SUSPND и наращивается подпрограммой RESUME (см. п. 3.6.12). Выполнение задания будет приостановлено лишь в том случае, если счетчик содержит отрицательное значение. Если вызывается RESUME перед SUSPND, подпрограмма SUSPND не выполняется.

2. Подпрограммы SUSPND и RESUME должны быть согласованы по числу обращений.

3. Вызов подпрограммы RESUME из программы пользователя или подпрограммы завершения увеличивает счетчик задержки.

4. Вызов подпрограммы SUSPND из подпрограммы завершения уменьшает счетчик задержки, но не приостанавливает выполнение задания. Задание продолжает выполняться до тех пор, пока оно само не вызовет SUSPND. В этом случае для продолжения выполнения задания требуется два вызова RESUME.

5. RESUME, вызванная из подпрограммы завершения и не согласованная с ранее выполненным вызовом SUSPND, может привести к продолжению выполнения задания после запланированного останова до того, как истечет весь промежуток времени, заданный по ITWAIT (см. п. 3.2.8).

**Пример:**

```
INTEGER IAREA(4)
COMMON /RDBLK/ IBUF(256)
EXTERNAL RDFIN
.
.
IF (IREADF(256,IBUF,IBLK,ICHAN,IAREA,RDFIN).NE.0)
+GOTO 100
.
.
CALL SUSPND
.
.
END
SUBROUTINE RDFIN(IARG1,IARG2)
COMMON/RDBLK/ IBUF(256)
.
.
CALL RESUME
.
.
END
```

### 3.6.16. UNLOCK

Назначение: освобождение USR из памяти и загрузка программы пользователя, если при выполнении подпрограммы LOCK выполнялся свопинг USR.

Обращение: CALL UNLOCK

**ПРИМЕЧАНИЯ.**

1. Подпрограммы UNLOCK и LOCK должны быть согласованы по числу обращений. Если обращений к LOCK боль-

ше чем к UNLOCK, то USR остается блокированной в памяти. Лишние UNLOCK игнорируются.

2. При выполнении двух заданий в режиме FB подпрограммы LOCK/UNLOCK рекомендуется использовать только в необходимых случаях. Если одно задание блокирует USR в памяти (LOCK), другое задание не может использовать USR до тех пор, пока USR не будет освобождена (UNLOCK).

3. В режиме FB вызов CSI (ICSI) выполняет действие, аналогичное UNLOCK.

4. В режиме XM USR всегда резидентна.

**Пример:**

```
CALL LOCK
```

```
·  
·
```

```
CALL UNLOCK
```

### 3.7. Операции с данными INTEGER\*4

Библиотека системных подпрограмм содержит подпрограммы преобразования данных типа REAL\*4, REAL\*8, INTEGER\*2, INTEGER\*4 и выполнения арифметических операций с данными INTEGER\*4.

Данное типа INTEGER\*4 занимает два слова памяти. Первое слово (ячейка с младшим адресом) содержит младшие разряды, второе слово (ячейка со старшим адресом) содержит знак и старшие разряды. Диапазон представления данных INTEGER\*4 от  $-2^{**31}+1$  до  $2^{**31}-1$  (знаком \*\* обозначена операция возведения в степень). Данные INTEGER\*4 могут задаваться с помощью оператора DATA с указанием младшей и старшей части числа.

**Пример:**

```
INTEGER*4 J
```

```
INTEGER*2 I(2)
```

```
EQVIVALENCE (J, I) :NF
```

```
DATA I/3,0/ ! присваивает J значение 3
```

При задании отрицательного числа старшая часть числа (ячейка со старшим адресом) должна быть продолжением младшей части в дополнительном коде.

**Пример:**

```
INTEGER*4 J
```

```
INTEGER*2 I(2)
```

```
EQVIVALENCE (J, I)
```

```
DATA I/-4, -1/ ! присваивает I значение - 4
```

Данные параметры INTEGER\*4 могут быть заданы следующим образом:

INTEGER\*2 J(2)

DATA J/3,0/

Для обозначения типа параметра в качестве первого символа параметра используются следующие буквы:

I — для параметра типа INTEGER\*2;

J — для параметра типа INTEGER\*4;

A — для параметра типа REAL\*4;

D — для параметра типа REAL\*8.

### 3.7.1. AJFLT

Назначение: преобразование данного типа INTEGER\*4 в REAL\*4 и присвоение результата функции.

Обращение: A=AJFLT(JS)

Описание параметров:

JS — имя преобразуемой переменной или имя элемента массива типа INTEGER\*4.

Значение функции: данное типа REAL\*4.

**Пример:**

REAL\*4 VALUE,AJFLT

INTEGER\*4 JVAL

VALUE=AJFLT(JVAL)\*3.5

### 3.7.2. DJFLT

Назначение: преобразование данного типа INTEGER\*4 в REAL\*8 и присвоение результата функции.

Обращение: D=DJFLT(JS)

Описание параметров:

JS — имя преобразуемой переменной или имя элемента массива типа INTEGER\*4.

Значение функции: данное типа REAL\*8.

**ПРИМЕЧАНИЕ.** Если используется подпрограмма DJLT, то тип функции DJFLT должен быть определен в программе явно (REAL\*8 DJFLT) или неявно (IMPLICIT REAL\*8(D)). По умолчанию предполагается тип REAL\*4.

**Пример:**

INTEGER\*4 JVAL

REAL\*8 DJFLT,D

D=DJFLT(JVAL)

### 3.7.3. IAJFLT

Назначение: преобразование данного типа INTEGER\*4 в REAL\*4.

Обращение: I=IAJFLT(JS,AR)

Описание параметров:

JS — имя преобразуемой переменной или имя элемента массива типа INTEGER\*4;

AR — имя переменной REAL\*4 или имя элемента массива для приема результата преобразования.

Значение функции:

I=-2 потеря значащих цифр во время преобразования;

I=-1 результат меньше 0;

I=0 результат равен 0;

I=1 результат больше 0.

Пример:

```
INTEGER*4 JVAL
REAL*4 RESULT
```

·  
·

```
IF (IAJFLT(JVAL,RESULT).EQ.-2) TYPE 99
99 FORMAT ('OVERFLOW IN INTEGER*4 TO REAL CONVERSION')
```

### 3.7.4. IDJFLT

Назначение: преобразование данного типа INTEGER\*4 в REAL\*8.

Обращение: I=IDJFLT(JS,DE)

Описание параметров:

JS — имя преобразуемой переменной или имя элемента массива типа INTEGER\*4;

DE — имя переменной или имя элемента массива типа REAL\*8 для приема результата преобразования.

Значение функции:

I=-1 результат меньше 0;

I=0 результат равен 0;

I=1 результат больше 0.

Пример:

```
INTEGER*4 JJ
REAL*8 DJ
```

·  
·

```
IF (IDJFLT(JJ,DJ).LE.0) TYPE 99
99 FORMAT ('VALUE IS NOT POSITIVE')
```

### 3.7.5. IJCVT

Назначение: преобразование данного типа INTEGER\*4 в INTEGER\*2.

Обращение: I=IJCVT(JS[,IS])

Описание параметров:

JS — имя преобразуемой переменной или имя элемента массива типа INTEGER\*4;  
IS — имя переменной INTEGER\*2 для приема результата преобразования.

Значение функции:

I = -2 переполнение;  
I = -1 результат меньше 0;  
I = 0 результат равен 0;  
I = 1 результат больше 0.

**ПРИМЕЧАНИЕ.** Если параметр IS не используется, то результат преобразования присваивается JS.

**Пример:**

```
INTEGER*4 JVAL  
INTEGER*2 IVAL
```

```
IF (IJCVT(JVAL,IVAL).EQ.-2) TYPE 99  
99 FORMAT ('NUMBER TOO LARGE IN IJCVT CONVERSION')
```

### 3.7.6. JADD

Назначение: сложение данных типа INTEGER\*4.

Обращение: I = JADD(J1, J2, JR)

Описание параметров:

J1 — имя переменной или имя элемента массива типа INTEGER\*4, первое слагаемое;

J2 — имя переменной или имя элемента массива типа INTEGER\*4, второе слагаемое;

JR — имя переменной или имя элемента массива типа INTEGER\*4 для приема результата.

Значение функции:

I = -2 переполнение;  
I = -1 результат меньше 0;  
I = 0 результат равен 0;  
I = 1 результат больше 0.

**Пример:**

```
INTEGER*4 JOP1, JOP2, JRES
```

```
IF (JADD(JOP1, JOP2, JRES).EQ.-2) GO TO 100
```

### 3.7.7. JAFIX

Назначение: преобразование данного типа REAL\*4 в INTEGER\*4.

Обращение: I = JAFIX(AS, JR)

Описание параметров:

AS — константа, выражение или имя преобразуемой переменной REAL\*4;  
JR — имя переменной или имя элемента массива типа INTEGER\*4 для приема результата преобразования.

Значение функции:

I=-2 переполнение;  
I=-1 результат меньше 0;  
I=0 результат равен 0;  
I=1 результат больше 0.

**Пример:**

```
INTEGER*4 JOP1
ACCEPT 99,A
99 FORMAT (F15.0)
IF (JAFIX(A,JOP1).EQ.-2) GO TO 100
```

### 3.7.8. JCMP

Назначение: сравнение данных типа INTEGER\*4.

Обращение: I=JCMP(J1,J2)

Описание параметров:

J1 — имя переменной INTEGER\*4 или имя элемента массива, первый операнд;

J2 — имя переменной INTEGER\*4 или имя элемента массива, второй операнд.

Значение функции:

I=-1 J1 < J2;  
I=0 J1 = J2;  
I=1 J1 > J2.

**ПРИМЕЧАНИЕ.** Функция имеет тип INTEGER\*2.

**Пример:**

```
INTEGER*4 JOPX,JOPY
IF (JCMP(JOPX,JOPY)) 10,20,30
```

### 3.7.9. JDFIX

Назначение: преобразование данного типа REAL\*8 в INTEGER\*4.

Обращение: I=JDFIX(DS,JR)

Описание параметров:

DS — константа, выражение или имя переменной REAL\*8;  
JR — имя переменной или имя элемента массива типа INTEGER\*4 для приема результата преобразования.

**Значение функции:**

- I = -2 переполнение;
- I = -1 результат меньше 0;
- I = 0 результат равен 0;
- I = 1 результат больше 0.

**Пример:**

```
INTEGER*4 JNUM
REAL*8 DPNUM
.
.
20 TYPE 98
98 FORMAT ('ENTER POSITIVE INTEGER')
ACCEPT 99,DPNUM
99 FORMAT (F20.0)
IF (JDFIX(DPNUM,JNUM).LT.0) GOTO 20
.
.
```

### 3.7.10. JDIV

**Назначение:** деление данных типа INTEGER\*4.

**Обращение:** I = JDIV (J1,J2,JR[,JO])

**Описание параметров:**

J1 — имя переменной или имя элемента массива типа INTEGER\*4, делимое;

J2 — имя переменной или имя элемента массива типа INTEGER\*4, делитель;

JR — имя переменной или имя элемента массива типа INTEGER\*4, частное;

JO — имя переменной INTEGER\*4, остаток со знаком переменной J1.

**Значение функции:**

- I = -3 деление на ноль;
- I = -1 частное меньше 0;
- I = 0 частное равно 0;
- I = 1 частное больше 0.

**Пример:**

```
INTEGER*4 JN1,JN2,JQUO
.
.
CALL JDIV (JN1,JN2,JQUO)
```

### 3.7.11. JICVT

Назначение: преобразование данного типа INTEGER\*2 в INTEGER\*4.

Обращение: I=JICVT(JS,JR)

Описание параметров:

IS — имя преобразуемой переменной или имя элемента массива типа INTEGER\*2;

JR — имя переменной INTEGER\*4 или имя элемента массива для приема результата преобразования.

Значение функции:

I=-1 результат меньше 0;

I=0 результат равен 0;

I=1 результат больше 0.

Пример:

```
INTEGER*4 JVAL  
CALL JICVT(478,JVAL)
```

### 3.7.12. JJCVT

Назначение: преобразование времени из внутреннего представления в данное INTEGER\*4 или наоборот.

Обращение: CALL JJCVT(JS)

Описание параметров:

JS — имя преобразуемой переменной во внутреннем представлении или INTEGER\*4.

### ПРИМЕЧАНИЯ.

1. Действие данной подпрограммы заключается в перестановке слов ячеек памяти: содержимое ячейки со старшим адресом записывается по адресу ячейки с младшим адресом и наоборот.

2. Запись времени во внутреннем представлении приведена в п. 3.5.1.

Пример:

```
INTEGER*4 TIME
```

:

:

```
CALL GTIM(TIME)  
CALL JJCVT(TIME)
```

### 3.7.13. JMOV

Назначение: присваивание значения переменной INTEGER\*4.

Обращение: I=JMOV(JS,JT)

Описание параметров:

JS — имя переменной INTEGER\*4, значение которой присваивается JT;

JT — имя переменной INTEGER\*4, которой присваивается значение.

Значение функции:

I = -1 результат меньше 0;

I = 0 результат равен 0;

I = 1 результат больше 0.

**Пример:**

INTEGER\*4 INT1

·  
·  
·

IF (JMOV(INT1,INT1).NE.0) GOTO 300

### 3.7.14. JMUL

Назначение: умножение данных типа INTEGER\*4.

Обращение: I = JMUL (J1,J2,JR)

Описание параметров:

J1 — имя переменной или имя элемента массива типа INTEGER\*4, множимое;

J2 — имя переменной или имя элемента массива типа INTEGER\*4, множитель;

JR — имя переменной или имя элемента массива типа INTEGER\*4 для приема произведения.

Значение функции:

I = -2 переполнение;

I = -1 произведение меньше 0;

I = 0 произведение равно 0;

I = 1 произведение больше 0.

**Пример:**

INTEGER\*4 J1,J2,JRES

·  
·  
·

IF (JMUL(J1,J2,JRES) + 1) 100,10,20

### 3.7.15. JSUB

Назначение: вычитание данных типа INTEGER\*4.

Обращение: I = JSUB (J1,J2,JR)

Описание параметров:

J1 — имя переменной или имя элемента массива типа INTEGER\*4, уменьшаемое;

J2 — имя переменной или имя элемента массива типа INTEGER\*4, вычитаемое;

JR — имя переменной или имя элемента массива типа INTEGER\*4 для приема разности.

Значение функции:

I = -2 переполнение;  
I = -1 результат меньше 0;  
I = 0 результат равен 0;  
I = 0 результат больше 0.

**Пример:**

```
INTEGER*4 JOP1,JOP2,J3
```

.

.

```
CALL JSUB (JOP1,JOP2,J3)
```

### **3.8. Операции с текстовыми данными**

Библиотека системных подпрограмм содержит набор подпрограмм для обработки текстовых данных. Текстовое данное задается в виде массива LOGICAL\*1 или буквенно-цифрового литерала и может содержать любую последовательность символов, коды которых находятся в интервале от 001 до 127. Текстовое данное, которое имеет значение «пусто» (данное, не содержащее символов), может быть задано переменной или константой, принимающей нулевое значение, или переменной или константой логического типа, принимающей значение «ложь».

Каждый символ в текстовом данном имеет свою позицию. Позиции символов в текстовом данном нумеруются последовательно слева направо, начиная с 1.

Максимальная длина текстового данного 32767 символов.

#### **3.8.1. Размещение текстового данного в массиве**

Во время выполнения подпрограмм необходимо следить за тем, чтобы текстовое данное не выходило за границы массива, отведенного для него. Для задания длины текстового данного используется параметр LE. С параметром LE может использоваться параметр ER — переменная логического типа, которой первоначально ФОРТРАН-программой должно быть присвоено значение «ложь». Если результирующее текстовое данное усекается до длины LE, то параметру ER подпрограммой, в вызове которой он используется, присваивается значение «истина». При записи текстового данного в массив LOGICAL\*1 следует добавлять код 000 в качестве признака конца текстового данного. Системные подпрограммы это делают автоматически.

Одномерный массив типа LOGICAL\*1 может быть использован для размещения одного текстового данного, длина которого изменяется от нуля до величины, на единицу меньшей размера массива.

Массив, в который будет помещен результат, должен

содержать число  $LE+1$  элементов (один элемент предназначен для размещения признака конца текстового данного).

Для размещения одномерного массива текстовых данных может быть использован двумерный массив типа `LOGICAL*1`. Каждое текстовое данное массива должно иметь длину на единицу меньшую размера массива по первому измерению. Число текстовых данных в массиве определяется по второму измерению массива.

**Пример:**

`LOGICAL*1 W(21,10)`

В данном примере создается массив текстовых данных, который состоит из 10 текстовых данных, каждое из которых может содержать до 20 символов.

В массиве типа `LOGICAL*1` может размещаться массив текстовых данных, имеющий максимум 6 измерений (максимальное число измерений массива `LOGICAL*1` равно 7).

### 3.8.2. Передача текстовых данных

Массивы `LOGICAL*1`, содержащие текстовые данные, могут помещаться в `COMMON` и обращение к ним каких-либо или всех подпрограмм осуществляется через аналогичный `COMMON`. Необходимо иметь ввиду, что если массив `LOGICAL*1` имеет нечетное число элементов, то всем последующим переменным блока `COMMON` будут присвоены нечетные адреса.

Массив `LOGICAL*1` имеет нечетное число элементов в том случае, если произведение размеров по всем измерениям нечетно.

**Пример:**

`LOGICAL*1 B(10,7) ! (10*7) = 70` четное число элементов  
`LOGICAL*1 H(21) ! 21` нечетное число элементов

Если массивы с нечетным числом элементов размещают в `COMMON`, то их следует помещать или последними или попарно, чтобы получить четное число элементов.

**Пример:**

`COMMON A1,A2,A3(10),H(21)`

Массив нечетного размера помещен в конце `COMMON`.  
`COMMON A1,A2,H(21),H1(7),A3(10)`

Массивы нечетных размеров `H` и `H1` расположены попарно.

Эти предосторожности относятся только к переменным и массивам `LOGICAL*1`.

Второй способ передачи текстовых данных подпрограммам — использование фактических и формальных параметров подпрограмм.

Текстовое данное может быть передано путем указания имени массива, содержащего это данное в качестве фактического параметра.

**Пример:**

LOGICAL\*1 A(21) ! текстовое данное «А» содержит не более 20 символов

CALL SUBR(A) ! передать текстовое данное «А» подпрограмме SUBR

Если максимальная длина текстового данного, используемого в качестве формального параметра подпрограммы, неизвестна, или подпрограмма используется для обработки текстовых данных различной длины, то формальный параметр подпрограммы должен быть объявлен как одномерный массив LOGICAL\*1.

**Пример:**

LOGICAL\*1 ARG(1)

В этом случае системные подпрограммы обработки текстовых данных будут правильно определять длину текстового данного.

Если в подпрограмму передается многомерный массив текстовых данных, то он должен быть описан в вызываемой программе с теми же размерами по измерениям, с какими он указан в вызывающей программе.

### 3.8.3. CONCAT

Назначение: объединение двух текстовых данных.

Обращение: CALL CONCAT(V1,V2,RT[,LE[,ER]])

Описание параметров:

V1 — имя массива, содержащего первое текстовое данное;

V2 — имя массива, содержащего второе текстовое данное;

RT — имя массива, в который помещается результат;

LE — целое или имя переменной типа целый, определяющей максимальную длину результирующего текстового данного;

ER — имя логической переменной, принимающей значение «истина», если результирующее текстовое данное усекается до длины LE.

Метод:

к текстовому данному V1 присоединяется текстовое данное V2 и код 000; полученное текстовое данное записывается в RT. Если одно текстовое данное имеет значение «пусто», то в RT копируется другое текстовое данное. Если оба текстовых данных имеют значение «пусто», то RT также имеет значение «пусто». После выполнения подпрограммы старое содержимое массива RT теряется.

**Пример:**

LOGICAL\*1 IN(22),OUT(30),STR(7)

CALL CONCAT(STR,IN,OUT,29)

#### 3.8.4. GETSTR

Назначение: ввод форматной записи с указанного логического устройства в массив. Конечные пробелы удаляются и присоединяется код 000.

Обращение: CALL GETSTR(LN,RT,LE,ER)

Описание параметров:

LN — целое или имя переменной типа целый, определяющей номер логического устройства, с которого осуществляется ввод записи;

RT — имя массива, принимающего форматную запись;

LE — целое или имя переменной типа целый, определяющей максимальную длину вводимой записи;

ER — имя логической переменной, принимающей значение «истина» в случае возникновения ошибки;

ER=—1 обнаружен конец файла;

ER=—2 аппаратная ошибка;

ER=—3 запись содержит больше символов, чем определено в параметре LE.

Ограничения:

1. Подпрограмма GETSTR не должна одновременно использоваться в подпрограммах основной программы и подпрограмме завершения одного задания.

2. Если GETSTR используется в подпрограмме завершения, то она не должна быть первой операцией ввода-вывода на логическом устройстве.

**ПРИМЕЧАНИЕ.** Конечный пробел представляет собой пробел, справа от которого отсутствуют символы, отличные от пробела.

**Пример:**

LOGICAL\*1 STRING(81),ERR

CALL GETSTR(5,STRING,80,ERR)

#### 3.8.5. INDEX

Назначение: поиск символа или последовательности символов в текстовом данном.

Обращение: CALL INDEX(V1,PT[,E],MT) или  
MT=INDEX(V1,PT[,E])

Описание параметров:

V1 — имя массива, содержащего просматриваемое текстовое данное;

PT — искомый символ или последовательность символов;

E — целое или имя переменной типа целый, определяющей номер позиции, начиная с которой осуществляется поиск;

MT — целое или имя переменной типа целый, определяющей номер позиции первого появления искомого символа или последовательности символов. MT=0, если символ (последовательность символов) не найден.

**ПРИМЕЧАНИЕ.** Если параметр E не задан, поиск начинается с первого символа массива.

**Пример:**

```
CALL SCOPY('ABCDEFGHI',STRING)
```

```
CALL INDEX(STRING,'EFG',,M)
```

```
CALL INDEX(STRING,'XYZ',,N)
```

```
CALL INDEX('ABCABCABC','ABC',5,L)
```

### 3.8.6. INSERT

Назначение: замена символов одного текстового данного символами другого текстового данного.

Обращение: CALL INSERT(M1,RS,E[,MT])

Описание параметров:

M1 — имя массива, содержащего вводимое текстовое данное;  
RS — имя массива, содержащего изменяемое текстовое данное;

E — целое или имя переменной типа целый, определяющей номер позиции в RS, начиная с которой осуществляется замена символов;

MT — целое или имя переменной типа целый, определяющей максимальное число заменяемых символов.

**ПРИМЕЧАНИЯ.**

1. Если параметр MT не задан, все символы, расположенные справа от E в текстовом данном массива RS, заменяются символами текстового данного массива M1.

2. Вводимое и изменяемое текстовые данные могут находиться в одном массиве, если MT определено и не превосходит разницы значений параметра E и максимальной длины текстового данного модифицируемого массива.

**Пример:**

```
CALL SCOPY('ABCDEFGHIJ',S1)
```

```
CALL SCOPY(S1,S2)
```

```
CALL INSERT('123',S1,6,3)
```

```
CALL INSERT('123',S2,4)
```

### 3.8.7. ISCOMP/SCOMP

Назначение: сравнение двух текстовых данных:

Обращение: CALL SCOMP(V1,V2,I) или  
I=ISCOMP(V1,V2)

Описание параметров:

V1 — имя массива, содержащего первое текстовое данное;  
V2 — имя массива, содержащего второе текстовое данное;  
I — целое или имя переменной типа целый, определяющей номер позиции несовпадающих символов.

Значение функции:

I<0 подпрограмма выполнена, V1<V2;  
I=0 подпрограмма выполнена, V1=V2;  
I>0 подпрограмма выполнена, V1>V2.

Метод: текстовые данные сравнивать слева направо по-символьно. Если текстовые данные имеют неодинаковую длину, то текстовое данное, имеющее меньшую длину, дополнить пробелами. Пустое текстовое данное заменить пробелами.

Ограничения: в качестве параметров подпрограмм ISCOMP/SCOMP не может использоваться буквенно-цифровой литерал.

### 3.8.8. VERIFY/IVERIF

Назначение: определение позиции несовпадающих символов двух текстовых данных.

Обращение: CALL VERIFY(V1,V2,I) или  
I=IVERIF(V1,V2)

Описание параметров:

V1 — имя массива, содержащего первое текстовое данное;  
V2 — имя массива, содержащего второе текстовое данное;  
I — целое или имя переменной типа целый, определяющей номер позиции несовпадающих символов.

Значение функции:

I=0 текстовые данные совпадают или текстовое данное V1 является пустым;  
I=N N — номер позиции первого символа V1, не совпадающего с V2;  
I=1 текстовое данное V2 — пустое, V1 не является пустым.

Ограничения: в качестве параметров подпрограмм VERIFY/IVERIF не может использоваться буквенно-цифровой литерал.

Пример:

LOGICAL\*1 INSTR(81)

·  
·  
·

```
CALL VERIFY(INSTR,'0123456789',I)
IF(IEQ.1) STOP 'NUMBER MISSING'
IF(IEQ.0) I=LEN(INSTR)
IF(IGT.5) STOP 'TOO MANY DIGITS'
NUM=IVALUE(INSTR,I)
```

END

### 3.8.9. LEN

Назначение: определение числа символов в текстовом данном.

Обращение: I=LEN(V1)

Описание параметров:

V1 — имя массива, содержащего текстовое данное.

**ПРИМЕЧАНИЯ.**

1. В качестве параметра подпрограммы LEN не может использоваться буквенно-цифровой литерал.

2. Нулевой байт, являющийся признаком конца текстового данного, не входит в число символов текстового данного.

3. Длина пустого текстового данного — 0.

**Пример:**

```
LOGICAL*1 STRING(73)
```

```
TYPE 99,(STRING(I),I=1,LEN(STRING))
99 FORMAT('0',132A1)
```

### 3.8.10. PUTSTR

Назначение: вывод форматной записи на логическое устройство.

Обращение: CALL PUTSTR(LN,M1,CA,ER)

Описание параметров:

LN — целое или имя переменной типа целый, определяющей номер логического устройства, на которое осуществляется вывод записи;

M1 — имя массива, содержащего форматную запись;

CA — символ, добавляемый в начало записи для управления кареткой;

ER — имя логической переменной, принимающей значение «истина» в случае возникновения ошибки:

ER=—1 обнаружен конец файла;

ER=—2 аппаратная ошибка.

Ограничения:

1. Подпрограмма PUTSTR не должна использоваться одновременно в подпрограммах основной программы и подпрограмме завершения одного задания.

2. Если PUTSTR используется в подпрограмме завершения, то она не должна быть первой операцией ввода-вывода на логическом устройстве.

#### ПРИМЕЧАНИЯ.

1. Форматная запись массива M1 должна оканчиваться нулевым байтом.

2. Если CA=0, то первый символ M1 будет первым символом записи на логическом устройстве.

#### Пример:

```
LOGICAL*1 STRING(81),ERR
```

```
·  
·
```

```
CALL PUTSTR(7,STRING,'0',ERR)
```

#### 3.8.11. REPEAT

Назначение: копирование текстового данного заданное число раз.

Обращение: CALL REPEAT(M1,RT,IK[,LE[,ER]])

Описание параметров:

M1 — имя массива, содержащего копируемое текстовое данное;

RT — имя массива для приема результата;

IK — целое или имя переменной типа целый, определяющей число копирований;

LE — целое или имя переменной типа целый, определяющей максимальную длину результирующего текстового данного;

ER — имя логической переменной, принимающей значение «истина», если результирующее текстовое данное усекается до длины LE.

#### ПРИМЕЧАНИЯ.

1. Копируемое и результирующее текстовые данные могут находиться в одном массиве, если число копирований IK равно 1 или 0.

2. Если IK=1, действие REPEAT аналогично SCOPY.

3. Если IK=0, в массив RT помещается пустое текстовое данное. Старое содержимое RT теряется.

#### Пример:

```
LOGICAL*1 SIN(21),SOUT(101)
```

```
CALL REPEAT (SIN,SOUT,5)
```

#### 3.8.12. SCOPY

Назначение: копирование текстового данного.

Обращение: CALL SCOPY(M1,RS[,LE[,ER]])

M1 — имя массива, содержащего копируемое текстовое данное;

RS — имя массива для приема копируемого текстового данного (результата);

LE — целое или имя переменной типа целый, определяющей максимальную длину результирующего текстового данного;

ER — имя логической переменной, принимающей значение «истина», если текстовое данное усекается до длины LE.

#### ПРИМЕЧАНИЯ:

1. Копирование текстового данного заканчивается при появлении признака конца текстового данного или достижении заданной длины текстового данного.

2. Копируемое и результирующее текстовые данные могут находиться в одном массиве.

3. После выполнения подпрограммы старое содержимое массива RS теряется.

#### Пример:

```
LOGICAL*1 STRING(80)
```

```
CALL SCOPY('THIS IS THE INITIAL VALUE',STRING)
```

#### 3.8.13. STRPAD

Назначение: дополнение текстового данного пробелами справа до заданной длины.

Обращение: CALL STRPAD(V1,IR[,ER])

Описание параметров:

V1 — имя массива, содержащего текстовое данное, которое будет дополняться пробелами;

IR — целое или имя переменной типа целый, определяющей длину результирующего текстового данного;

ER — имя логической переменной, принимающей значение «истина», если длина исходного текстового данного превышает значение IR.

ПРИМЕЧАНИЕ. Если длина исходного текстового данного больше или равна заданной, дополнение текстового данного пробелами не производится.

#### Пример:

```
LOGICAL*1 STR(81)
```

```
·  
·  
·
```

```
CALL STRAD(STR,80)
```

```
PRINT 100,(STR(I),I=1,80)
```

```
100 FORMAT(80A1)
```

### 3.8.14. SUBSTR

Назначение: копирование текстового данного с указанного номера позиции.

Обращение: CALL SUBSTR(M1,RT,IP[,LE])

Описание параметров:

M1 — имя массива, содержащего копируемое текстовое данное;

RT — имя массива для приема копируемого текстового данного (результата);

IP — целое или имя переменной типа целый, определяющей номер позиции в массиве M1, начиная с которого осуществляется копирование текстового данного.

**ПРИМЕЧАНИЯ.**

1. Если параметр LE не задан, результирующее текстовое данное будет содержать все символы копируемого текстового данного, начиная с позиции IP.

2. Если LE=0, в массив RT записывается текстовое данное.

3. После выполнения подпрограммы старое содержимое RT теряется.

### 3.8.15. TRANSL

Назначение: модификация текстового данного путем замены и перестановки символов.

Обращение: CALL TRANSL(M1,RT,SL[,LP])

Описание параметров:

M1 — имя массива, содержащего исходное текстовое данное;

RT — имя массива (для приема модифицированного текстового данного);

SL — имя массива, содержащего символы для модификации;

LP — имя массива, содержащего символы, которые должны быть модифицированы.

В массив RT помещается массив M1, модифицированный символами SL. Если массив M1 содержит символ, указанный в LP, то в массиве RT он заменяется соответствующим символом из SL. Если массив LP не указан, то предполагается, что он состоит из 127 символов КОИ-7, коды которых расположены в возрастающем порядке, начиная с 001. Если текстовые данные массивов SL и LP отличаются по числу символов, то длинное текстовое данное усекается до короткого.

**ПРИМЕЧАНИЕ.** Для выполнения подпрограммы TRANSL требуется область стека приблизительно из 64 слов. Это следует учитывать при работе с данной подпрограммой.

**Примеры:**

1. CALL TRANSL(GA,GB,'—!','?')

GA копируется в GB, причем все точки в GA заменяются на

минусы, а знаки «?» на «!».

2. В данном примере осуществляется перестановка символов текстового данного

'THE HORN BLOWS AT MIDNIGHT'

в следующем порядке:

```
' AT MIDNIGHT THE HORN BLOWS'  
LOGICAL*1 STRING(27),RESULT(27),PATRN(27)  
CALL SCOPY('THE HORN BLOWS AT MIDNIGHT',STRING)  
DO 10 I=16,26  
10  RATR(1-15)=I  
   RATR(12)=15  
   DO 20 I=1,14  
20  RATR(1+12)=I  
   RATR(27)=0  
   CALL TRANSL(PATR,RESULT,STRING)
```

### 3.8.16. TRIM

Назначение: удаление конечных пробелов из текстового данного.

Обращение: CALL TRIM(V1)

Описание параметров:

V1 — имя массива, содержащего исходное текстовое данное.

**ПРИМЕЧАНИЯ.**

1. Если все символы текстового данного являются пробелами, то текстовое данное заменяется пустым.

2. Если текстовое данное не имеет конечных пробелов, то оно не изменяется.

**Пример:**

```
LOGICAL*1 STRING(81)  
'ACCEPT 100,(STRING(I),I=1,80)  
100  FORMAT(80A1)  
CALL SCOPY(STRING,STRING,80)  
CALL TRIM(STRING)
```

### 3.9. Преобразования RADIX-50

#### 3.9.1. IRAD50

Назначение: упаковка символов КОИ-7 в RADIX-50 и подсчет числа упакованных символов.

Обращение: N=IRAD50(G,BC,P)

Описание параметров:

G — константа, определяющая число упаковываемых символов КОИ-7;

BC — буквенно-цифровой литерал или имя массива, содержащего упаковываемые символы КОИ-7;

P — имя массива для приема результата.

Значение функции: N — число упакованных символов КОИ-7.

Метод: три символа КОИ-7 упаковываются в одно слово

**RADIX-50.** Число образуемых слов RADIX-50 определяется выражением  $(G+2)/3$ , имеющим целое значение.

Выполнение подпрограммы прекращается при появлении символа, не имеющего эквивалента RADIX-50, или после упаковки указанного числа символов КОИ-7.

**Пример:**

```
REAL*8 FSPEC  
CALL IRAD50(12,'SYOTEMP DAT',FSPEC)
```

### 3.9.2. R50ASC

Назначение: распаковка символов RADIX-50 в КОИ-7.

Обращение: CALL R50ASC(G,BC,P)

Описание параметров:

G — константа, определяющая число получаемых символов КОИ-7;

BC — имя массива, содержащего распаковываемые символы RADIX-50 ( $(G+2)/3$  слова);

P — имя массива для приема результата.

**ПРИМЕЧАНИЕ.** Символы RADIX-50, не имеющие эквивалента КОИ-7 (т. е. коды, абсолютная величина которых больше 174777(8)), заменяются символом «?».

**Пример:**

```
REAL*8 NAME  
LOGICAL*1 OUTPT(12)
```

.

.

```
CALL R50ASC(12,NAME,OUTPT)
```

### 3.9.3. RAD50

Назначение: упаковка шести символов КОИ-7 в RADIX-50.

Обращение: A= RAD50(BC)

Описание параметров:

BC — буквенно-цифровой литерал или имя массива, содержащего упаковываемые символы КОИ-7.

Значение функции: символы RADIX-50, записанные в две ячейки памяти как данное типа REAL\*4.

**ПРИМЕЧАНИЕ.** Подпрограмма RAD50 может использоваться для записи спецификаций файлов в коде RADIX-50.

**Пример:**

следующие командные строки эквивалентны:

```
A= RAD50(LINE)
```

и

```
CALL IRAD50(6,LINE,A)
```

## 3.10. Работа в мультитерминальном режиме

### 3.10.1. MTATCH (режим FB/XM)

Назначение: установление связи задания с указанным терминалом.

Обращение: I=MTATCH(UN[,SA,],[NJ])

Описание параметров:

SA — восьмеричная константа, определяющая адрес слова асинхронного состояния терминала;

NJ — номер задания, связанного с терминалом, если терминал занят. Значение параметра NJ устанавливается подпрограммой MTATCH.

Значение функции:

I=0 подпрограмма выполнена;

I=3 номер несуществующего устройства;

I=5 устройство связано с другим заданием;

I=6 адрес слова асинхронного состояния терминала находится за пределами допустимых виртуальных адресов (режим XM).

#### ПРИМЕЧАНИЯ.

1. Подпрограмма MTATCH должна быть выполнена до того, как какое-либо задание может использовать терминал в мультитерминальном режиме.

2. Задание использования слова асинхронного состояния терминала осуществляется при генерации системы ФОДОС-2. Формат слова асинхронного состояния терминала приведен в приложении 4.

3. Если слово асинхронного состояния терминала не используется, значение параметра SA должно быть равно нулю.

Пример:

```
INTEGER*2 UNIT,SBLOK(4),STAT(8),ASW,STRING(41),PROMPT(8)
LOGICAL*1 TEND(11)
REAL*4 TESTM(9)
DATA PROMPT/'EN','TE','R','ST','RI','NG','>',"200/
DATA TEND/' ','E','N','D',' ','T','E','S','T',' ','O/
DATA TS/'ST','ATCH','GET','SET',' ',' ',' ',' ','DTCH'/
TYPE 106
L=1
IF(MTSTAT(STAT).NE.0)GOTO 999
TYPE 99,STAT(3)
TYPE 100
ACCEPT 101,UNIT
IF(UNIT.EQ.99) STOP 'END OF MULTI-TERMINAL TEST'
TYPE 110
ACCEPT 111,IASW
IF(IASW.EQ.'Y')IER=MTATCH(UNIT,ASW,JOB)
IF(IASW.NE.'Y')IER=MTATCH(UNIT,O,JOB)
L=2
IF(IER)GOTO 999
L=3
```

```

        IF(MTGET(UNIT,SBLOK(1)).NE.0)GOTO 999
        TYPE 102,UNIT,SBLOK
        CALL SETUP(SBLOK,UNIT)
        L=4
        IF(MTSET(UNIT,SBLOK(1)).NE.0)GOTO 999
        TYPE 102,UNIT,SBLOK
20     TYPE 103
        TYPE 104
        TYPE 105
30     CALL MTIN(UNIT,J)
        CALL MTDOUT(UNIT,J)
        IF(J.NE.10)GOTO 30
        CALL MTRCTO(UNIT)
        IF(SBLOK(1).AND."10000")GOTO 40
        SBLOK(1)=SBLOK(1).OR."10000
        IF(MTSET(UNIT,SBLOK(1)).NE.0)GOTO 999
        GOTO 30
40     SBLOK(1)=SBLOK(1).AND..NOT."10000

        IF(MTSET(UNIT,SBLOK(1)).NE.0)GOTO 999
        IF(IASW.NE.'Y')GOTO 60
        TYPE 109
50     IF(.NOT.ASW.AND."40000")GOTO 50
55     CALL MTIN(UNIT,J)
        CALL MTDOUT(UNIT,J)
        IF(J.NE.10)GOTO 55
        CALL MTRCTO(UNIT)
60     CALL GTLIN(STRING,PROMT)
        CALL MTPRNT(UNIT,STRING)
        CALL MTPRNT(UNIT,TEND)
        L=9
        TYPE 108,UNIT
        IF(MTDOTCH(UNIT).EQ.0)GOTO 5
999    TYPE 909,TS(L),IER
        GOTO 5
99     .FORMAT('OTHER ARE',I3,'UNITS ON THIS SYSTEM')
100    FORMAT('UNIT # TO BE TESTED?')
101    FORMAT(I2)
102    FORMAT('  OUNIT',I3,' STATUS=',408)
103    FORMAT('  OGO TO TT BEGIN TESTED ENTER 2 LINES + <BK>')
104    FORMAT('  1ST LINE: INPUT WILL BE ECHOED THEN REPEATED')
105    FORMAT('  2ND LINE: TEST - INPUT ECHOED VIA MTDOUT')
106    FORMAT('  1  SYSLIB MT ROUTINE TEST PROGRAM')
108    FORMAT('  ABOUT TO DETATCH UNIT#',I2)
109    FORMAT('  TEST ASW - INPUT WILL BE ECHOED')
110    FORMAT('TEST ASYNCH STATUS WORD FUNCTION?')
111    FORMAT(A1)
909    FORMAT(' OMT',A4,' ERROR CODE =',I3)
        END
        SUBROUTINE SETUP(SBLOK,UNIT)
        INTEGER SBLOK(4),UNIT
        TYPE 100
        ACCEPT 101,J
        IF(J)SBLOK(1)=J
        TYPE 102
        ACCEPT 101,J
        TYPE 103

```

```

ACCEPT 101, I
IF (I.OR.J) SBLOK(3)=I*256+J
5 TYPE 104
ACCEPT 105, I
IF (I) SBLOK(4)=SBLOK(4)/256*256+I
RETURN
100 FORMAT('CONFIG BIT MASK: ')
101 FORMAT(06)
102 FORMAT('CHAR REQUIRING FILLER: ')
103 FORMAT('# OF FILL CHARS: ')
104 FORMAT('CARRIAGE WIDTH: ')
105 FORMAT(13)
END

```

### 3.10.2. MTDTCH (режим FB/XM)

Назначение: прерывание связи задания с терминалом.

Обращение: I=MTDTCH(UN)

Значение функции:

I=0 подпрограмма выполнена;

I=2 терминал не связан с заданием;

I=3 номер несуществующего устройства.

ПРИМЕЧАНИЕ. Подпрограмма MTDTCH используется совместно с подпрограммой MTATCH.

### 3.10.3. MTGET (режим FB/XM)

Назначение: определение состояния указанного терминала.

Обращение: I=MTGET(UN,SA[,NJ])

Описание параметров:

SA — имя четырехсловного массива INTEGER\*2, определяющего область памяти для приема информации о состоянии терминала;

NJ — номер задания, связанного с терминалом, если терминал занят.

Значение функции:

I=0 подпрограмма выполнена;

I=2 терминал не связан с заданием;

I=3 номер несуществующего устройства;

I=4 терминал связан с другим заданием (номер задания присваивается параметру NJ);

I=6 адрес слова асинхронного состояния терминала находится за пределами допустимых виртуальных адресов (режим XM).

### 3.10.4. MTIN (режим FB/XM)

Назначение: ввод символа с указанного терминала.

Обращение: I=MTIN(UN,DA[,CC] [,OS])

Описание параметров:

DA — имя массива для приема вводимых с терминала символов;

CC — константа, определяющая число вводимых символов;

OS — константа, определяющая число фактически введенных символов.

Значение функции:

I=0 подпрограмма выполнена;

I=1 отсутствует символ в буфере ввода;

I=2 терминал не связан с заданием;

I=3 номер несуществующего устройства.

**ПРИМЕЧАНИЯ.**

1. Если параметр CC не указан, то вводится один символ.

2. Подпрограмма MTIN является мультитерминальной формой подпрограммы ITTINR (см. п. 3.2.6).

### **3.10.5. MTOUT (режим FB/XM)**

Назначение: вывод символа на указанный терминал.

Обращение: I=MTOUT(UN,DA[,CC] [,OS])

Описание параметров:

DA — имя массива, содержащего выводимые символы;

CC — константа, определяющая число выводимых символов;

OS — константа, определяющая число фактически выведенных символов.

Значение функции:

I=0 подпрограмма выполнена;

I=1 буфер вывода переполнен;

I=2 терминал не связан с заданием;

I=3 номер несуществующего терминала;

I=5 адрес буфера пользователя находится за пределами программы (в режиме XM).

**ПРИМЕЧАНИЕ.** Подпрограмма MTOUT является мультитерминальной формой подпрограммы ITTOUR (см. п. 3.2.7).

### **3.10.6. MTPRNT (режим FB/XM)**

Назначение: вывод текстового данного на указанный терминал.

Обращение: I=MTPRNT(UN,SI)

Описание параметров:

SI — буквенно-цифровой литерал или имя массива, содержащего текстовое данное, которое должно быть выведено.

Значение функции:

I=0 подпрограмма выполнена;

I=1 буфер вывода переполнен;

I=2 терминал не связан с заданием;

I=3 номер несуществующего терминала;

I=5 адрес буфера пользователя находится за пределами программы (режим XM).

**ПРИМЕЧАНИЕ.** Подпрограмма MTPRNT является мультитерминальной формой подпрограммы PRINT (см. п. 3.2.11).

### 3.10.7. MTRCTO (режим FB/XM)

Назначение: отмена команды CY/D.

Обращение: I=MTRCTO(UN)

Значение функции:

I=0 подпрограмма выполнена;

I=2 терминал не связан с заданием;

I=3 номер несуществующего устройства.

ПРИМЕЧАНИЕ. Подпрограмма MTRCTO является мультитерминальной формой подпрограммы RCTRLO (см. п. 3.6.11).

### 3.10.8. MTSET (режим FB/XM)

Назначение: задание характеристик указанного терминала.

Обращение: I=MTSET(UN,MS)

Описание параметров:

MS — имя четырехсловного массива INTEGER\*2, определяющего область памяти, содержащую характеристики терминала.

Значение функции:

I=0 подпрограмма выполнена;

I=2 терминал не связан с заданием;

I=3 номер несуществующего устройства;

I=6 адрес слова асинхронного состояния терминала находится за пределами допустимых виртуальных адресов (режим XM).

### ПРИМЕЧАНИЯ.

1. Для восстановления прежних характеристик терминала необходимо перезагрузить систему ФОДОС-2.

2. Информация по установлению характеристик терминала приведена в документе [1] при описании запроса .MTSET.

### 3.10.9. MTSTAT (режим FB/XM)

Назначение: получение информации о состоянии мультитерминальной системы.

Обращение: I=MTSTAT(A8)

Описание параметров:

A8 — имя восьмисловного массива для записи следующей информации:

A8(1) — смещение от начала резидентного монитора первого контрольного блока терминала;

A8(2) — смещение от начала резидентного монитора контрольного блока терминала для системного терминала;

A8(3) — число контрольных блоков терминала, созданных в системе (0—16(10));

A8(4) — размер контрольного блока терминала (в битах);

A8(5)—A8(8) — не используются.

Значение функции:

I=0 подпрограмма выполнена;

I=5 область A8 недоступна в режиме XM.

### 3.11. Вспомогательные операции

#### 3.11.1. IADDR

Назначение: определение абсолютного адреса памяти константы, выражения или переменной.

Обращение: I=IADDR(AK)

Описание параметров:

AK — константа, выражение или переменная, адрес памяти которой надо определить.

Пример:

IADDR можно использовать для определения адреса внешнего имени:

```
EXTERNAL CAREA
```

```
J=IADDR(CAREA)
```

#### 3.11.2. IGETSP

Назначение: назначение свободной области памяти для выполнения программы.

Обращение: I=IGETSP(MN,MX,IR)

Описание параметров:

MN — целое, определяющее минимальный объем свободной памяти (число слов);

MX — целое, определяющее максимальный объем свободной памяти;

IR — имя переменной типа целый, которой подпрограмма присваивает адрес начала свободной области памяти.

Значение функции:

I=-1 недостаточно свободной памяти для выделения требуемого минимального объема памяти;

I=N размер выделенной области памяти ( $MN \leq N \leq MX$ ).

ПРИМЕЧАНИЕ. Если недостаточно свободной памяти для выполнения программы, печатаются сообщения об ошибках с номерами 29, 30, 42.

#### 3.11.3. INTSET

Назначение: назначение подпрограммы на языке ФОРТРАН для обслуживания прерываний.

Обращение: I=INTSET(VT,PR,ID,CR)

Описание параметров:

VT — восьмеричная константа или имя переменной типа целый, определяющей адрес вектора прерывания;

PR — целое, определяющее приоритет (4—7), по которому устройство запрашивает прерывание;

ID — переменная типа целый, определяющая идентификатор, передаваемый подпрограмме обслуживания прерываний;

CR — имя подпрограммы на языке ФОРТРАН, назначаемой для обслуживания прерываний.

Значение функции:

I=0 подпрограмма выполнена;

I=1 недопустимый вектор прерывания;

I=3 недостаточно памяти для организации связи.

Ограничения:

1. INTSET не может быть вызвана из подпрограммы завершения или обслуживания прерываний.

2. Подпрограмма обслуживания прерываний не может вызывать USR.

3. Бит разрешения прерывания не должен быть установлен на соответствующем устройстве раньше окончания выполнения INTSET.

ПРИМЕЧАНИЯ.

1. Имя подпрограммы обслуживания прерываний CR должно быть объявлено как внешнее имя в программе, которая вызывает INTSET. Подпрограмма должна иметь один параметр:

SUBROUTINE CR(ID)

INTEGER ID

2. Параметр ID позволяет различать прерывания от разных векторов, если подпрограмма CR обслуживает несколько устройств. Если подпрограмма обслуживания прерываний введена, значение параметра ID будет равно значению, указанному для идентификатора в соответствующем вызове INTSET.

3. При выполнении INTSET в режиме FB и XM рекомендуется использовать подпрограмму DEVICE для предотвращения прерываний, не связанных с окончанием программы.

4. Если до окончания выполнения подпрограммы обслуживания прерываний происходит более двух новых прерываний по тому же вектору, выдается сообщение:

?SYSLIB—F—INTERRUPT OVERRUN

и выполнение задания прекращается.

5. Программный запрос .PROTECT закрепляет один из векторов за данным заданием, но если вектор прерывания защищен, сообщение об ошибке не выдается. Описание запроса .PROTECT приведено в документе [1].

6. Подпрограмма обслуживания прерываний выполняется как подпрограмма завершения по программному запросу .SYNCH.

7. Аргумент PR может быть использован в программном запросе .INTEN.

**Пример:**  
 EXTERNAL CLKSUB  
 .  
 .  
 .  
 I=INTSET («104,6,0,CLKSUB)  
 IF (I.NE.0) GOTO 100  
 .  
 .  
 .  
 END  
 SUBROUTINE CLKSUB(ID)  
 .  
 .  
 .  
 END

#### 3.11.4. IPEEK

Назначение: чтение слова по абсолютному адресу.

Обращение: I=IPEEK(AB)

Описание параметров:

AB — восьмеричная константа или имя переменной типа целый, определяющей абсолютный адрес.

Значение функции: I=N, где N — слово по абсолютному адресу AB.

**ПРИМЕЧАНИЯ.**

1. Подпрограмму можно использовать для проверки регистров устройства или любой ячейки памяти.

2. Если AB нечетно, происходит прерывание программы и управление передается монитору.

**Пример:**

ISWIT=IPEEK («177570)

#### 3.11.5. IPEEKB

Назначение: чтение байта по абсолютному адресу.

Обращение: I=IPEEKB(AB)

Описание параметров:

AB — восьмеричная константа или имя переменной типа целый, определяющей абсолютный адрес байта.

Значение функции: I=N, где N — байт по абсолютному адресу AB.

**ПРИМЕЧАНИЕ.** Подпрограмму можно использовать для проверки регистров устройства или любого байта памяти.

**Пример:**

IERR=IPEEKB («53)

#### 3.11.6. IPOKE

Назначение: запись слова по абсолютному адресу.

Обращение: CALL IPOKE (AB,IV)

Описание параметров:

AB — восьмеричная константа или имя переменной типа целый, определяющей абсолютный адрес, по которому осуществляется запись;

IV — восьмеричная константа или имя переменной типа целый, определяющей значение, которое должно быть записано по абсолютному адресу.

**ПРИМЕЧАНИЯ.**

1. Подпрограмму IPOKE можно использовать для записи значений в регистры устройства.

2. Если AB нечетно, происходит прерывание программы и управление передается монитору.

**Пример:**

В данном примере значение переменной IVAl записывается по адресу клавишного регистра:

CALL IPOKE («177570,IVAL)

Для установки разряда 12 в ССЗ без изменения других разрядов используется следующая команда:

CALL IPOKE («44,«10000.OR.IPEEK («44))

### 3.11.7. IPOKEB

Назначение: запись байта по абсолютному адресу.

Обращение: I=IPOKEB (AB,IV)

Описание параметров:

AB — восьмеричная константа или имя переменной типа целый, определяющей абсолютный адрес, по которому осуществляется запись;

IV — восьмеричная константа или имя переменной типа целый, определяющей значение, которое должно быть записано по абсолютному адресу.

**ПРИМЕЧАНИЕ.** IPOKEB можно использовать для записи значений в регистры устройства.

### 3.11.8. ISPY

Назначение: чтение слова по адресу относительно смещения от начального адреса резидентного монитора (RMON).

Обращение: I=ISPY (RM)

Описание параметров:

RM — имя переменной типа целый, определяющей смещение от начала RMON.

Значение функции: I=N, где N — слово по относительному адресу.

**ПРИМЕЧАНИЕ.** Смещение от начала резидентного монитора определяется с помощью программного запроса .GVAL (см. [1]).

**Пример:**  
IF (ISPY («300).AND.1) GOTO 200

### 3.11.9. IPUT

Назначение: изменение содержимого ячейки памяти, размещенной с заданным смещением от начала резидентного монитора.

Обращение: I=IPUT(IO,VL)

Описание параметров:

IO — восьмеричная константа, определяющая смещение от начала резидентного монитора ячейки, содержимое которой будет модифицировано;

VL — целое для замены текущего содержимого ячейки.

Значение функции: I=N, где N — старое (замененное) значение ячейки памяти.

**Пример:**  
ISIZE=IPUT («314,100)

## 4. СООБЩЕНИЯ

Ниже приведено сообщение библиотеки:

?SYSLIB—F—INTERRUPT OVERRUN

Причина. До окончания выполнения подпрограммы обслуживания прерываний произошло более двух новых прерываний по тому же вектору.

Действие. Изменить программу, уменьшив, если возможно, число обращений к подпрограмме обслуживания прерываний и разместив их в программе равномерно. Сократить подпрограмму обслуживания прерываний.

## ПРИЛОЖЕНИЕ 1

### СПИСОК СИСТЕМНЫХ ПОДПРОГРАММ

Имя под- программы	Пункт	Наименование
1	2	3
CLOSEC	3.1.1	Операции с файлами Закрытие канала
ICLOSE		
IDELET	3.1.2	Стирание файла
IENTER	3.1.3	Открытие нового файла
IFPROT	3.1.4	Управление защитой файлов
IRENAM	3.1.5	Переименование файла
ISFDAT	3.1.6	Изменение даты создания файла

1	2	3
LOOKUP	3.1.7	Открытие существующего файла
IABTIO	3.2.1	Прерывание ввода-вывода на указанном канале
GTLIN	3.2.2	Ввод текстового данного с системного терминала или косвенного файла
IRCV D	3.2.3	Прием данных от другого задания
IRCVDC		
IRCVDF		
IRCVDW		
IREAD	3.2.4	Ввод данных
IREADC		
IREADF		
IREADW		
ISDAT	3.2.5	Передача данных другому заданию
ISDATC		
ISDATF		
ISDATW		
ITTINR	3.2.6	Ввод символа с системного терминала
ITTOUR	3.2.7	Вывод символа на системный терминал
IWAIT	3.2.8	Ожидание завершения ввода-вывода
IWRITE	3.2.9	Вывод данных
IWRITC		
IWRITF		
IWRITW		
MWAIT	3.2.10	Ожидание завершения передачи данных между заданиями
PRINT	3.2.11	Вывод текстового данного на системный терминал
ICDFN	3.3.1	Операции с каналами
ICHCPY	3.3.2	Увеличение числа каналов
ICSTAT	3.3.3	Установление логической связи с каналом другого задания
ISAVES	3.3.8	Получение информации о состоянии
IFREEC	3.3.4	Канала
IGETC	3.3.5	Возвращение канала
ILUN	3.3.6	Назначение канала
IREOPN	3.3.7	Определение номера канала, с которым связано логическое устройство
PURGE	3.3.9	Повторное открытие канала
		Освобождение канала

1	2	3
IASIGN	3.4.1	Описание устройств и файлов
ICSI	3.4.2	Установление логической связи между файлом и номером логического устройства Синтаксический анализ командной строки
CVTTIM	3.5.1	Работа с таймером
GTIM	3.5.2	Преобразование времени из внутреннего представления в часы, минуты, секунды и доли секунды
ICMKT	3.5.3	Определение текущего времени во внутреннем представлении
ISCHED	3.5.4	Отмена выполнения ISCHED, ITIMER, MRKT
ISDTTM	3.5.5	Планирование выполнения подпрограммы завершения на языке ФОРТРАН в заданное время суток
ISLEEP	3.5.6	Установка системной даты и времени
ITIMER	3.5.7	Задержка выполнения основного задания на заданный промежуток времени (время — в часах, минутах, секундах и долях секунды)
ITWAIT	3.5.8	Планирование выполнения подпрограммы завершения на языке ФОРТРАН по истечении заданного промежутка времени
IUNTIL	3.5.9	Задержка выполнения основного задания на заданный промежуток времени (время — во внутреннем представлении)
JTIME	3.5.10	Задержка выполнения основного задания до заданного времени суток
MRKT	3.5.11	Преобразование времени во внутреннее представление
SECNDS	3.5.12	Планирование выполнения подпрограммы завершения на языке АССЕМБЛЕРА по истечении заданного промежутка времени
TIMASC	3.5.13	Определение прошедшего времени
TIME	3.5.14	Преобразование времени из внутреннего представления в часы, минуты, секунды
CHAIN	3.6.1	Определение текущего времени в часах, минутах, секундах
DEVICE	3.6.2	Системные операции
GTJB	3.6.3	Передача управления другому фоновому заданию
IGTJB	3.6.4	Запись данных по указанным адресам
IDSTAT	3.6.4	Получение информации о выполняемом задании
IFETCH	3.6.5	Получение информации о состоянии устройства
IQSET	3.6.6	Загрузка драйвера
		Увеличение числа элементов очереди ввода-вывода

1	2	3
ISPFN ISPFNC ISPFNF ISPFNW ITLOCK	3.6.7  3.6.8	Выполнение специальной операции ввода-вывода  Блокирование USR в памяти, если она не используется другим заданием
LOCK RCHAIN	3.6.9 3.6.10	Блокирование USR в памяти Получение доступа к переменным переданным по CHAIN
RCTRLO RESUME	3.6.11 3.6.12	Отмена команды CU/D Продолжение выполнения задания, приостановленного по SUSPND
SCCA SETCMD	3.6.13 3.6.14	Отмена команды CU/C Передача командной строки клавиатурному монитору
SUSPND	3.6.15	Задержка выполнения задания и разрешение выполнения подпрограммы завершения
UNLOCK	3.6.16	Освобождение USR из памяти
AJFLT	3.7.1	Операции с данными INTEGER*4 Преобразование данного типа INTEGER*4 в REAL*4 и присвоение результата преобразования функции
DJFLT	3.7.2	Преобразование данного типа INTEGER*4 в REAL*8 и присвоение результата функции
IAJFLT	3.7.3	Преобразование данного типа INTEGER*4 в REAL*4
IDJFLT	3.7.4	Преобразование данного типа INTEGER*4 в REAL*8
IJCVT	3.7.5	Преобразование данного типа INTEGER*4 в INTEGER*2
JADD JAFIX	3.7.6 3.7.7	Сложение данных типа INTEGER*4 Преобразование данного типа REAL*4 в INTEGER*4
JCMP JDFIX	3.7.8 3.7.9	Сравнение данных типа INTEGER*4 Преобразование данного типа REAL*8 в INTEGER*4
JDIV JICVT	3.7.10 3.7.11	Деление данных типа INTEGER*4 Преобразование данного типа INTEGER*2 в INTEGER*4
JJCVT	3.7.12	Преобразование времени из внутреннего представления в INTEGER*4 или наоборот
JMOV	3.7.13	Присваивание значений переменной INTEGER*4
JMUL JSUB	3.7.14 3.7.15	Умножение данных типа INTEGER*4 Вычитание данных типа INTEGER*4
CONCAT	3.8.3	Операции с текстовыми данными Объединение двух текстовых данных

1	2	3
GETSTR	3.8.4	Ввод форматной записи с логического устройства
INDEX	3.8.5	Поиск символа или последовательности символов в текстовом данном
INSERT	3.8.6	Замена части текстового данного
ISCOMP	3.8.7	Сравнение текстовых данных
SCOMP		
VERIFY	3.8.8	Определение позиции несовпадающих символов двух текстовых данных
IVERIF	3.8.9	Определение числа символов в текстовом данном
LEN	3.8.10	Вывод форматной записи на устройство
PUTSTR	3.8.11	Копирование текстового данного заданное число раз
REPEAT		Копирование текстового данного
SCOPY	3.8.12	Дополнение текстового данного пробелами
STRPAD	3.8.13	Копирование части текстового данного
SUBSTR	3.8.14	Модификация текстового данного
TRANSL	3.8.15	Удаление конечных пробелов из текстового данного
TRIM	3.8.16	Преобразование RADIX-50
IRAD50	3.9.1	Упаковка символов КОИ-7 в RADIX-50
R50ASC	3.9.2	Распаковка символов RADIX-50 в КОИ-7
RAD50	3.9.3	Упаковка шести символов КОИ-7 в RADIX-50
MTATCH	3.10.1	Работа в мультитерминальном режиме
MTDTCH	3.10.2	Установление связи задания с указанным терминалом
MTGET	3.10.3	Прерывание связи задания с терминалом
MTIN	3.10.4	Определение состояния указанного терминала
MTOUT	3.10.5	Ввод символа с указанного терминала
MTPRNT	3.10.6	Вывод символа на указанный терминал
MTRCTO	3.10.7	Вывод текстового данного на указанный терминал
MTSET	3.10.8	Отмена команды СУ/D
MTSTAT	3.10.9	Задание характеристик указанного терминала
IADDR	3.11.1	Получение информации о состоянии мультитерминальной системы
IGETSP	3.11.2	Вспомогательные операции
INTSET	3.11.3	Определение абсолютного адреса памяти константы, выражения или переменной
IPEEK	3.11.4	Назначение свободной области памяти
		Назначение подпрограммы на языке ФОРТРАН для обслуживания прерываний
		Чтение слова по абсолютному адресу

1	2	3
IPEEKB	3.11.5	Чтение байта по абсолютному адресу
IPOKE	3.11.6	Запись слова по абсолютному адресу
IPOKEB	3.11.7	Запись байта по абсолютному адресу
ISPY	3.11.8	Чтение слова по относительному адресу
IPUT	3.11.9	Изменение содержимого ячейки памяти по относительному адресу

В таблице данного приложения приведен список системных подпрограмм.

#### ПРИМЕЧАНИЯ.

1. Подпрограммы, подчеркнутые одной чертой, выполняются только в режиме FB и XM.
2. Подпрограммы, подчеркнутые дважды, выполняются в режиме SJ при наличии их поддержки во время генерации монитора.

## ПРИЛОЖЕНИЕ 2

### ОСОБЕННОСТИ ВЫПОЛНЕНИЯ ПОДПРОГРАММ

В данном приложении приведены системные подпрограммы, требующие для своего выполнения дополнительных условий: наличия **USR** в памяти, элемента очереди, а также подпрограммы, вызывающие подпрограммы завершения и подпрограммы, обращение к которым осуществляется только на языке **ФОРТРАН**. Такие подпрограммы помечены в таблице с помощью «да». Подпрограммы, не приведенные в данной таблице, не требуют каких-либо дополнительных условий для своего выполнения.

Имя подпрограммы	Требование <b>USR</b>	Использование элемента очереди	Вызов подпрограммы завершения	Обращение только на <b>ФОРТРАНе</b>
1	2	3	4	5
CLOSEC	да			
ICLOSE	да			
GETSTR	да			да
GTLIN	да			
IASIGN				да
ICDFN	да			да
ICSI	да			
IDelet	да			

1	2	3	4	5
IDSTAT	да			
IENTER	да			
IFETCH	да			
IFREEC				да
IGETC				да
IGETSP				да
ILUN				да
INTSET			да	да
IQSET	да			да
IRCV		да		
IRCVDC		да	да	
IRCVDF		да	да	да
IRCVDW		да		
IREAD		да		
IREADC		да	да	
IREADF		да	да	
IREADW		да		да
IRENAM	да			
ISCHED		да	да	да
ISDAT		да		
ISDATC		да	да	
ISDATF		да	да	да
ISDATW		да		
ISLEEP		да		
ISPFN		да		
ISPFNC		да		
ISPFNF		да	да	да
ISPFNW		да		
ITIMER		да	да	да
ITLOCK	да			
ITWAIT		да		
IUNTIL		да		
IWRITC		да	да	
IWRITE		да		
IWRITF		да	да	да
IWRITW		да		
LOCK	да			
LOOKUP	да			
MRKT		да	да	
MWAIT		да		
PUTSTR	да			
SECNDS				да
				да

ПРИМЕЧАНИЕ. Параметры подпрограмм, использующих  
USR, не должны размещаться в области свопинга USR.

## ПРИЛОЖЕНИЕ 3

### ФОРМАТ СЛОВА СОСТОЯНИЯ КАНАЛА

В таблице данного приложения приведено назначение рядов слова состояния канала.

Ряд	Значение	Назначение
0	1	Аппаратная ошибка на канале
1—5	N	N — индекс устройства в таблицах монитора: описывает физическое устройство, связанное с используемым каналом
6	1	На канале выполняется запрос .RENAME
7	1	На канале выполнен запрос .ENTER; справочник устройства должен модифицироваться по запросу .CLOSE
8—12	1	Номер сегмента справочника (от 1 до 37(8)), в котором находится запись о текущем файле
13	1	Обнаружен конец файла
14	1	Не используется
15	1	Канал занят

## ПРИЛОЖЕНИЕ 4

### ФОРМАТ СЛОВА АСИНХРОННОГО СОСТОЯНИЯ ТЕРМИНАЛА

В таблице данного приложения приведено назначение рядов слова асинхронного состояния терминала.

Ряд	Значение	Назначение
0—12		Не используются
13	1	Буфер вывода терминала пуст
14	1	Терминал включен
15	1	С терминала подана команда СУ/С дважды

### ПЕРЕЧЕНЬ ССЫЛОЧНЫХ ДОКУМЕНТОВ

1. Операционная система ФОДОС-2  
Редактор связей  
Руководство оператора
2. Операционная система ФОДОС-2

- ФОРТРАН/ФОДОС-2  
Транслятор с ФОРТРАНА  
Руководство программиста
3. ФОРТРАН/ФОДОС-2  
Библиотека ФОРТРАНА  
Руководство программиста
4. ФОРТРАН/ФОДОС-2  
Описание языка

## СООБЩЕНИЯ СИСТЕМЫ

### 1. СОКРАЩЕНИЯ И УСЛОВНЫЕ ОБОЗНАЧЕНИЯ

В пределах каждого раздела сообщения располагаются в алфавитном порядке.

Некоторые сообщения, описанные в настоящем документе, содержат одну или несколько приведенных ниже общих ссылок, которые выбираются непосредственно из выполняемой программы:

***	— метка, имя или значение, которое транслятор с языка ФОРТРАН выбирает из выполняемой программы;
X	— переключатель, который выбирается из введенной команды;
AAAAAA	— метка или имя, которое ФОДОС-2 выбирает из выполняемой программы;
A	— символ, используемый ФОДОС-2 в качестве сокращения имени файла или переключателя;
DEV:FILNAM.TYP	— спецификация файла, которую система выбирает из введенной команды;
MMMMMM, NNNNNN	— числовые значения, которые ФОДОС-2 рассматривает как часть сообщений: контрольные суммы, адреса или смещения;
N	— число, обозначающее номер устройства в спецификации файла.

**ПРИМЕЧАНИЕ.** Общие ссылки не используются в упорядочении сообщений по алфавиту.

Для того, чтобы найти интересное сообщение в настоящем документе, следует выполнить следующее:

1) определить, какому разделу этого документа принадлежит сообщение;

2) игнорировать специальный символ (например, знак вопроса (?), кавычки («»));

3) игнорировать число или имя в сообщении, которое является специфическим для программы или файла пользователя;

4) рассмотреть сообщение, учитывая только оставшиеся символы.

В этом документе каждое системное сообщение приводится в той форме, в какой оно воспроизводится на терминале или в листинге.

Формат системных сообщений следующий:

?прог—Т—сообщение

где прог — имя программы;

Т — тип ошибки;

<сообщение> — текст сообщения.

Типы ошибок и причины их вызвавшие приведены в табл. 1.

Таблица 1

Тип ошибки	Обозначение	Причина
1	2	3
Информационное сообщение	I	Выполнение продолжается. Программа обнаружила условие, о котором информирует Сообщение выдается на терминал или в файл листинга
Предупреждение	W	Выполнение продолжается. Программа обнаружила условие, которое может вызвать ошибку. Соответствующие действия могут быть необходимы. Сообщение выдается на терминал или в файл листинга
Ошибка	E	Выполнение может завершиться. Программа обнаружила ошибку, которая вызовет другие ошибки. Сообщение выдается на терминал или в файл листинга
Неустранимая	F	Выполнение завершается. Программа обнаружила серьезную ошибку. Следует ввести другую команду для продолжения обработки. Сообщение выдается на терминал или в файл листинга

1	2	3
Безусловное прерывание	U	Выполнение завершается. Программа обнаружила очень серьезную ошибку, которая мешает дальнейшей обработке. Сообщение выдается на терминал или в файл листинга

**ПРИМЕЧАНИЕ.** По команде монитора SET можно изменить тип ошибки в ответе системы. Например, команда SET ERROR влияет на выполнение косвенного командного файла, вызывая предусмотренные прерывания. Более подробно о команде SET ERROR см. в [1].

Формат сообщений об ошибках процессора косвенных управляющих файлов (IND) отличается от формата сообщений, описанного выше, и зависит от того, произошла ли ошибка во время обработки директив управляющего файла или ввода-вывода данных.

Формат сообщений об ошибках управляющего файла следующий:

?прог—Т—сообщение

строка управляющего файла

где прог

— имя программы;

T

— тип ошибки;

<сообщение>

— текст сообщения;

строка управляющего файла

— строка в управляющем файле, которая вызвала ошибку.

Сообщения об ошибках ввода-вывода включают первичные и вторичные сообщения. Формат первичных сообщений об ошибках ввода-вывода такой же, как и формат сообщений об ошибках управляющего файла.

Формат сообщений об ошибках ввода-вывода следующий:

?прог—Т—первичное сообщение

вторичное сообщение

строка управляющего файла

Формат сообщений об ошибках транслятора с языка ФОРТРАН и библиотеки ФОРТРАН, интерпретатора с языка БЕЙСИК, экранного редактора (K13) и редактора командной строки (SL) описан соответственно в разделах 2, 3 и 6.

## 2. СООБЩЕНИЯ ТРАНСЛЯТОРА С ЯЗЫКА ФОРТРАН И БИБЛИОТЕКИ ФОРТРАН

### 2.1. Сообщения транслятора с языка ФОРТРАН

Транслятор с языка ФОРТРАН, обрабатывая исходную программу, осуществляет предварительный, синтаксический и семантический контроль.

2.1.1. Сообщения предварительного контроля появляются в поле исходной программы листинга непосредственно после предложения, к которому они относятся, при этом трансляция программы не прекращается.

Сообщения предварительного контроля имеют вид

**\*\*\*\*\*<Y>**

где Y — буква латинского алфавита, используемая для обозначения ошибки.

Список ошибок предварительного контроля приведен в табл. 2.

Сообщения синтаксического и семантического контроля появляются в листинге программы в поле диагностики и имеют вид

**IN LINE NNNN, ERROR: SSSS**

или

**IN LINE NNNN, WARNING: SSSS**

где NNNN — номер внутренней последовательности предложения;

ERROR — сообщение об ошибке;

WARNING — предупреждение об ошибке (выводится при задании переключателя /W);

SSSS — текст сообщения.

Отметка **\*\*\*\*** в тексте сообщения означает, что в данном месте текста будет указано имя переменной, массива или метка.

На этапе синтаксического и семантического контроля могут возникнуть неустранимые ошибки.

Сообщения о неустранимых ошибках выводятся на терминал. Появление этих ошибок вызывает прекращение трансляции программы.

Неустранимые ошибки возникают в случае неработоспособности устройства, неправильной подачи команды или если оператор сложен для обработки. Текущая команда или оператор игнорируется, трансляция прекращается.

Сообщения о неустранимых ошибках имеют вид  
**?FORTRAN—F—SSSS**

где SSSS — текст сообщения.

По окончании трансляции на терминал выводится сообщение:

?FORTRAN—I—[S] ERRORS:N, WARNING:M

где S — имя программного модуля;

N — число ошибок;

M — число сообщений, предупреждающих об ошибках.

Таблица 2

Обозначение	Вид ошибки	Причина ошибки	Действие
1	2	3	4
B	W	В позициях 1—5 строки-продолжения встретился символ, отличный от пробела	Символы в этих позициях игнорируются
C	W	Неправильное продолжение: комментарий или начальная строка первого предложения является строкой-продолжением	Строка игнорируется
E	W	Отсутствует заключительная строка	Модуль транслируется
H	ER	Константа Холлерита содержит более 255 символов	Константа игнорируется
I	W	Использован символ, не входящий в алфавит языка и не содержащийся в текстовой константе или в комментарии	Символ игнорируется
K	W	Недопустимый символ в метке предложения	Метка игнорируется
L	W	Строка содержит более 80 символов (включая символы пробела и табуляции)	Строка усекается
M	W	Множественное определение метки	Повторные метки игнорируются
P	ER	Предложение содержит непарные скобки	Предложение игнорируется
S	ER	Синтаксическая ошибка в записи оператора (например, многочисленные знаки равенства)	Оператор игнорируется
U	ER	Недопустимый формат предложения	Предложение игнорируется

ПРИМЕЧАНИЕ. В графе «Вид ошибки» ER обозначает ошибку, W — предупреждение об ошибке.

### 2.1.2. Сообщения синтаксического и семантического контроля

ACCESS='DIRECT' REQUIRES FORM='UNFORMATTED'  
 Причина. Ключевое слово FORM='FORMATTED' было указано для файла прямого доступа. ФОРТРАН обес-

печивает только бесформатный ввод-вывод прямого доступа.

Действие. Согласовать формат файла и метод доступа.

**ADJUSTABLE DIMENSIONS ILLEGAL FOR ARRAY\*\*\***

Причина. Имя массива или имена переменных, представляющих регулируемые размеры, не указаны в подпрограмме как формальные параметры типа целый.

Действие. Массиву присваивается единичная размерность. Исправить исходную программу.

**ARRAY \*\*\* EXCEEDS MAXIMUM SIZE**

или

**ARRAY EXCEEDS MAXIMUM SIZE**

Причина. Недостаточно памяти для указанного массива (первое сообщение) или для всех массивов в подпрограмме.

Действие. Проверить объявления массивов и уменьшить общие требования к памяти для массивов.

**ARRAY \*\*\* HAS TOO MANY DIMENSIONS**

Причина. Массив имеет более семи измерений.

Действие. Исправить объявление массива и предложения, связанные с ним.

**\*\*\* ATTEMPTS TO EXTEND COMMON BLOCK BACKWARDS**

Причина. Попытка расширения блока COMMON влево объявлением эквивалентности.

Действие. Согласовать объявления общих объектов и эквивалентности.

**COMMON BLOCK EXCEEDS MAXIMUM SIZE**

Причина. Попытка предоставить для блока COMMON больше места, чем физически возможно (более 32К слов).

Действие. Исправить предложение.

**CONSTANT IN FORMAT STATEMENT NOT IN RANGE**

Причина. Целая константа в объявлении формата находится вне допустимых пределов (1—255).

Действие. Исправить объявление формата.

**DANGLING OPERATOR**

Причина. В выражении пропущен операнд.

Действие. Исправить неправильное выражение.

**DEFECTIVE DOTTED KEYWORD**

Причина. Неправильная запись знака операции отношения.

Действие. Исправить запись отношения.

**DEFINE FILE MODE MUST BE 'U'**

Причина. Третий параметр в скобках в объявлении DEFINE FILE не 'U'.

Действие. Исправить параметр в объявлении.

DO TERMINATOR \*\*\*\* PRECEDES DO STATEMENT

Причина. Отсутствует закрывающий оператор тела цикла.

Действие. Поместить закрывающий оператор тела цикла за оператором DO.

EXPECTING LEFT PARENTHESIS AFTER \*\*\*\*

Причина. Отсутствует левая скобка за именем массива или именем функции.

Действие. Исправить предложение.

EXPECTING LEFT PARENTHESIS AFTER SUBPROGRAM NAME

Причина. Не указан список параметров за именем SUBROUTINE или FUNCTION.

Действие. Проверить предложение: нет ли опечатки, не используется ли одно и то же имя для переменной и подпрограммы.

EXTRA CHARACTERS AT END OF STATEMENT

Причина. За допустимым предложением в строке следуют дополнительные символы.

Действие. Проверить: не пропущена ли запятая, нет ли лишних символов в конце предложения и не ошибочен ли символ в шестой позиции следующей строки.

FLOATING CONSTANT NOT IN RANGE

Причина. Значение константы с плавающей точкой в выражении близко к нулю.

Действие. Использовать, если возможно, константу 0.0.

ILLEGAL ADJACENT OPERATOR

Причина. Два знака операции расположены друг за другом.

Действие. Исправить неправильное выражение.

ILLEGAL CHARACTERS IN EXPRESSION

Причина. Недопустимый символ в выражении.

Действие. Исправить выражение.

ILLEGAL DO TERMINATOR ORDERING AT LABEL \*\*\*\*

Причина. Неправильная организация гнезда в операторе цикла.

Действие. Организовать правильно гнездо.

ILLEGAL DO TERMINATOR STATEMENT \*\*\*\*

Причина. Недопустимый закрывающий оператор тела цикла

Действие. Проверить, чтобы закрывающий оператор тела цикла не был оператором GOTO, арифметическим IF, RETURN, другим оператором DO или логическим IF, содержащим один из этих операторов.

ILLEGAL ELEMENT IN I/O LIST

Причина. Список ввода-вывода содержит синтаксическую ошибку.

Действие. Исправить список ввода-вывода.

**ILLEGAL ENCODE/DECODE FORMAT SPECIFIER**

Причина. В операторе ENCODE или DECODE второй параметр в скобках (указатель формата) не является меткой объявления формата или именем массива, содержащим спецификацию формата.

Действие. Исправить данный параметр.

**ILLEGAL ENCODE/DECODE LENGTH EXPRESSION**

Причина. В операторе ENCODE или DECODE первый параметр в скобках (число преобразуемых символов) не является целым выражением.

Действие. Исправить данный параметр.

**ILLEGAL ENCODE/DECODE TARGET**

Причина. В операторе ENCODE или DECODE третий параметр в скобках не является именем массива, элементом массива или именем переменной.

Действие. Исправить данный параметр.

**ILLEGAL INITIAL VALUE EXPRESSION IN DO STATEMENT**

Причина. В операторе DO недопустимое значение начального параметра.

Действие. Исправить значение начального параметра.

**ILLEGAL STATEMENT IN BLOCK DATA**

Причина. В модуле-блоке данных обнаружено недопустимое предложение.

Действие. Исключить недопустимое предложение из модуля-блока данных.

**ILLEGAL STATEMENT ON LOGICAL IF**

Причина. Оператор, содержащийся в логическом IF, не является допустимым (например, оператор DO или другой логический IF).

Действие. Исправить оператор IF.

**ILLEGAL SUBSCRIPTS OR SUBPROGRAM ARGUMENT**

Причина. Недопустимый элемент в списке параметров подпрограммы или в индексе массива.

Действие. Исправить предложение.

**ILLEGAL TYPE FOR OPERATOR**

Причина. В экспоненциальной функции или условном логическом операторе использована переменная недопустимого типа.

Действие. Проверить тип переменной в данной функции или операторе.

**ILLEGAL USAGE OF OR MISSING LEFT PARENTHESIS**

Причина. Присутствует недопустимая или отсутствует левая скобка.

Действие. Исправить ошибочное выражение.

#### INTEGER OVERFLOW

Причина. Целая константа или значение выражения находится вне диапазона от  $-32767$  до  $+32767$ .

Действие. Изменить константу или выражение так, чтобы его значение находилось в указанном диапазоне.

#### INVALID COMPLEX CONSTANT

Причина. Недопустимый формат комплексной константы.

Действие. Исправить формат комплексной константы.

#### INVALID DIMENSIONS FOR ARRAY \*\*\*

Причина. Недопустимые измерения массива.

Действие. Проверить: чтобы ноль не использовался в списке границ, число измерений должно находиться в диапазоне от 1 до 7.

#### INVALID END= OR ERR= KEYWORD

Причина. Недопустимый формат параметра `END=` или `ERR=` в операторе ввода или вывода.

Действие. Исправить оператор ввода или вывода.

#### INVALID EQUIVALENCE

Причина. Объявление эквивалентности недопустимо, или оно не согласуется с предыдущим объявлением эквивалентности.

Действие. Исправить объявление эквивалентности.

#### INVALID FORMAT SPECIFIER

Причина. Указатель формата не является ни меткой объявления формата, ни именем массива, содержащим спецификацию формата.

Действие. Исправить указатель формата.

#### INVALID IMPLICIT RANGE SPECIFIER

Причина. В объявлении типа `IMPLICIT` обнаружен небуквенный символ или указана последовательность символов не в алфавитном порядке.

Действие. Исправить неявное объявление типа.

#### INVALID LOGICAL UNIT

Причина. Неверная ссылка на номер логического устройства.

Действие. Исправить номер устройства так, чтобы он был целой переменной или константой в пределах от 1 до 99.

#### INVALID OCTAL CONSTANT

Причина. Значение восьмеричной константы находится вне диапазона (0—177777) или содержит цифры, отличные от 0—7.

Действие. Исправить восьмеричную константу.

#### INVALID OPTIONAL LENGTH SPECIFIER

Причина. Недопустимый указатель длины в объявлении типа данных.

Действие. Исправить объявление типа данных.

#### INVALID RADIX-50 CONSTANT

Причина. Недопустимый символ в константе RADIX-50.

Действие. Исправить константу RADIX-50.

#### INVALID STATEMENT LABEL REFERENCE

Причина. Ссылка на метку неправильной конструкции.

Действие. Проверить, чтобы метка состояла из десятичных цифр от одной до пяти, расположенных в первых пяти позициях строки, и не состояла из одних нулей.

#### INVALID PROGRAM NAME

Причина. Символическое имя, использованное в операторе вызова подпрограммы или в ссылке на функцию, недопустимо.

Действие. Записать правильно имя.

#### INVALID TARGET FOR ASSIGNMENT

Причина. В арифметическом или логическом операторе присваивания слева от знака равенства указано не имя переменной и не имя элемента массива.

Действие. Исправить оператор.

#### INVALID TYPE SPECIFIER

Причина. Недопустимый тип данных.

Действие. Исправить тип данных.

#### INVALID USAGE OF SUBROUTINE OR FUNCTION NAME

Причина. Используется имя функции или подпрограммы в объявлении DIMENSION, COMMON, DATA, EQUIVALENCE или в объявлении типа данных.

Действие. Исключить имя функции или подпрограммы из данного объявления.

#### INVALID VARIABLE NAME

Причина. Недопустимое имя переменной.

Действие. Исправить имя переменной.

#### LABEL ON DECLARATIVE STATEMENT

Причина. Метка предложения недопустима.

Действие. Удалить метку.

#### MISSING ASSIGNMENT OPERATOR

Причина. В арифметическом или логическом операторе присваивания пропущен знак равенства или он находится не на своем месте.

Действие. Исправить оператор.

**MISSING COMMA**

Причина. Отсутствует запятая.

Действие. Исправить предложение.

**MODE OF EXPRESSION MUST BE INTEGER**

Причина. Используется не целая константа или переменная, например в операторе DO начальный параметр, конечный параметр или параметр приращения имеет не целое значение.

Действие. Исправить оператор.

**MISSING COMMA IN OPEN OR CLOSE KEYWORD LIST**

Причина. Два ключевых слова в операторе OPEN или CLOSE не разделены запятой.

Действие. Исправить оператор.

**MISSING DELIMITER IN EXPRESSION**

Причина. Пропущен знак операции между двумя операндами в выражении.

Действие. Исправить выражение.

**MISSING EXPRESSION**

Причина. Пропущено требуемое выражение (например, конечный параметр в операторе DO).

Действие. Скорректировать предложение.

**MISSING LABEL**

Причина. В операторе отсутствует метка или указана синтаксически неверно.

Действие. Исправить оператор.

**MISSING LABEL LIST AFTER COMMA**

Причина. В операторе перехода по предписанию отсутствует список меток.

Действие. Исправить оператор.

**MISSING LEFT PARENTHESIS AFTER OPEN OR CLOSE**

Причина. В операторе OPEN или CLOSE отсутствует левая скобка перед списком ключевых слов.

Действие. Исправить оператор.

**MISSING OPERATOR AFTER EXPRESSION**

Причина. Выражение в операторе не ограничено запятой, правой скобкой или другим оператором.

Действие. Исправить оператор.

**MISSING QUOTATION MARK**

Причина. В операторе FIND номер логического устройства и номер записи не разделены апострофом.

Действие. Исправить оператор.

**MISSING RIGHT PARENTHESIS**

Причина. Правые скобки находятся не на месте или пропущены.

Действие. Исправить предложение.

#### MISSING 'TO' IN ASSIGN STATEMENT

Причина. Отсутствует 'TO' в операторе ASSIGN.

Действие. Исправить оператор.

#### MISSING VALUE FOR KEYWORD IN OPEN OR CLOSE STATEMENT

Причина. Отсутствует значение ключевого слова в операторе OPEN или CLOSE.

Действие. Исправить оператор.

#### MISSING VARIABLE

Причина. В предложении пропущена переменная.

Действие. Исправить предложение.

#### MISSING VARIABLE OR CONSTANT

Причина. Вместо переменной или константы использован разделитель (запятая, скобки и так далее).

Действие. Проверить формат предложения и исправить его.

#### MODE OF EXPRESSION MUST BE INTEGER

Причина. Выражение или его значение не принадлежит к типу целый.

Действие. Исправить выражение.

#### MODES OF VARIABLE \*\*\*\* AND DATA ITEM DIFFER

Причина. В объявлении DATA тип переменной и связанный с ним элемент списка не согласованы.

Действие. Согласовать тип переменной с элементом списка.

#### MULTIPLE DECLARATION OR VARIABLE \*\*\*\*

Причина. Многократное описание массива или объявление типа переменной.

Действие. Оставить одно объявление типа переменной или одно описание массива.

#### MULTIPLE DECLARATION OF OPEN OR CLOSE KEYWORD

Причина. Многократное определение ключевого слова в операторе OPEN или CLOSE.

Действие. Оставить одно определение ключевого слова.

#### OPEN OR CLOSE KEYWORD VALUE MUST BE QUOTED STRING

Причина. Значение ключевого слова в операторе OPEN или CLOSE не заключено в апострофы.

Действие. Заключить в апострофы значение ключевого слова.

#### OPEN OR CLOSE STATEMENT REQUIRES UNIT=SPECIFIER

Причина. Отсутствует ключевое слово UNIT в операторе OPEN или CLOSE.

Действие. Исправить оператор.

#### PARENTHESES NESTED TOO DEEPLY

**Причина.** В объявлении формата описатели полей, разделители полей или основные группы, заключенные в скобки, превышают восьмикратное вложение.

**Действие.** Проверить спецификацию формата.

**PROGRAM OR BLOCK DATA STATEMENT MUST BE FIRST**

**Причина.** Заголовок головного модуля или заголовок блока данных не является первым предложением модуля.

**Действие.** Исправить заголовок.

**R—SCALE FACTOR NOT IN RANGE —127 TO +127**

**Причина.** Масштабный множитель в спецификации формата находится вне диапазона (от —127 до +127).

**Действие.** Исправить масштабный множитель.

**REFERENCE TO INCORRECT TYPE OF LABEL \*\*\*\***

**Причина.** Логически неверное использование ссылки на метку.

**Действие.** Исправить программу.

**REFERENCE TO UNDEFINED STATEMENT LABEL**

**Причина.** Ссылка на метку, неопределенную в данном модуле.

**Действие.** Исправить программу.

**STATEMENT MUST BE UNLABELED**

**Причина.** Объявление внутренней функции имеет метку.

**Действие.** Удалить метку.

**STATEMENT TOO COMPLEX**

**Причина.** Объявление внутренней функции имеет более десяти параметров или является слишком длинным для его обработки.

**Действие.** Уменьшить список формальных параметров до десяти или разбить объявление на два или более.

**SUBROUTINE OR FUNCTION STATEMENT MUST BE FIRST**

**Причина.** Заголовок SUBROUTINE, FUNCTION или BLOCK DATA не является первым в программном модуле.

**Действие.** Исправить модуль.

**SUBSCRIPT OF ARRAY \*\*\*\* NOT IN RANGE**

**Причина.** Индекс массива принимает недопустимое значение.

**Действие.** Исправить индекс.

**SYNTAX ERROR**

**Причина.** Недопустимый формат.

**Действие.** Исправить ошибочное предложение.

**SYNTAX ERROR IN INTEGER OR FLOATING CONSTANT**

**Причина.** Недопустимый формат целой, вещественной, двойной точности или комплексной константы.

**Действие.** Исправить данную константу.

**SYNTAX ERROR IN LABEL LIST**

**Причина.** В списке меток содержится ошибка.

**Действие.** Проверить формат списка меток и убедиться, что каждая метка является меткой выполняемого предложения. Исправить оператор.

#### TARGET MUST BE ARRAY

**Причина.** В операторе ENCODE или DECODE третий параметр в скобках является именем виртуального массива или элемента виртуального массива.

**Действие.** Исправить данный параметр.

#### UNARY OPERATOR HAS TOO MANY OPERANDS

**Причина.** Указано два или более операндов для операции, использующей только один операнд.

**Действие.** Проверить ошибки печати в операторе, пропущенные операции и пропущенные скобки. Исправить оператор.

#### UNLABELED FORMAT STATEMENT

**Причина.** Объявление формата не имеет метки.

**Действие.** Поместить объявление формата.

#### UNRECOGNIZED KEYWORD IN OPEN OR CLOSE STATEMENT

**Причина.** Недопустимое ключевое слово в операторе OPEN или CLOSE.

**Действие.** Проверить, нет ли опечаток в операторе OPEN или CLOSE и исправить их.

#### UNRECOGNIZED VALUE FOR OPEN OR CLOSE KEYWORD

**Причина.** Ключевое слово в операторе OPEN или CLOSE указано с недопустимым значением. Например, DISROSE='SURE'.

**Действие.** Исправить оператор.

#### USAGE OF VARIABLE \*\*\*\* INVALID

**Причина.** Использование переменной недопустимо в данном предложении. Например, недопустимо использование в качестве параметра в объявлении эквивалентности переменной из блока COMMON, массива переменных или формального параметра, или недопустимо размещение формального параметра или внешнего имени в блоке COMMON.

**Действие.** Исправить предложение.

#### VALUE OF CONSTANT NOT IN RANGE

**Причина.** Значение константы превышает допустимый диапазон.

**Действие.** Исправить предложение.

#### VARIABLE \*\*\*\* INVALID IN ADJUSTABLE DIMENSION

**Причина.** В списке формальных параметров, представляющих регулируемые массивы, встречается имя переменной нецелого типа.

Действие. Исправить программу.

**WRONG NUMBER OF OPERANDS FOR BINARY OPERATOR**

Причина. В операторе отсутствует второй необходимый операнд.

Действие. Проверить ошибки печати и исправить оператор.

**WRONG NUMBER OF SUBSCRIPTS FOR ARRAY \*\*\*\***

Причина. Число индексных выражений не соответствует объявленному числу измерений массива.

Действие. Исправить ссылку на массив.

#### **2.1.2.1. Сообщения, предупреждающие об ошибках**

**LOOP ENTRY AT LABEL \*\*\*\***

Причина. Передача управления осуществляется извне в тело цикла, содержащее указанную метку.

Действие. Исправить передачу управления.

**NON-STANDARD STATEMENT ORDERING**

Причина. Нарушен порядок следования предложений.

Действие. Исправить порядок следования предложений.

**POSSIBLE MODIFICATION OF \*\*\*\***

Причина. Недопустимая модификация переменной. Например, управляющая переменная изменяется операторами внутри цикла или осуществляется передача управления на метку внутри тела цикла из оператора, находящегося вне тела цикла.

Действие. Исправить программу.

**VARIABLE \*\*\*\* IS NOT WORD ALIGNED**

Причина. Переменная или массив, которые не относятся к типу LOGICAL\*1, помещены в объявлении эквивалентности с переменными или массивами типа LOGICAL\*1 или в блоке COMMON после переменной или массива типа LOGICAL\*1.

Действие. Выровнять переменную или массив по границе слова.

**VARIABLE \*\*\*\* NAME EXCEEDS SIX CHARACTERS**

Причина. Имя переменной \*\*\*\* превышает шесть символов.

Действие. Исправить имя.

#### **2.1.2.2. Сообщения о неустранимых ошибках**

**?FORTRAN-F-CODE GENERATION STACK OVERFLOW**

Причина. Переполнение стека. Оператор программы сложен для обработки.

Действие. Упростить сложный оператор.

**?FORTRAN-F-COMPILER FATAL ERROR, ANALYSIS FOLLOWS**

Причина. Произошел сбой транслятора. Далее следует краткое сообщение о характере сбоя.

Действие. Если сообщение содержит рекомендации по устранению сбоя, следует воспользоваться ими.

?FORTRAN—F—CONSTANT SUBSCRIPT STACK OVERFLOW

Причина. Программа имеет оператор с большим количеством индексных констант.

Действие. Упростить оператор.

?FORTRAN—F—DEVICE FULL

Причина. На томе, используемом для вывода объектного файла или файла листинга, недостаточно свободного места или полностью заполнен справочник.

Действие. Освободить место на томе или использовать другой том.

?FORTRAN—F—DYNAMIC MEMORY OVERFLOW

Причина. Недостаточно оперативной памяти для трансляции программного модуля.

Действие. Разбить программный модуль на подпрограммы или транслировать существующий модуль на ЭВМ с большей памятью.

?FORTRAN—F—ERROR READING SOURCE FILE

Причина. Ошибка при считывании входного файла.

Действие. Проверить готовность и исправность оборудования. Повторить трансляцию.

?FORTRAN—F—ERROR WRITING LISTING FILE

Причина. Ошибка при записи файла листинга.

Действие. Проверить готовность и исправность оборудования. Проверить диск на плохие блоки. Повторить трансляцию.

?FORTRAN—F—ERROR WRITING OBJECT FILE

Причина. Ошибка при записи объектного файла.

Действие. Проверить готовность и исправность оборудования. Проверить диск на плохие блоки. Повторить трансляцию.

?FORTRAN—F—FILE NOT FOUND

Причина. Входной файл, указанный в командной строке, не найден.

Действие. Проверить, существует ли файл с указанным именем. Ввести правильную командную строку.

?FORTRAN—F—HELP NOT FOUND

Причина. На томе отсутствует файл FORTRA.HLP.

Действие. Скопировать файл FORTRA.HLP с дистрибутивного тома, если требуется вспомогательная информация.

?FORTRAN—F—ILLEGAL VALUE FOR /X SWITCH

Причина. В командной строке указан недопустимый параметр переключателя /X.

Действие. Ввести правильную командную строку.

?FORTRAN—F—ILLEGAL COMMAND

Причина. Недопустимый формат командной строки.

Действие. Ввести правильную командную строку.

?FORTRAN—F—ILLEGAL DEVICE

Причина. В командной строке обнаружено недопустимое имя устройства.

Действие. Ввести правильную командную строку.

?FORTRAN—F—OPTIMIZER STACK OVERFLOW

Причина. Оператор программы сложен для обработки или в программном модуле много общих подвыражений.

Действие. Упростить сложные операторы.

?FORTRAN—F—SUBEXPRESSION STACK OVERFLOW

Причина. При трансляции обнаружен оператор, который может вызвать переполнение стека во время выполнения программы.

Действие. Упростить сложный оператор.

?FORTRAN—F—UNKNOWN SWITCH — /X

Причина. В командной строке обнаружен недопустимый переключатель.

Действие. Ввести правильную командную строку.

## 2.2. Сообщения библиотеки ФОРТРАН

Сообщения об ошибках выводятся на терминал во время выполнения программы в одной из двух форм: полной или краткой.

Полная форма сообщения:

?ERR NN текст

Краткая форма сообщения:

?ERR NN

где NN — номер ошибки;  
текст — краткое описание ошибки.

По умолчанию выдается полная форма сообщения. Сообщение в краткой форме выдается в том случае, если программа пользователя связана с модулем \$SHORT библиотеки ФОРТРАНа (по переключателю/I в командной строке редактора связей). При использовании модуля \$SHORT библиотеки ФОРТРАНа сохраняется 850 слов памяти. В файле на диске в формате .SAV или .REL из-за этого сохраняется 3—4 блока.

Ошибки, обнаруживаемые библиотекой ФОРТРАНа, подразделяются на четыре вида:

IG — ошибка фиксируется, но сообщение об ошибке не

выводится на терминал. Выполнение программы пользователя продолжается;

W — сообщение об ошибке выводится на терминал, выполнение программы пользователя продолжается;

F — сообщение об ошибке выводится на терминал. Выполнение программы пользователя прекращается и управление передается монитору;

S:N — сообщение об ошибке выводится на терминал. Выполнение программы пользователя продолжается до N-го появления этой ошибки. В последнем случае эта ошибка считается неустранимой (F).

По подпрограмме SETERR библиотеки ФОРТРАНа можно изменить реакцию библиотеки на ошибки с номерами 1—16. Реакцию библиотеки на ошибки с номерами 0 и 20—68 изменять не рекомендуется во избежании непредвиденных ситуаций.

Если выполнение программы прервано вследствие неустранимой ошибки, рекомендуется использовать команду монитора CLOSE для закрытия открытых файлов, при этом часть информации, которая должна быть записана в файл, может быть потеряна.

Ниже приведены сообщения об ошибках с указанием вида ошибки и кратким пояснением.

0 NON—FORTRAN ERROR CALL

F

Причина. Эта ошибка может возникнуть в следующих случаях:

- 1) для основного задания с использованием подпрограммы завершения, которая вызывается подпрограммой системной библиотеки (SYSLIB), не было выделено достаточно места по команде FRUN/N:M для вызова программы завершения;
- 2) недостаточно памяти для фонового задания;
- 3) при работе с монитором SJ одна подпрограмма завершения прервала другую подпрограмму завершения;
- 4) при выполнении подпрограммы на языке АССЕМБЛЕР, связанной с программой на ФОРТРАНе, возникло прерывание по TRAP с неопознанным кодом ошибки.

Действие. 1) Определить объем памяти, требуемый для основного задания и подпрограммы завершения;  
2) увеличить размер свободной области памяти на томе: стереть или переписать на другой том

файлы, в которых нет необходимости, и сжать том;  
3) использовать монитор FB для разрешения одно-  
временного выполнения более одной подпрограм-  
мы завершения;  
4) проанализировать программу на языке АССЕМ-  
БЛЕР

1 INTEGER OVERFLOW

F

Причина. Результат операции умножения, деления или воз-  
ведения в степень целого числа превысил значение  
32767.

Действие. Исправьте программу. Используйте вещественные  
числа.

2 INTEGER ZERO DIVIDE

F

Причина. Деление на ноль при выполнении операции с це-  
лыми числами.

Действие. Исправьте программу.

3 COMPILER GENERATED ERROR

F

Причина. Исходный оператор не протранслирован полностью  
из-за ошибки.

Действие. Просмотрите листинг программы и исправьте  
ошибки, обнаруженные во время трансляции.

4 COMPUTED GOTO OUT OF RANGE

W

Причина. Значение переменной типа целый в операторе пе-  
рехода по предписанию не совпадает со значением  
ни одной из меток списка или значение выражения  
в вычисляемом операторе перехода меньше едини-  
цы или больше числа меток списка.

Действие. Исправьте оператор.

5 INPUT CONVERSION ERROR

C:3

Причина. При вводе форматной записи обнаружен неверный  
символ.

Действие. Символу присваивается значение пусто. Исправьте  
неправильную запись.

6 OUTPUT CONVERSION ERROR

IG

Причина. При выводе форматной записи число символов в  
представлении числа превышает указанную шири-  
ну поля.

Действие. Поле вывода заполняется звездочками. Исправьте объявление формата.

10 FLOATING OVERFLOW

C:3

Причина. Результат операции превышает наибольшее допустимое вещественное число.

Действие. Результату присваивается нулевое значение. Исправьте программу.

11 FLOATING UNDERFLOW

IG

Причина. Результат операции меньше наименьшего допустимого вещественного числа.

Действие. Результату присваивается нулевое значение. Исправьте программу.

12 FLOATING ZERO DIVIDE

F

Причина. Деление на ноль при выполнении операции с вещественными числами.

Действие. Результату присваивается нулевое значение. Исправьте программу.

13 SQRT OF NEGATIVE NUMBER

C:3

Причина. Попытка вычислить квадратный корень из отрицательного числа.

Действие. Результату присваивается нулевое значение. Исправьте программу.

14 UNDEFINED EXPONENTIATION OPERATION

F

Причина. Попытка выполнить недействительную операцию возведения в степень.

Действие. Результату присваивается нулевое значение. Исправьте программу.

15 LOG OF ZERO OR NEGATIVE NUMBER

F

Причина. Попытка вычислить логарифм отрицательного числа или нуля.

Действие. Результату присваивается нулевое значение. Исправьте программу.

16 WRONG NUMBER OF ARGUMENTS

F

Причина. Основная внешняя функция из библиотеки ФОРТРАНа или подпрограмма из библиотеки системных подпрограмм вызвана с неправильным числом параметров.

- Действие. Исправьте формат вызова функции или подпрограммы.
- 20 INVALID LOGICAL UNIT NUMBER
- F
- Причина. Номер логического устройства отличен от 1—99.
- Действие. Исправьте оператор.
- 21 OUT OF AVAILABLE LOGICAL UNITS
- F
- Причина. Попытка одновременно открыть логических устройств больше допустимого числа (по умолчанию максимальное число — 6).
- Действие. Вновь транслируйте исходную программу с переключателем /UNITS:M (N:M).
- 22 INPUT RECORD TOO LONG
- F
- Причина. Во время операции ввода размер записи превысил максимально допустимую длину (по умолчанию максимальная длина записи 136 байт).
- Действие. Вновь транслируйте исходную программу с переключателем /RECORD:M (/R:M).
- 23 HARDWARE I/O ERROR
- F
- Причина. Аппаратная ошибка во время операции ввода-вывода.
- Действие. Проверьте готовность и исправность оборудования. Проверьте диск на плохие блоки. Повторите операцию.
- 24 ATTEMPT TO READ/WRITE PAST END OF FILE
- F
- Причина. 1) Чтение после последней записи файла последовательного доступа;  
2) ссылка на номер несуществующей записи файла прямого доступа;  
3) на томе недостаточно места для записи файла.
- Действие. 1) Используйте параметр «END=» в операторе ввода последовательного доступа;  
2) исправьте программную логику в операторе ввода прямого доступа;  
3) освободите место на томе или используйте другой том для операции записи.
- 25 ATTEMPT TO READ AFTER WRITE
- F
- Причина. Попытка считать выходной файл непосредственно после его записи на устройство последовательного доступа.

Действие. Исправьте программу. Перед считыванием файла с устройства последовательного доступа необходимо выполнить операцию REWIND или BACKSPACE.

26 RECURSIVE I/O NOT ALLOWED

F

Причина. Элементом списка вывода оператора WRITE является выражение, содержащее ссылку на внешнюю функцию, которая сама выполняет операцию ввода-вывода.

Действие. Исправьте программу.

27 ATTEMPT TO USE DEVICE NOT IN SYSTEM

F

Причина. Попытка использовать устройство, которое не определено в системе.

Действие. Присвойте с помощью команды монитора ASSIGN имя требуемого логического устройства или исправьте ошибку в операторе.

28 OPEN FAILED FOR FILE

F

Причина. 1) Заданный файл не найден;

2) на томе нет места;

3) выбранный канал уже использован.

Действие. 1) Проверьте наличие заданного файла;

2) освободите место на томе (сотрите ненужные файлы и сожмите том) или используйте другой том;

3) используйте другой канал.

29 NO ROOM FOR DEVICE HANDLER ФОДОС

F

Причина. Недостаточно места в памяти для драйвера заданного устройства.

Действие. Перепишите файл на системное устройство или устройство, драйвер которого является резидентным. Освободите часть оперативной памяти (удалите ненужные драйверы, снимите основное задание, используйте монитор SJ).

30 NO ROOM FOR BUFFERS

F

Причина. Недостаточно места в памяти для размещения буферов ввода-вывода.

Действие. 1) Уменьшите число логических устройств, которые одновременно открыты в момент обнаружения ошибки;

- 2) используйте один буфер вместо двух;
- 3) освободите часть оперативной памяти.

31 NO AVAILABLE I/O CHANNEL  
F

Причина. Запрошено для ввода-вывода одновременно больше каналов, чем доступно в системе (15, исключая терминал).

Действие. Освободите каналы, которые не используются в данный момент.

32 FMTD—UNFMTD—RANDOM I/O TO SAME FILE  
F

Причина. Комбинация форматного и бесформатного ввода-вывода для одного и того же файла.

Действие. Исправьте программу.

33 ATTEMPT TO READ PAST END OF RECORD  
F

Причина. Попытка считать запись большей длины, чем фактически имеется в файле

Действие. Проверьте длину записи в файле. Исправьте программу.

34 UNFMTD I/O TO TT OR LP  
F

Причина. Попытка вывести бесформатную запись на терминал или печатающее устройство.

Действие. Назначьте другому логическому устройству, позволяющему вывод бесформатных записей, номер рассматриваемого устройства, используя команду монитора ASSIGN, подпрограмму ASSIGN библиотеки ФОРТРАНа или IASIGN библиотеки системных подпрограмм.

35 ATTEMPT TO OUTPUT TO READ ONLY FILE  
F

Причина. Попытка записать файл, открытый для чтения.

Действие. Проверьте задание параметров при вызове подпрограмм ASSIGN библиотеки ФОРТРАНа или IASIGN библиотеки системных подпрограмм. Проверьте программу.

36 BAD FILE SPECIFICATION STRING  
F

Причина. Не может быть обработан параметр в подпрограмме ASSIGN библиотеки ФОРТРАНа, указывающий на спецификацию файла.

Действие. Проверьте задание соответствующего параметра в подпрограмме ASSIGN.

37 RANDOM ACCESS READ/WRITE BEFORE DEFINE FILE

F

Причина. Попытка выполнить операцию чтения или записи прямого доступа до выполнения оператора DEFINE FILE.

Действие. Исправьте программу.

38 RANDOM I/O NOT ALLOWED ON TT OR LP

F

Причина. Попытка выполнить операцию ввода-вывода прямого доступа на терминале или печатающем устройстве.

Действие. Назначьте другому логическому устройству, позволяющему операции ввода-вывода прямого доступа, номер рассматриваемого устройства, используя команду монитора ASSIGN, подпрограмму ASSIGN библиотеки ФОРТРАНа или IASIGN библиотеки системных подпрограмм.

39 RECORD LARGER THAN RECORD SIZE IN DEFINE FILE

F

Причина. Длина записи больше указанной в операторе DEFINE FILE.

Действие. Уменьшите число элементов в списке ввода-вывода или переопределите длину записи.

40 REQUEST FOR A BLOCK LARGER THAN 65535

F

Причина. Попытка обратиться к блоку устройства, абсолютный адрес которого превышает 65535.

Действие. Исправьте программу.

41 DEFINE FILE ATTEMPTED ON AN OPEN UNIT

F

Причина. Попытка открыть файл с помощью оператора DEFINE FILE на устройстве, на котором уже открыт файл.

Действие. Закройте открытый файл, используя подпрограмму CLOSE библиотеки ФОРТРАНа, перед использованием DEFINE FILE.

42 MEMORY OVERFLOW COMPILING OBJECT TIME FORMAT

F

Причина. Переполнение памяти при задании спецификации формата в массиве.

Действие. Используйте объявление формата для операторов ввода-вывода. Освободите часть оперативной памяти.

43 SYNTAX ERROR IN OBJECT TIME FORMAT

F

- Причина. Синтаксическая ошибка в спецификации формата, заданной в массиве.
- Действие. Исправьте спецификацию формата.
- 44 2ND RECORD REQUEST IN ENCODE/DECODE  
F
- Причина. Использование оператора ENCODE или DECODE более чем для одной записи.
- Действие. Исправьте объявление формата, связанное с ENCODE или DECODE (объявление может относиться только к одной записи), проверьте наличие символа «/» в объявлении формата.
- 45 INCOMPATIBLE VARIABLE AND FORMAT TYPES  
F
- Причина. Попытка вывести вещественные данные в формате целого числа или целые данные в формате вещественного числа.
- Действие. Согласуйте описатели полей в объявлении формата с типом соответствующей переменной.
- 46 INFINITE FORMAT LOOP  
F
- Причина. Объявление формата не содержит описателей полей, используемых при передаче переменных в операторе ввода-вывода.
- Действие. Исправьте объявление формата.
- 47 ATTEMPT TO STORE OUTSIDE PARTITION ФОДОС  
F
- Причина. Попытка записать элемент массива в ячейку, адрес которой находится за пределами памяти, отведенной для данного модуля.
- Действие. Исправьте программу.
- 48 UNIT ALREADY OPEN  
F
- Причина. Попытка выполнить операцию, недопустимую для открытого файла.
- Действие. Закройте файл, используя подпрограмму CLOSE библиотеки ФОРТРАНа или оператор CLOSE, перед выполнением операции.
- 49 ENDFILE ON RANDOM FILE  
F
- Причина. Оператор ENDFILE содержит номер логического устройства, которое открыто для файла прямого доступа.
- Действие. Исправьте программу.

50 KEYWORD VALUE ERROR IN OPEN STATEMENT

F

Причина. В операторе OPEN ключевому слову присвоено недопустимое значение.

Действие. Исправьте оператор OPEN.

51 INCONSISTENT OPEN/CLOSE STATEMENT SPECIFICATIONS

F

Причина. В операторе OPEN или CLOSE файл типа 'NEW' или 'SCRATCH' указан вместе с ключевым словом READONLY или ключевое слово READONLY используется вместе со значением ключевого слова DISP='DELETE'.

Действие. Исправьте оператор.

52 ATTEMPT TO DELETE A PROTECTED FILE

W

Причина. Указание ключевого слова 'DISP='DELETE' в операторе OPEN или CLOSE для защищенного файла.

Действие. Ликвидируйте защиту файла или исправьте оператор.

53 LIST-DIRECTED I/O SYNTAX ERROR

W

Причина. Синтаксическая ошибка в операторе ввода-вывода с преобразованием по списку.

Действие. Исправьте оператор.

59 USR NOT LOCKED ФОДОС

F

Причина. Не была подана команда монитора SET USR NOSWAP перед запуском программы в основном режиме, хотя при трансляции программы использовался переключатель /NOSWAP (/U).

Действие. Проанализируйте действие переключателя /NOSWAP и транслируйте программу либо без переключателя, либо подайте команду SET USR NOSWAP перед запуском программы.

60 STACK OVERFLOWED

F

Причина. Это сообщение появляется при работе в фоновом режиме. Переполнение стека, что может привести к нарушению выхода из подпрограммы или нарушению выполнения операции открытия файла и трассировки.

Действие. Выделите дополнительную область для стека, ис-

пользуя переключатель /B:M при связывании объектных модулей. Проверьте программу.

61 ILLEGAL MEMORY REFERENCE

F

Причина. Обращение по несуществующему адресу памяти.  
Действие. Если ошибка появилась в программе на языке АС-СЕМБЛЕР, необходимо исправить исходную программу. Если ошибка появилась в программе на языке ФОРТРАН, следует убедиться, что библиотека ФОРТРАНа соответствует конфигурации технических средств.

62 FORTRAN START FAIL

F

Причина. Программа загружена в память, но не осталось свободной памяти для инициализации рабочей области и буферов библиотеки ФОРТРАНа.

Действие. При работе в фоновом режиме освободите часть оперативной памяти. При работе в основном задании используйте команду FRUN/N:M.

63 ILLEGAL INSTRUCTION

F

Причина. Попытка выполнить резервную операцию (например, операцию с плавающей запятой на ЭВМ, не имеющей аппаратных средств для работы с плавающей запятой).

Действие. Если ошибка появилась в программе на языке АССЕМБЛЕРА необходимо исправить исходную программу. Если ошибка появилась в программе на языке ФОРТРАН, следует убедиться, что библиотека ФОРТРАНа соответствует конфигурации технических средств.

64 VIRTUAL ARRAY INITIALIZATION FAILURE

F

Причина. Транслятор не может инициализировать виртуальный массив. Ошибка может возникнуть в следующих случаях:

- 1) требования памяти для виртуальных массивов превышают имеющуюся оперативную память;
- 2) неправильно использована библиотека ФОРТРАНа. Программа может использовать виртуальные массивы при работе под управлением монитора SJ или FB при указании соответствующей библиотеки или при работе под управлением монитора XM без указания этой библиотеки;

3) отсутствуют технические средства для поддержки области логических адресов (PLAS) при работе под управлением монитора ХМ.

**Действие.** Уменьшите требования к объему виртуальной памяти за счет указания верхних границ с меньшими значениями. Правильно используйте библиотеку ФОРТРАНа.

65 VIRTUAL ARRAY MAPPING ERROR

F

**Причина.** Попытка обратиться за пределы расширенной памяти, отведенной для виртуальных массивов в программе.

**Действие.** Проверьте, чтобы значения индексов элементов виртуальных массивов, к которым происходит обращение, не превышали значения верхних границ.

66 UNSUPPORTED OPEN/CLOSE KEYWORD OR  
F OPTION

**Причина.** В операторе OPEN или CLOSE указано ключевое слово или его значение, которое не поддерживается системой.

**Действие.** Исправьте оператор.

67 UNSUPPORTED OPEN/CLOSE KEYWORD OR  
W OPTION

**Причина.** В операторе OPEN или CLOSE указано ключевое слово или его значение, которое не имеет смысла в данной системе.

**Действие.** Исправьте оператор.

68 DIRECT ACCESS RECORD SIZE ERROR

F

**Причина.** Размер записи в файле прямого доступа превышает 32767 двойных слов.

**Действие.** Исправьте программу.

### 3. СООБЩЕНИЯ ИНТЕРПРЕТАТОРА С ЯЗЫКА БЕЙСИК

Интерпретатор БЕЙСИК/ФОДОС осуществляет синтаксическую проверку введенных предложений и команд. В случае выявления ошибки на терминал выводится соответствующее сообщение.

Сообщения об ошибках БЕЙСИК/ФОДОС имеют следующий формат:

?MMMMMMMM AT LINE XXXXX

или

?LLL AT LINE XXXXX

где MMMMMMMM — полное сообщение;  
LLL — краткое сообщение;  
XXXXX — номер строки, содержащей ошибку.

Формат сообщения задается во время генерации системы БЕЙСИК/ФОДОС.

Если ошибка обнаружена в операторе режима непосредственного исполнения или команде, то AT LINE XXXXX не печатается.

При возникновении неустранимой системной ошибки сообщение об ошибке выдается на системный терминал в формате ?F—LLL

и управление передается монитору системы ФОДОС. В этом случае пользователю необходимо перезагрузить систему.

Ниже приведены ошибки, выявляемые БЕЙСИК/ФОДОС, и даны рекомендации по их устранению. Цифрой указан код ошибки.

?ADR

?ADDRESS CHECK ERROR

Причина. Системная ошибка.

Действие. Если ошибка повторяется, обратитесь к обслуживающему персоналу.

89

?ARG

?ARGUMENT ERROR

Причина. Параметры в вызове функции не соответствуют числу, значению или типу параметров, определенных для данной функции.

Действие. Изменить параметры в соответствии с требованиями данной функции.

44

?ATL

?ARRAYS TOO LARGE

Причина. Недостаточно оперативной памяти для массивов, определенных в объявлении DIM.

Действие. Уменьшите размеры массивов или сократите размер программы. Вместо объявления массивов используйте объявления виртуальных массивов.

26

?ATT

?ATTEMPT TO ASSIGN TOO MANY TERMINALS

Причина. Попытка назначить непривилегированным пользователем большего числа терминалов, чем допускает данная версия БЕЙСИК/ФОДОС.

Действие. Допустимое число терминалов определяется во время генерации системы.

50

?BDR

?BAD DATA READ

Причина. Неправильный тип данных, вводимых из файла или по объявлению DATA.

Действие. Ввести правильно данные.

53

?BLG

?BAD LOG

Причина. Параметр в функции LOG или LOG10 равен нулю или отрицательному числу.

Действие. Устранимая ошибка. Результату присваивается нулевое значение, и выполнение программы продолжается.

101

?BRT

?BAD DATA RETYPE FROM ERROR

Причина. Данные, введенные оператором INPUT или INPUT#, не соответствуют указанному типу.

Действие. Устранимая ошибка. Введите правильно данные и выполнение программы будет продолжаться.

161

?BSO

?BUFFER STORAGE OVERLOW

Причина. Недостаточно памяти для файла в области ввода-вывода пользователя или системы.

Действие. Сократите размер программы.

7

?CAO

?CHANNEL ALREADY OPEN

Причина. В операторе OPEN указан номер канала, который был уже использован.

Действие. Исправьте программу: соедините файл с другим номером канала или удалите предыдущий оператор.

103

?CCP

?CHECKSUM ERROR IN COMPILED PROGRAM

Причина. Ошибка при выполнении команды COMPILE.

Действие. Используйте копию программы, созданную по команде SAVE или REPLACE.

?CHN

?CHANNEL NUMBER FAULT

Причина. Системная ошибка.

Действие. Если ошибка повторяется, обратитесь к обслуживающему персоналу.

13

?CIE

?CHANNEL I/O ERROR

Причина. Ошибка при открытии или закрытии канала.

Действие. Проверьте готовность и исправность оборудования. Проверьте размер файла (файл не должен иметь длину, равную нулю).

121

?COO

?COMMON OUT OF ORDER

Причина. Изменен порядок следования элементов в объявлении COMMON.

Действие. Проверьте порядок следования элементов в списках COMMON для каждого сегмента.

9

?CNO

?CHANNEL NOT OPEN

Причина. В операторе PRINT#,PRINT#USING,INPUT#,IF END# или CLOSE номер канала не соединен с открытым файлом.

Действие. Перед вводом-выводом данных необходимо открыть файл с помощью оператора OPEN.

32

?CPE

?CHANNEL POOL EMPTY

Причина. Все номера каналов для ввода-вывода заняты.

Действие. Повторите операцию.

58

?CVO

?CONTROL VARIABLE OUT OF RANGE

Причина. Выражение в операторе ON GO TO или ON GO SUB имеет значение ноль или имеет большее значение, чем указанное число переходов в операторе.

Действие. Исправьте программу.

?DEV

?NO DEVICE HANDLER

Причина. Не загружен драйвер устройства.

Действие. Если драйвер находился на системном устройстве и был правильно загружен, то при повторении ошибки, обратитесь к обслуживающему персоналу.

14

?DHE

?DEVICE HARDWARE ERROR

Причина. Аппаратная ошибка во время операции ввода-вывода. Причиной может быть включение или выключение защиты записи или во время операции с лентой обнаружен физический конец ленты.

Действие. Проверьте готовность и исправность оборудования. Проверьте диск на плохие блоки. Для магнитной ленты или магнитной ленты кассетного типа запишите логический конец с помощью оператора OPEN с параметром MODE.

?DIR

?DIRECTORY I/O ERROR

Причина. Ошибка при чтении справочника устройства. Ее может вызывать включение защиты записи.

Действие. Если ошибка повторяется и не вызвана включением защиты записи, сообщите об этом администратору системы.

?DRO

?DIRECTORY OVERFLOW

Причина. В справочнике устройства нет места для новой записи файла.

Действие. Удалите все старые файлы с этого устройства или используйте другое устройство.

8

?DNA

?DEVICE NOT AVAILABLE

Причина. Попытка назначить личным общее или уже занятое устройство.

Действие. Используйте другое устройство.

21

?DNE

?DEVICE NOT ENABLED

Причина. Устройство неправильно подключено или находится в автономном режиме.

Действие. Правильно подключите устройство или используйте другое устройство.

61

?DVO

?DIVISION BY ZERO

Причина. Попытка деления на ноль.

Действие. Устраняемая ошибка. Результату присваивается

нулевое значение и выполнение программы продолжается.

49

?EER

?EXPONENTIATION ERROR

**Причина.** Попытка возвести отрицательное число в нецелую степень или в целую степень, абсолютное значение которой больше 255. При вычислении формируется комплексное число, которое не может быть представлено в БЕЙСИК/ФОДОС.

**Действие.** Устраняемая ошибка. Результату присваивается нулевое значение, и выполнение программы продолжается.

59

?EIE

?EXCESS INPUT IGNORED

**Причина.** С терминала введено больше данных, чем указано в операторе INPUT.

**Действие.** Устраняемая ошибка. Лишние данные игнорируются и выполнение программы продолжается.

108

?ENL

?END NOT LAST

**Причина.** За оператором END следуют строки программы.

**Действие.** Исправьте программу.

87

?ETC

?EXPRESSION TOO COMPLEX

**Причина.** Переполнение стека при вычислении сложного выражения.

**Действие.** Упростите выражение путем разбиения его на более короткие.

?F—EMT

**Причина.** Недопустимый код EMT.

?F—OVL

**Причина.** Ошибка при чтении оверлейного сегмента.

?F—SYS

**Причина.** Неустраняемая системная ошибка.

?F—TRP

**Причина.** Прерывание по вектору 4 или 10.

**Действие.** Ошибки F—EMT, F—OVL, F—SYS, F—TRP являются неустраняемыми системными ошибками. Управление передается монитору системы ФОДОС.

Пользователю необходимо перезагрузиться. Если какая-нибудь из этих ошибок повторяется и не вызвана ошибкой в программе на языке АССЕМБЛЕР, обратитесь к обслуживающему персоналу.

83

?FAD

?FUNCTION ALREADY DEFINED

Причина. Многократное определение внутренней функции.  
Действие. Измените имена функций так, чтобы каждая функция определялась только один раз.

?FCH

?ERROR IN FETCH

Причина. Устройство содержит плохой блок или находится в автономном режиме.  
Действие. Если устройство находится не в автономном режиме и не содержит плохого блока, то при повторении ошибки обратитесь к обслуживающему персоналу.

5

?FNF

?FILE NOT FOUND

Причина. Указанный файл не найден на устройстве.  
Действие. Проверьте существование файла на устройстве. Введите правильно оператор или команду.

241

?FOV

?FLOATING OVERFLOW

Причина. Результат операции превышает наибольшее допустимое вещественное число.  
Действие. Устраняемая ошибка. Результату присваивается нулевое значение, и выполнение программы продолжается.

10

?FPV

?FILE PRIVILEGE VIOLATION

Причина. Попытка нарушить защиту файла.  
Действие. Операции с файлом может выполнить только привилегированный пользователь.

91

?FSV

?NESTED FOR STATEMENTS WITH SAME CONTROL VARIABLE

Причина. Использована одна и та же управляющая переменная для вложенных операторов FOR.  
Действие. Используйте различные управляющие переменные во вложенных операторах FOR.

11

?FTS

?FILE TOO SHORT

Причина. Неверно определено значение ключевого слова FILESIZE в операторе OPEN. Файл не закрыт, данные теряются.

Действие. Увеличить значение ключевого слова FILESIZE в операторе OPEN. Если ошибка возникает при выполнении команд SAVE или REPLACE, сохраните программу на магнитной ленте, магнитной ленте кассетного типа или перфоленте и сообщите об этом администратору системы.

242

?FUN

?FLOATING UNDERFLOW

Причина. Результат операции меньше наименьшего допустимого вещественного числа.

Действие. Устраняемая ошибка. Результату присваивается нулевое значение, и выполнение программы продолжается.

92

?FWN

?FOR WITHOUT NEXT

Причина. Оператор FOR указан без соответствующего оператора NEXT, определяющего конец цикла.

Действие. Исправьте программу: каждому оператору FOR должен соответствовать оператор NEXT.

42

?HFE

?HANDLER FETCH ERROR

Причина. Ошибка при загрузке драйвера устройства.

Действие. Системная ошибка. Если она повторяется, обратитесь к обслуживающему персоналу.

46

?INC

?ILLEGAL CHANNEL NUMBER

Причина. Недопустимый номер канала, или оператор IF END указывает на файл терминала.

Действие. Исправьте ошибку.

102

?IDM

?ILLEGAL DIM

Причина. Синтаксическая ошибка в объявлении DIM или COMMON.

Действие. Исправьте ошибку.

29

?IEF

?ILLEGAL END OF FILE IN COMPILED FILE

Причина. Ошибка при выполнении команды COMPILE. Недопустимый конец файла протранслированной программы.

Действие. Используйте копию программы, созданную по команде SAVE или REPLACE.

23

?IFL

?ILLEGAL FILE LENGTH

Причина. Размер выходного файла, определенный в ключевом слове FILESIZE оператора OPEN, превышает максимально допустимый размер, определенный при генерации системы. Ошибка возникает, если в ключевом слове FILESIZE указано значение меньше 1 для непривилегированного пользователя и меньше —1 для привилегированного пользователя.

Действие. Уточните максимально допустимый размер выходного файла и измените значение, указанное в ключевом слове FILESIZE оператора OPEN.

2

?IFS

?ILLEGAL FILE SPECIFICATION

Причина. Синтаксическая ошибка в спецификации файла.

Действие. Исправьте ошибку.

141

?IID

?ILLEGAL I/O DIRECTION

Причина. Попытка записи в файл открытый для чтения, или наоборот.

Действие. Верно определите ключевые слова в операторе OPEN.

119

?IIM

?ILLEGAL IN IMMEDIATE MODE

Причина. Использование оператора INPUT в непосредственном режиме.

Действие. Используйте оператор INPUT в программном режиме.

106

?INS

?ILLEGAL NUMBER OF SUBSCRIPTS

Причина. Использование двумерного массива как одномерного или наоборот.

Действие. Исправьте описание массива в объявлении DIM.

51

?IOV

?INTEGER OVERFLOW

Причина. Результат выполнения операции с целыми числами больше допустимого.

Действие. Устраняемая ошибка. Результату присваивается нулевое значение, и выполнение программы продолжается. Используйте для вычисления вещественные числа.

148

?IRS

?ILLEGAL RECORDSIZE

Причина. Значение ключевого слова RECORDSIZE в операторе OPEN меньше 1 для непривилегированного пользователя равно 0 или меньше -1 для привилегированного пользователя.

Действие. Значение ключевого слова RECORDSIZE должно быть больше нуля; привилегированные пользователи могут использовать значение 1.

96

?ISE

?INPUT STRING ERROR

Причина. В текстовой константе, введенной с помощью оператора INPUT, отсутствуют закрывающие кавычки.

Действие. Устраняемая ошибка. Текстовой константе присваивается значение от открывающих кавычек до ограничителя строки, и выполнение программы продолжается.

24

?IUN

?ILLEGAL USER NUMBER

Причина. Недопустимый формат идентификатора пользователя в функции SYS(2), или использован идентификатор другого пользователя.

Действие. Правильно введите идентификатор пользователя.

47

?LTL

?LINE TOO LONG

Причина. Попытка ввести строку, содержащую более 132 символов. Строка игнорируется.

Действие. Разбейте строку на несколько строк.

66

?MSP

?MISSING SUBPROGRAM

Причина. В операторе CALL указано имя несуществующей подпрограммы.

Действие. Проверьте, что все вызываемые подпрограммы существуют и их имена введены правильно.

25

?MTE

?MULTI—TERMINAL ERROR

Причина. Ошибка при использовании мультитерминального драйвера.

Действие. Системная ошибка. Обратитесь к обслуживающему персоналу.

4

?NER

?NOT ENOUGH ROOM

Причина. На устройстве недостаточно места для размещения выходного файла или предпринята попытка прочитать одновременно на магнитной ленте более одного файла.

Действие. Измените значение выражения в ключевом слове FILESIZE оператора OPEN или укажите другое устройство, если ошибка возникает при обращении к устройству общего доступа.

30

?NFS

?NOT FILE STRUCTURED

Причина. Попытка открыть файл прямого доступа на устройстве нефайловой структуры.

Действие. В операторе OPEN укажите устройство файловой структуры.

54

?NGS

?NEGATIVE SQUARE ROOT

Причина. Попытка вычислить квадратный корень отрицательного числа.

Действие. Устраняемая ошибка. Результату присваивается нулевое значение, и выполнение программы продолжается.

81

?NRC

?NO ROOM FOR CALL

Причина. Недостаточно оперативной памяти для выполнения оператора CALL.

Действие. Сократите размер программы.

35

?NRH

?NO ROOM FOR HANDLER

Причина. Недостаточно оперативной памяти в системной области ввода-вывода для размещения драйвера указанного устройства.

Действие. Используйте другое устройство или загрузите драйвер еще раз.

136

?NSF

?NOT A SEQUENTIAL FILE

Причина. Попытка использовать оператор IF END, INPUT, LINPUT или PRINT для файла прямого доступа.

Действие. Исправьте программу.

88

?NSM

?NUMBERS AND STRINGS MIXED

Причина. Попытка выполнить совместно действия с числовыми и текстовыми данными.

Действие. Убедитесь, что числовые данные присваиваются числовым переменным, а текстовые данные — текстовым переменным.

6

?NVD

?NOT A VALID DEVICE

Причина. Указано недопустимое имя устройства или устройство недоступно пользователю системы БЕЙСИК/ФОДОС.

Действие. Введите правильно имя устройства.

93

?NWF

?NEXT WITHOUT FOR

Причина. Программа содержит оператор NEXT без соответствующего оператора FOR или предпринята попытка передать управление внутрь цикла.

Действие. Убедитесь, что каждый цикл начинается оператором FOR и завершается оператором NEXT. Вход в цикл FOR—NEXT осуществляется только с помощью оператора FOR.

57

?OOD

?OUT OF DATA

Причина. Оператор READ запрашивает данные, в то время как список данных уже исчерпан.

Действие. Введите дополнительные данные или используйте оператор RESTORE.

118

?PRU

?PRINT USING ERROR

Причина. Недопустимый формат оператора PRINT USING.

Действие. Исправьте формат оператора.

18

?PSF

?PRIVILEGED SYSTEM FUNCTION

Причина. Попытка непривилегированным пользователем вызвать привилегированную системную функцию.

Действие. Вызов системной функции отменяется.

111

?PTB

?PROGRAM TOO BIG

Причина. Размер программы превышает объем оперативной памяти, отведенной для пользователя.

Действие. Сократите размер программы.

110

?RES

?RESEQUENCE ERROR

Причина. Использован недопустимый параметр в команде RESEQ.

Действие. Исправьте команду.

115

?RPL

?USE REPLACE

Причина. Попытка записать на устройство файловой структуры файл с именем, уже существующим на данном устройстве. Операция не выполняется.

Действие. Используйте команду REPLACE для записи нового файла.

72

?RWG

?RETURN WITHOUT GO SUB OR RESUME WITHOUT ERROR

Причина. Оператор RETURN встречается до выполнения оператора GO SUB или оператор RESUME — до возникновения ошибки.

Действие. Исправьте программу. Каждая подпрограмма должна завершаться оператором RETURN и оператор RESUME должен следовать за условием возникновения ошибки.

55

?SOB

?SUBSCRIPT OUT OF BOUNDS

Причина. Значение индекса меньше 0, больше 32767 или превышает верхнюю границу в объявлении DIM.

Действие. Правильно используйте индексы массива.

126

?SSO

?STRING STORAGE OVERFLOW

Причина. Недостаточно оперативной памяти для хранения программы.

Действие. Сократите размер программы.

99

?STL

?STRING TOO LONG

Причина. Число символов в текстовой константе больше 255.

Действие. Разбейте текстовую константу на несколько.

109

?SUB

?SUBSTITUTE ERROR

Причина. Не указан символ-ограничитель в команде SUB.

Действие. Исправьте команду.

98

?SYN

?SYNTAX ERROR

Причина. Синтаксическая ошибка.

Действие. Исправьте в программе оператор или объявление, или введенную команду.

100

?TIC

?TOO MANY ITEMS IN COMMON

Причина. Число данных в объявлении COMMON превышает 255.

Действие. Уменьшите число данных в объединении COMMON.

86

?TLT

?LINE TOO LONG TO TRANSLATE

Причина. Введенная строка превышает область памяти, отведенную для ее трансляции. Строка игнорируется.

Действие. Разбейте строку на две строки.

17

?TMC

?TOO MANY CHANNELS

Причина. Число файлов, которое может быть одновременно открыто непривилегированным пользователем с помощью оператора OPEN, больше допустимого.  
Действие. Закройте открытый файл. Вновь выполните операцию.

36

?TMG

?TOO MANY GO SUB

Причина. Число вложенных операторов GO SUB превысило 20.

Действие. Ограничьте вложенность операторов GO SUB до 20 уровней.

27

?TNA

?TERMINAL NOT ASSIGNED

Причина. Попытка использовать терминал, неназначенный личным.

Действие. Назначьте терминал личным по команде ASSIGN.

97

?UAC

?UNDIMENSIONED ARRAY IN CALL

Причина. Массив, указанный в операторе CALL, не определен в объявлении DIM.

Действие. Определите массив в объявлении DIM.

74

?UFN

?UNDEFINED FUNCTION

Причина. Не определена внутренняя функция.

Действие. Внутреннюю функцию можно определить только после выполнения команды RUN или оператора CHAIN.

243

?ULN

?UNDEFINED LINE NUMBER

Причина. Ссылка на неопределенный номер строки.

Действие. Исправьте программу.

?USR

?ILLEGAL ARRAY CHANNEL ALREADY IN USE

Причина. Системная ошибка.

Действие. Проверьте программу. Если ошибка повторяется, обратитесь к обслуживающему персоналу.

43

?VCU

?VIRTUAL ARRAY CHANNEL ALREADY IN USE

**Причина.** Номер канала, указанный в объявлении виртуального массива, был уже указан в объявлении.  
**Действие.** Исправьте программу. Используйте различные номера каналов.

#### **4. СООБЩЕНИЯ ОПЕРАЦИОННОЙ СИСТЕМЫ ФОДОС-2**

Ниже приведены сообщения, выдаваемые операционной системой ФОДОС-2. Формат сообщений описан в разделе 1.

?ВА—U—BC

**Причина.** В управляющем файле обнаружена ошибка (наиболее вероятно — после редактирования файла).

**Действие.** Проверить управляющий файл. Получить его повторно.

?ВА—U—FE

**Причина.** Оператор подал директиву \F.

**Действие.** Не требуется.

?ВА—U—IO

**Причина.** Ошибка чтения управляющего файла или записи в регистрационный файл (наиболее вероятно — переполнение регистрационного файла).

**Действие.** Увеличить размер регистрационного файла с помощью конструкции [:N].

?ВА—U—LU

**Причина.** Программа WATCH не смогла найти свободный канал для использования. (Это возможно, если все 16 (десятичное) каналов открыты).

**Действие.** Проверить программу и убедиться, что канал оставлен открытым и не используется.

?WATCH—F—ABORT JOB

**Причина.** Ошибка во время транслирования программой WATCH файла команд.

**Действие.** Распечатать файл и устранить ошибки.

?WATCH—F—AMBIGUOUS COMMAND

**Причина.** Неоднозначное определение команды.

**Действие.** Определить команду однозначно.

?WATCH—F—AMBIGUOUS OPTION

**Причина.** Неоднозначное определение переключателя.

**Действие.** Определить переключатель однозначно.

?WATCH—F—WATCH FATAL ERROR

**Причина.** Аппаратная ошибка во время работы WATCH.

**Действие.** Проверить исправность и правильность включения аппаратуры. Проверить диск на плохие блоки.

- Снять защиту записи.  
**?WATCH—F—WATCH HANDLER NOT RESIDENT**  
 Причина. Драйвер BA.SYS отсутствует в памяти.  
 Действие. Загрузить драйвер по команде LOAD.
- ?WATCH—F—STACK OVERFLOW**  
 Причина. Недопустимая глубина вложения команд \$CALL в пакете заданий.  
 Действие. Проверить пакет заданий.
- ?WATCH—F—CHANNEL IN USE**  
 Причина. Сбой в работе операционной системы.  
 Действие. Перезагрузить WATCH.
- ?WATCH—F—DISMOUNT ERROR**  
 Причина. Указанное логическое имя устройства не существует.  
 Действие. Присвоить устройству логическое имя.
- ?WATCH—F—DUPLICATE OPTION**  
 Причина. В управляющем операторе несколько раз указан один и тот же переключатель.  
 Действие. Исправить пакет заданий.
- ?WATCH—F—EOF WITH NO \$EOJ**  
 Причина. Файл не ограничен командой \$EOJ.  
 Действие. Проверить задание.
- ?WATCH—F—FILE NOT FOUND**  
 Причина. Входной файл не найден или для входного файла, который не является управляющим, определен переключатель /X.  
 Действие. Проверить введенную командную строку.
- ?WATCH—F—INPUT ERROR**  
 Причина. Аппаратная ошибка чтения файла команд (.BAT).  
 Действие. Проверить исправность и правильность включения аппаратуры. Проверить диск на плохие блоки.
- ?WATCH—F—INPUT FILE**  
 Причина. В командной строке не указан входной файл.  
 Действие. Ввести правильную командную строку.
- ?WATCH—F—INSUFFICIENT MEMORY**  
 Причина. Программе WATCH недостаточно памяти для загрузки драйвера устройства, необходимого для операции ввода-вывода или для выполнения команд \$MOUNT или \$DISMOUNT в пакете заданий.  
 Действие. Проверить командную строку или команды \$MOUNT или \$DISMOUNT в пакете заданий.
- ?WATCH—F—INVALID «+»**  
 Причина. Неправильное использование символа «+», напри-

мер, в команде \$RUN, или в выходной спецификации.

Действие. Проверить задание (командную строку).

?WATCH—F—INVALID CHARACTER

Причина. Указан недопустимый символ кода КОИ-7.

Действие. Распечатать регистрационный файл для обнаружения ошибки.

?WATCH—F—INVALID COMMAND LINE

Причина. В командной строке обнаружена ошибка.

Действие. Ввести правильную командную строку.

?WATCH—F—INVALID CONSTRUCTION

Причина. Неправильный формат оператора IF; недопустимая команда 'текст'.

Действие. Проверить задание.

?WATCH—F—INVALID COPY OF HANDLER

Причина. Копия драйвера BA.SYS в оперативной памяти имеет ошибку.

Действие. Выгрузить драйвер и загрузить его повторно.

?WATCH—F—INVALID DEVICE

Причина. В командной строке указано несуществующее или недопустимое устройство.

Действие. Проверить введенную командную строку. Использовать другое устройство.

?WATCH—F—INVALID LOG DEVICE

Причина. Указано недопустимое регистрационное устройство (MT: или PC:).

Действие. Проверить введенную командную строку, присвоить имя LOG другому устройству.

?WATCH—F—INVALID OPTION

Причина. В командной строке обнаружен недопустимый переключатель.

Действие. Ввести правильную командную строку.

?WATCH—F—INVALID OPTION COMBINATION

Причина. В управляющем операторе обнаружена недопустимая комбинация переключателей.

Действие. Ввести правильную командную строку.

?WATCH—F—INVALID SEQUENCE ARGUMENT

Причина. Идентифицирующий номер в команде \$SEQUENCE не является числовым.

Действие. Исправить команду.

?WATCH—F—INVALID VARIABLE

Причина. Указанная переменная не является буквой (A—Z).

Действие. Проверить задание.

?WATCH—F—INVALID VID

- Причина. Идентификатор тома в команде \$MOUNT определен неправильно.
- Действие. Исправить команду.  
?WATCH—F—LINE TOO LONG
- Причина. Длина командной строки более 80 символов.
- Действие. Ввести правильную командную строку.  
?WATCH—F—LOG DEVICE ERROR
- Причина. Аппаратная ошибка при выводе на регистрационное устройство.
- Действие. Проверить исправность и правильность включения аппаратуры. Проверить диск на плохие блоки. Отменить защиту записи.  
?WATCH—F—«\$» MISSING
- Причина. В первой позиции командной строки отсутствует символ «\$».
- Действие. Проверить задание.  
?WATCH—F—NO CONTROL FILE
- Причина. Попытка записи управляющего файла (.CTL) на устройство нефайловой структуры.
- Действие. Исправить командную строку.  
?WATCH—F—NO \$EOJ
- Причина. В пакете заданий командам \$EOJ или \$SEQUENCE не предшествует команда \$EOJ конца предыдущего задания.
- Действие. Исправить пакет заданий.  
?WATCH—F—NO FILE
- Причина. Спецификация файла отсутствует, где она необходима.
- Действие. Исправить пакет заданий.  
?WATCH—F—NO FILE NAME BEFORE «.»
- Причина. В спецификации файла указан тип файла, но отсутствует его имя.
- Действие. Указать правильную спецификацию файла.  
?WATCH—F—NO «.» IN \$LIB
- Причина. В команде \$LIB используется символ «.».
- Действие. Исправить команду.  
?WATCH—F—NO LOGICAL DEVICE
- Причина. В команде \$MOUNT не указано логическое имя устройства.
- Действие. Исправить команду.  
?WATCH—F—NO PHYSICAL DEVICE
- Причина. В команде \$MOUNT не указано физическое имя устройства.
- Действие. Исправить команду.  
?WATCH—F—OUTPUT DEVICE FULL

Причина. На указанном устройстве недостаточно места для управляющего файла (.CTL).

Действие. Сжать том по команде SQUEEZE. Стереть или записать на другой том ненужные файлы. Использовать другой том.

#### ?WATCH—F—OUTPUT ERROR

Причина. Устройство вывода для управляющего файла — магнитная лента, или ошибка при выводе управляющего файла.

Действие. Проверить введенную командную строку. Проверить исправность и правильность включения аппаратуры. Проверить диск на плохие блоки. Отменить защиту записи.

#### ?WATCH—F—OUTPUT FILE NOT OPEN

Причина. Сбой в работе операционной системы. Возможно, что ошибка в WATCH.

Действие. Использовать другую копию WATCH.

#### ?WATCH—F—PLEASE ASSIGN LOG,LST

Причина. Устройство листинга или регистрационное устройство не определено.

Действие. Присвоить устройствам логические имена LOG: и LST: по команде ASSIGN.

#### ?WATCH—F—PLEASE LOAD LOG HANDLER

Причина. Драйвер регистрационного устройства отсутствует в памяти.

Действие. Загрузить драйвер по команде LOAD.

#### ?WATCH—F—PROTECTED FILE ALREADY EXISTS

Причина. Попытка создать файл с именем уже существующего защищенного файла.

Действие. Использовать команду монитора UNPROTECT или переключатель PIP/Z для изменения защиты существующего файла, или указать другое имя для создания нового файла.

#### ?WATCH—F—RETURN FROM CALL ERROR

Причина. Ошибка при возврате из подпрограммы (\$CALL).

Действие. Проверить управляющий файл; если необходимо, получить его повторно. Повторить операцию.

#### ?WATCH—F—SEPARATOR MISSING

Причина. Спецификация файла не ограничена пробелом, символом «+», точкой или возвратом каретки.

Действие. Исправить команду.

#### ?WATCH—F—TOO MANY FILE DESCRIPTORS

Причина. В командной строке определено более 6-ти спецификаций файлов.

- Действие. Исправить команду.  
 ?WATCH—F—TOO MANY OUTPUT FILES
- Причина. В командной строке определено более двух выходных файлов.
- Действие. Ввести правильную командную строку.  
 ?WATCH—F—UNKNOWN COMMAND
- Причина. В задании обнаружена недопустимая команда.
- Действие. Проверить и исправить задание.  
 BE NNNNNN
- Причина. Встретилась незаданная точка разрыва.
- Действие. Устранить точку разрыва.  
 ?BINCOM—E—DEVICE FULL DEV:FILNAM.TYP
- Причина. На устройстве недостаточно места для выходного файла.
- Действие. Увеличить размер свободной области на томе: удалить ненужные файлы или переписать их на другой том, сжать том по команде SQUEEZE.  
 ?BINCOM—E—DEVICE NOT ACTIVE DEV:
- Причина. Операция ввода-вывода невозможна на указанном устройстве или томе; устройство отключено; нет тома на устройстве; запрещена запись; устройство не описано в таблицах монитора.
- Действие. Проверить правильность включения устройства. Установить обслуживание устройства по команде INSTALL.  
 ?BINCOM—E—ERROR READING DIRECTORY DEV:
- Причина. Аппаратная ошибка чтения справочника указанного устройства.
- Действие. Проверить готовность и исправность оборудования. Проверить диск на плохие блоки.  
 ?BINCOM—E—INPUT ERROR DEV:FILNAM.TYP
- Причина. Ошибка чтения входного файла FILNAM.TYP.
- Действие. Проверить готовность и исправность оборудования. Повторить операцию.  
 ?BINCOM—E—INPUT FILE NOT FOUND DEV:FILNAM.TYP
- Причина. Входной файл, указанный в команде, не найден.
- Действие. Проверить, существует ли файл с указанным именем на указанном устройстве. Подать команду правильно.  
 ?BINCOM—E—INSUFFICIENT MEMORY
- Причина. Недостаточно оперативной памяти для выполнения заданной операции.
- Действие. Освободить часть оперативной памяти (удалить ненужные драйверы, удалить основное задание,

использовать монитор одного задания). Повторить выполнение операции.

**?BINCOM—E—INVALID COMMAND**

Причина. В командной строке обнаружена ошибка.

Действие. Ввести правильную командную строку.

**?BINCOM—E—INVALID DIRECTORY DEV:**

Причина. Структура справочника тома устройства не является структурой ФОДОС.

Действие. Использовать справочник тома устройства со структурой ФОДОС.

**?BINCOM—E—INVALID OPTION /X**

Причина. В командной строке обнаружен недопустимый переключатель.

Действие. Ввести правильную командную строку.

**?BINCOM—E—INVALID OPTION /X:VAL**

Причина. Не указан аргумент переключателя, либо указано неверное значение аргумента.

Действие. Ввести правильную командную строку.

**?BINCOM—E—OUTPUT ERROR DEV:FILNAM.TYP**

Причина. Аппаратная ошибка записи выходного файла. На томе, используемом для вывода, нет свободной области для записи выходного файла.

Действие. Проверить готовность и исправность оборудования. Проверить том на плохие блоки. Повторить выполнение операции. Указать в командной строке размер выходного файла. Использовать для вывода другой том.

**?BINCOM—E—OUTPUT FILE FULL DEV:FILNAM.TYP**

Причина. На томе нет места для записи файла. Обнаружен маркер физического конца тома (для магнитной ленты).

Действие. Сжать том по команде SQUEEZE. Удалить или записать на другой том ненужные файлы. Использовать конструкцию [N] для выходного файла. Использовать том.

**?BINCOM—I—COMPARING/DEV:FILNAM.TYP—DEV:FILNAM.TYP**

Причина. Сравнимые входные файлы имеют различия.

Действие. Не требуется.

**?BINCOM—I—NO DIFFERENCES FOUND**

Причина. Сравнимые файлы идентичны. BINCOM не создает выходной файл, если не указывается переключатель /O в командной строке или используется команда монитора DIFFERENCES/ALWAYS.

Действие. Не требуется.

?BINCOM—W—DEVICE IS LONGER DEV:

Причина. Сравнимые устройства имеют различный размер буфера.

Действие. Не требуется.

?BINCOM—W—DEVICE ARE DIFFERENT

Причина. Сравнимые устройства различны.

Действие. Не требуется.

?BINCOM—W—FILE IS LONGER DEV:FILNAM.TYP

Причина. Сравнимые файлы имеют различный размер.

Действие. Не требуется.

?BINCOM—W—FILES ARE DIFFERENT

Причина. Сравнимые файлы различны.

Действие. Не требуется.

?BOOT—F—NO BOOT VOLUME

Причина. На томе отсутствует начальный загрузчик.

Действие. Использовать переключатель DUP /U или команду монитора COPY/BOOT для записи начального загрузчика на том.

?BOOT—U—CONFLICTING SYSGEN OPTIONS

Причина. Используются несовпадающие варианты генерации драйвера устройства и RMON. Выполнение завершается.

Действие. Проверить соответствие вариантов генерации.

?BOOT—U—HANDLER FILE NOT FOUND

Причина. Не найден драйвер системного устройства во время операции загрузки. Выполнение завершается.

Действие. Скопировать драйвер (XX.SYS) на системный том. Вновь повторить операцию загрузки.

?BOOT—U—INSUFFICIENT MEMORY

Причина. Объем оперативной памяти не позволяет разместить загружаемый монитор. Попытка загрузить монитор в меньший объем памяти, чем установлено в системе. Выполнение завершается.

Действие. Использовать другой монитор и переключатель DUP /U или команду монитора COPY/BOOT для записи начального загрузчика этого монитора на системный том. Проверить наличие памяти на устройстве для размещения монитора.

?BOOT—U—I/O ERROR

Причина. Ошибка ввода-вывода во время загрузки системы: на системном томе отсутствует начальный загрузчик монитора. Ошибка при загрузке активной магнитной ленты. Выполнение завершается.

Действие. Использовать команду COPY/BOOT:XX или переключатель DUP/U для записи начального загрузчика монитора на системный том. Если ошибка повторяется, проверить готовность и исправность оборудования. Убедиться, что адрес контроллера магнитной ленты соответствует стандартному.

?BOOT—U—MONITOR FILE NOT FOUND

Причина. На системном томе нет монитора с соответствующим данному загрузчику именем. Выполнение завершается.

Действие. Проверить правильность выбора системного тома.

?BOOT—U—NO MMU

Причина. Попытка загрузить монитор XM при отсутствии диспетчера памяти. Выполнение завершается.

Действие. Использовать монитор SJ или FB, или комплекс, имеющий диспетчер памяти. Вновь загрузить систему.

?BOOT—U—SWAP FILE TOO SMALL

Причина. Размер файла SWAP.SYS меньше 26 (десятичное) блоков. Выполнение завершается.

Действие. Скопировать файл SWAP.SYS на другой системный (!) том, или использовать переключатель DUP /C или команду монитора CREATE для создания файла необходимой длины. Можно также использовать переключатель DUP/T или команду монитора CREATE/EXTENSION для расширения файла.

?BOOT—U—SWAP.SYS NOT FOUND

Причина. На загружаемом томе отсутствует файл SWAP.SYS. Выполнение завершается.

Действие. Скопировать файл SWAP.SYS на другой системный том. Вновь загрузить систему.

?BOOT—W—ERROR READING HANDLER

Причина. Аппаратная ошибка чтения драйвера. Драйвер содержит плохие блоки.

Действие. Проверить готовность и исправность оборудования.

?BOOT—W—INVALID OR MISSING TT.SYS

Причина. Драйвер TT.SYS отсутствует на системном томе во время загрузки монитора SJ, не имеющего мультитерминальной поддержки. Вариант генерации драйвера TT: не соответствует варианту генерации монитора.

Действие. Записать файл TT.SYS на системный том и вновь запустить систему для установки TT:. Проверить

соответствие вариантов генерации драйвера TT: и монитора.

#### ?BUP—E—WRONG VOLUME NUMBER

Причина. Во время операции восстановления установлен том-копия, номер которого ошибочен.

Действие. Установить соответствующий номер тома-копии. Повторить операцию.

#### ?BUP—F—CHANNEL IN USE DEV:FILNAM.TYP

Причина. Ошибка при обращении к программе BUP.

Действие. Перезагрузить систему и повторить операцию. Если ошибка повторится, создать новую копию BUP.SAV и повторить операцию.

#### ?BUP—F—DEVICE IN USE

Причина. Указанное устройство используется другим заданием.

Действие. Прервать выполнение задания или дождаться его завершения. Повторить операцию.

#### ?BUP—D—DIRECTORY INPUT ERROR DEV:

Причина. Аппаратная ошибка во время чтения справочника.

Действие. Проверить исправность и правильность включения аппаратуры. Проверить том на плохие блоки. Повторить операцию. Если ошибка повторяется, скопировать максимально возможное число файлов на другой том. Отформатировать том и повторить операцию. Если ошибка повторяется снова, заменить том другим.

#### ?BUP—F—DIRECTORY OUTPUT ERROR DEV:

Причина. Аппаратная ошибка во время записи справочника.

Действие. См. ?BUP—F—DIRECTORY INPUT ERROR DEV:

#### ?BUP—F—DUPLICATE OPTION

Причина. Один и тот же переключатель указан в командной строке более одного раза.

Действие. Ввести правильную командную строку.

#### ?BUP—F—ENOUGH SPACE ON ONE VOLUME — USE PIP

Причина. Входные данные могут разместиться на одном выходном томе.

Действие. Необходимо использовать программу PIP (см. [2]).

#### ?BUP—F—ERROR READING DIRECTORY

Причина. См. ?BUP—F—DIRECTORY INPUT ERROR DEV:

Действие. См. ?BUP—F—DIRECTORY INPUT ERROR DEV:

#### ?BUP—F—FETCH ERROR DEV:

Причина. Ошибка при обращении к программе BUP.

Действие. Перезагрузить систему и повторить операцию. Если ошибка повторится, создать новую копию

BUP.SAV и повторить операцию.

?BUP—F—FILE NOT FOUND DEV:FILNAM.TYP

Причина. На том-копии не существует входного файла, указанного в командной строке. Управление передано монитору.

Действие. Заменить том-копию другим, на котором находится указанный входной файл. Повторить операцию.

?BUP—F—INPUT ERROR DEV:FILNAM.TYP

Причина. Аппаратная ошибка во время операции чтения.

Действие. См. ?BUP—F—DIRECTORY INPUT ERROR DEV:

?BUP—F—INSUFFICIENT MEMORY

Причина. Недостаточно памяти, необходимой для выполнения требуемой операции.

Действие. Удалить ненужные драйверы устройств. Удалить основное задание. Использовать монитор SJ. Разрешить свопинг USR по команде монитора SET USR SWAP.

?BUP—F—INVALID COMMAND

Причина. В командной строке обнаружена ошибка.

Действие. Ввести правильную командную строку.

?BUP—F—INVALID DEVICE DEV:

Причина. В командной строке указано недопустимое или не существующее устройство.

Действие. Проверить введенную командную строку. Использовать команду монитора SHOW для получения списка устройств, обслуживаемых системой.

?BUP—F—INVALID DEVICE COMBINATION

Причина. Одно и то же устройство и один и тот же привод указаны во входной и выходной спецификации.

Действие. Проверить командную строку.

?BUP—F—INVALID DIRECTORY

Причина. Справочник указанного устройства не структуры ФОДОС-2.

Действие. Инициализировать том перед использованием по команде монитора INITIALIZE.

?BUP—F—INVALID OPTION

Причина. В командной строке указан недопустимый переключатель.

Действие. Ввести правильную командную строку.

?BUP—F—INVALID OPTION COMBINATION

Причина. В командной строке обнаружена недопустимая комбинация переключателей.

Действие. Ввести правильную командную строку.

?BUP—F—NO ROOM ON OUTPUT DEVICE DEV:

Причина. Недостаточно места на выходном томе для завершения операции восстановления.

Действие. Заменить выходной том другим, с большим числом свободных блоков. Повторить операцию.

**ВUP—F—NOT A BACKUP VOLUME DEV:**

Причина. Попытка получения с помощью переключателя /L листинга справочника тома, который не является томом-копией.

Действие. Заменить том на нужную копию-том. Повторить операцию.

**?ВUP—F—OPERATION NOT COMPLETED**

Причина. Прервано выполнение операции программы ВUP, т. к. ответ на запрос начинался с «N», или нажато СУ/С.

Действие. Для продолжения операции на тот же запрос необходимо напечатать «Y» и нажать клавишу <BK>.

**?ВUP—F—OUTPUT ERROR DEV:FILNAM.TYP**

Причина. Аппаратная ошибка во время записи файла.

Действие. См. ?ВUP—F—DIRECTORY INPUT ERROR DEV:.

**?ВUP—F—TOO MANY BAD BLOCKS DEV:**

Причина. Во время операции восстановления или операции копирования с диска на диск было обнаружено более 25 плохих блоков.

Действие. Проверить, правильно ли установлен диск. Если правильно, то заменить его другим.

**?ВUP—F—VOLUME DOES NOT BELONG TO THIS SET**

Причина. Во время операции восстановления установлен том-копия с именем файла и номером тома-копии, соответствующими указанному входному файлу, однако число блоков, находящихся на установленном томе-копии, не соответствует числу блоков тома, восстановленного для восстановления (из другого набора томов-копий).

Действие. Использовать том-копию, являющийся частью данного набора томов-копий для восстановления.

**?ВUP—F—WILDCARDS NOT PERMITTED ON COMMAND LINE**

Причина. В командной строке присутствует недопустимая конструкция «\*».

Действие. Указать в командной строке имя файла вместо конструкции «\*».

**?ВUP—F—WRONG VERSION**

Причина. Попытка вызова программы ВUP системой ФО-ДОС.

Действие. Исключить такую попытку. Работать с программой BUP можно только в системе ФОДОС-2.

?BUP—I—BAD BLOCK SCAN STARTED...

Причина. Программа BUP начала сканирование тома на плохие блоки.

Действие. Не требуется.

?BUR—I—BAD BLOCKS DETECTED: USE ANOTHER VOLUME

Причина. Программа BUP обнаружила плохие блоки на указанном томе.

Действие. Заменить том другим и повторить операцию.

?BUP—I—COPY OPERATION IS COMPLETE

Причина. Программа BUP закончила копирование или восстановление файла (или тома).

Действие. Не требуется.

?BUP—I—CREATED VOLUME N

Причина. Программа BUP копирует информацию на указанный том-копию номер N.

Действие. Не требуется.

?BUP—I—NO BAD BLOCKS DETECTED

Причина. На указанном томе плохих блоков не обнаружено.

Действие. Не требуется.

?BUP—I—RESTORE OPERATION STARTER FROM VOLUME N

Причина. Начинается операция восстановления с тома-копии номер N.

Действие. Не требуется.

?BUP—W—BAD BLOCK READ AT NNNNNN

Причина. Во время операции восстановления или операции копирования с диска на диск во входном томе обнаружен плохой блок номер NNNNNN.

Действие. Не требуется.

?BUP—W—FILE NOT FOUND DEV:FILNAM.TYP

Причина. Во время операции восстановления на установленном входном томе-копии не обнаружен указанный в командной строке входной файл. Был использован том-копия из другого набора томов-копий или повторно установлен один из томов-копий.

Действие. Установить соответствующий том-копию (содержащий указанный входной файл и указанный номер тома-копии).

?BUP—W—INCOMPATIBLE OUTPUT VOLUME SIZE

Причина. На выходном устройстве установлен том, размер которого отличен от размера первого используемого в операции копирования тома-копии. Например, это сообщение появляется, если на выходном устройстве во время операции копирования были по-

следовательно установлены гибкие диски, как с одинарной, так и с двойной плотностью.

Действие. Использовать тома того же размера, что и первый том-копия. Повторить операцию.

?BUP—W—NOT A BACKUP VOLUME DEV:

Причина. Установленный том не инициализирован для использования его в операциях копирования или восстановления с помощью программы BUP.

Действие. В ответ на запрос напечатать «Y» и нажать клавишу <BK> для инициализации тома.

?CREF—F—CHAIN—ONLY XXXX

Причина. Попытка использовать команду R CREF или START для выполнения находящейся в памяти копии CREF.

Действие. (XXXX — часто используемая системная программа). Использовать языковой процессор для обращения к CREF.

?CREF—F—CRF FILE ERROR

Причина. Ошибка чтения файла DK:CREF.TMP.

Действие. Вновь запустить языковой процессор для создания нового входного файла CREF.

?CREF—F—DEVICE NOT FOUND

Причина. Попытка записи листинга на магнитную ленту или кассету при отсутствии драйвера магнитной ленты или кассеты. Входной файл CREF.TMP отсутствует на устройстве с произвольным доступом. Указано недопустимое устройство при обращении языкового процессора к CREF. Возможна системная ошибка.

Действие. Использовать команду LOAD для загрузки соответствующего драйвера устройства перед записью листинга на магнитную ленту или кассету. Убедиться, что файл CREF.TMP находится на устройстве с произвольным доступом. Повторить операцию. Если ошибка повторяется, создать новую копию программы CREF или системы.

?CREF—F—LST FILE ERROR

Причина. Ошибка записи таблицы перекрестных ссылок в листинг. Недостаточно места на выходном томе для размещения файла листинга. Возможна аппаратная ошибка.

Действие. Увеличить объем свободной памяти, удалив драйверы неиспользуемых устройств, а также основное

и системные задания. Проверить готовность и исправность оборудования.

?CSI—F—DEVICE FULL

Причина. На указанном устройстве нет места для размещения выходного файла, созданного программой пользователя с помощью запроса .CSIGEN.

Действие. Увеличить размер свободной области на томе: уничтожить или переписать на другой том ненужные файлы, сжать том по команде монитора SQUEEZE, использовать переключатель /ALLOCATE или конструкцию [N] для выходного файла.

?CSI—F—FILE NOT FOUND

Причина. Не найден входной файл, вызванный программой пользователя с помощью .CSIGEN.

Действие. Проверить поданную команду.

?CSI—F—INVALID COMMAND

Причина. Не указано имя выходного файла; длина командной строки превышает 80 символов, или командная строка содержит недопустимые символы, например, пробелы; число файлов, указанных в команде клавиатурного монитора, превышает допустимое; длина входной строки в запросе .GTLIN превышает 80 символов. Сообщение выдается в том случае, если программа пользователя использовала запрос .CSIGEN или .CSISPC.

Действие. Проверить поданную команду. Убедиться, что длина входной строки в запросе .GTLIN не превышает 80 символов.

?CSI—F—INVALID DEVICE

Причина. Устройство, указанное в программе пользователя с помощью запроса .CSIGEN, не существует. В системе не установлено обслуживание таких устройств, как TT: или BA:, возможно, отсутствует соответствующий драйвер. Попытка ввести команду DIRECTORY, когда система работает под управлением монитора SJ и файл TT.SYS отсутствует на системном устройстве.

Действие. Проверить поданную команду. Скопировать требуемый драйвер на системный том и вновь загрузить систему.

?CSI—F—PROT FILE

Причина. Файлы, указанные в программе пользователя с помощью запроса .CSIGEN, уже существуют и защищены.

Действие. Использовать другое имя для открытия нового файла, или уничтожить, переименовать или изменить уровень защиты существующего файла по команде монитора UNPROTECT или переключателю /Z программы PIP.

?DIR—F—DEVICE NOT ACTIVE

Причина. Операция ввода-вывода невозможна на указанном устройстве или томе; устройство отключено; нет тома на устройстве; запрещена запись; устройство не описано в таблицах монитора.

Действие. Проверить правильность включения устройства. Установить обслуживание устройства по команде монитора INSTALL.

?DIR—F—ERROR READING DIRECTORY

Причина. Ошибка чтения справочника тома.

Действие. Проверить исправность и правильность включения аппаратуры. Проверить диск на плохие блоки.

?DIR—F—INPUT ERROR DEV:

Причина. Произошла аппаратная ошибка при чтении входного файла.

Действие. Проверить исправность и правильность включения аппаратуры. Проверить том на плохие блоки.

?DIR—F—INSUFFICIENT MEMORY

Причина. Недостаточно памяти для получения упорядоченного справочника.

Действие. Использовать другой переключатель.

?DIR—F—INVALID COMMAND

Причина. В командной строке обнаружена ошибка.

Действие. Ввести правильную командную строку.

?DIR—F—INVALID DEVICE DEV:

Причина. Операция недопустима на указанном устройстве: устройство несправочной структуры (как, например, TT: или BA:) или в системе не установлено обслуживание указанного устройства.

Действие. Проверить введенную командную строку. Использовать команду монитора SHOW для получения списка устройств, обслуживаемых системой.

?DIR—F—INVALID DIRECTORY

Причина. Структура справочника тома устройства (метки тома для магнитной ленты) не является структурой ФОДОС-2.

Действие. Не требуется.

?DIR—F—INVALID OPTION: /X

**Причина.** В командной строке обнаружен недопустимый переключатель или комбинация переключателей.

**Действие.** Ввести правильную командную строку.

?DIR—F—OUTPUT ERROR DEV:FILNAM.TYP

**Причина.** Произошла аппаратная ошибка во время записи выходного файла.

**Действие.** См. ?DIR—F—INPUT ERROR DEV:FILNAM.TYP.

?DIR—F—OUTPUT FILE FULL

**Причина.** Выходной файл заполнен; на томе нет места для файла справочника.

**Действие.** Вывести справочник на печатающее устройство; Использовать конструкцию [N] для выходного файла.

?DIR—F—PROTECTED FILE ALREADY EXISTS DEV:FILNAM.TYP

**Причина.** Попытка создания защищенного файла, имеющего такое же имя, что и у уже существующего защищенного файла.

**Действие.** Воспользоваться командой монитора RENAME/NOPROTECT или переключателем PIP /Z для снятия защиты уже существующего файла или воспользоваться другим именем для создания нового файла.

?DIR—I—BACKUP FILE—USE/BACKUP OPTION

**Причина.** Файл, указанный в командной строке, был создан с помощью программы BUP (см. [2]).

**Действие.** Использовать команду монитора DIRECTORY/BACKUP.

?DUMP—F—DEVICE FULL DEV:FILNAM.TYP

**Причина.** На устройстве недостаточно места для выходного файла.

**Действие.** Увеличить размер свободной области на томе: удалить ненужные файлы или переписать их на другой том, сжать том по команде SQUEEZE.

?DUMP—F—FILE OR INPUT DEVICE NOT FOUND

**Причина.** Не найден входной файл; в системе нет драйвера заданного устройства ввода; в мониторе не установлено обслуживание заданного устройства ввода.

**Действие.** Ввести правильную командную строку; установить правильное обслуживание требуемого устройства.

?DUMP—F—INPUT ERROR DEV:FILNAM.TYP

**Причина.** Ошибка чтения.

**Действие.** Проверить готовность и исправность оборудования. Повторить выполнение операции.

?DUMP—F—INSUFFICIENT MEMORY

**Причина.** Недостаточно оперативной памяти для выполнения заданной операции.

**Действие.** Освободить часть оперативной памяти (удалить ненужные драйверы, удалить основное задание, использовать монитор одного задания). Повторить операцию.

?DUMP—F—INVALID COMMAND

**Причина.** В командной строке обнаружена ошибка.

**Действие.** Ввести правильную командную строку.

?DUMP—F—INVALID OPTION: /X

**Причина.** В командной строке обнаружен недопустимый переключатель.

**Действие.** Ввести правильную командную строку.

?DUMP—F—NO LP

**Причина.** В системе нет драйвера построчно-печатающего устройства.

**Действие.** Повторить операцию, задав доступное устройство вывода.

?DUMP—F—OUTPUT ERROR DEV:FILNAM.TYP

**Причина.** Ошибка записи; на томе, используемом для вывода, нет свободной области для записи выходного файла.

**Действие.** Проверить готовность и исправность оборудования; проверить том на плохие блоки. Повторить операцию, указав в командной строке размер выходного файла, или использовав для вывода другой том.

?DUMP—F—PROTECTED FILE ALREADY EXISTS DEV:FILNAM.TYP

**Причина.** Попытка создать файл с тем же именем, что и у имеющегося защищенного файла.

**Действие.** Снять защиту имеющегося файла по команде монитора UNPROTECT или по переключателю /Z программы PIP, или создать файл с другим именем.

?DUMP—F—UNEXPECTED EOF DEV:FILNAM.TYP

**Причина.** В заданной командной строке в переключателе /E:N (/O:N, /S:N) значение N превосходит длину входного файла.

**Действие.** Определить действительный размер входного файла. Повторить команду.

?DUP—E—OPERATION NOT COMPLETED

**Причина.** При использовании переключателя /W в ответ на запрос на выполнение операции напечатано «N» или нажато СУ/С.

**Действие.** Повторить команду. Для выполнения операции на-

печатать «Y» и нажать клавишу <BK> в ответ на запрос.

?DUP—F—BAD BLOCK IN SYSTEM AREA DEV:

Причина. Плохие блоки в системной области диска.

Действие. Отформатировать диск и повторить операцию. Если ошибка повторяется, диск необходимо заменить.

?DUP—F—CAN'T SQUEEZE SY: WHILE FOREGROUND LOADED

Причина. Попытка сжать системный том, когда системное или основное задание находится в памяти.

Действие. Удалить основное или системное задание из памяти и повторить команду.

?DUP—F—CHANNEL IN USE DEV:FILNAM.TYP

Причина. Ошибка при обращении (вызове) к программе DUP.

Действие. Перезагрузить систему и повторить операцию. Если ошибка повторится снова, создать новую копию DUP.SAV и повторить операцию.

?DUP—F—CHANNEL NOT OPEN DEV:FILNAM.TYP

Причина. Ошибка при обращении к программе DUP.

Действие. Перезагрузить систему и повторить операцию. Если ошибка повторится снова, создать новую копию DUP.SAV и повторить операцию.

?DUP—F—CONFLICTING SYSGEN OPTIONS

Причина. Используются несовпадающие варианты генерации драйвера устройства и RMON.

Действие. Проверить соответствие вариантов генерации.

?DUP—F—DEVICE FULL DEV:FILNAM.TYP

Причина. Недостаточно места на выходном томе для размещения указанного файла или данных.

Действие. Использовать переключатель /C, указав меньший размер файла, или использовать переключатель /S для объединения неиспользуемых областей на выходном томе. Если использовался переключатель /S для сжатия входного тома на выходной, инициализировать выходной том, указав меньшее число для сегментов справочника. Если использовались переключатели /I/F, скопировать на другой выходной том или использовать переключатель /S для создания достаточно большой смежной неиспользуемой области на имеющемся выходном томе.

?DUP—F—DIRECTORY FULL DEV:

Причина. Нет места в справочнике выходного тома для создания выходного файла.

Действие. С помощью переключателя /S объединить неис-

пользуемые области на выходном томе или использовать другой том.

?DUP—F—DIRECTORY INPUT ERROR DEV:

Причина. Ошибка чтения справочника указанного устройства.

Действие. Проверить исправность и правильность включения аппаратуры. Проверить том на плохие блоки. Повторить операцию. Если ошибка повторяется, скопировать максимально возможное число файлов на другой том. Отформатировать том и повторить операцию. Если ошибка повторится снова, заменить том другим.

?DUP—F—DIRECTORY OUTPUT ERROR DEV:

Причина. См. ?DUP—F—DIRECTORY INPUT ERROR DEV:.

Действие. См. ?DUP—F—DIRECTORY INPUT ERROR DEV:.

?DUP—F—FETCH ERROR DEV:

Причина. Произошел сбой в работе операционной системы.

Действие. Перезагрузить систему и повторить операцию. Если ошибка повторится, использовать новую копию DUP.SAV и драйвера указанного устройства. Повторить операцию.

?DUP—F—FILE NOT FOUND DEV:FILNAM.TYP

Причина. Указанный файл не найден.

Действие. Проверить командную строку. Повторить операцию.

?DUP—F—INPUT ERROR DEV:FILNAM.TYP

Причина. Аппаратная ошибка во время операции чтения.

Действие. Проверить исправность и правильность включения аппаратуры. Проверить диск на плохие блоки.

?DUP—F—INSUFFICIENT MEMORY

Причина. Недостаточно памяти под буферы для выполнения требуемой операции.

Действие. Выгрузить ненужные драйверы устройств. Выгрузить основное задание. Использовать монитор SJ. Разрешить свопинг USR по команде монитора SET USR SWAP.

?DUP—F—INVALID COMMAND

Причина. В командной строке обнаружена ошибка.

Действие. Ввести правильную командную строку.

?DUP—F—INVALID DEVICE DEV:

Причина. Указанное устройство не занесено в таблицу устройств монитора.

Действие. Проверить командную строку. Используя команду

монитора INSTALL, занести указанное устройство в таблицу устройств монитора.

?DUP—F—INVALID DIRECTORY DEV:

Причина. Том указанного устройства имеет недопустимую в системе ФОДОС-2 структуру справочника.

Действие. Инициализировать том, используя переключатель /Z.

?DUP—F—INVALID OPTION: /X

Причина. В командной строке указан недопустимый переключатель.

Действие. Проверить командную строку.

?DUP—F—INVALID RESTORE DATA DEV:

Причина. DU не в состоянии восстановить справочник тома с помощью переключателя /D, потому что в блоке номер 1 тома хранятся недопустимые данные.

Действие. Процедура восстановления тома невозможна.

(Справочник тома инициализирован безвозвратно).

?DUP—F—INVALID VALUE SPECIFIED WITH OPTION: /X

Причина. В командной строке указан недопустимый аргумент переключателя.

Действие. Проверить командную строку.

?DUP—F—NON BOOTABLE DRIVER DEV:FILNAM.TYP

Причина. Попытка загрузить устройство, которое не содержит начальный загрузчик. Отсутствует двоеточие (:) в имени физического устройства.

Действие. Использовать другое устройство для выполнения операции. Повторить команду, указав двоеточие в имени физического устройства.

?DUP—F—NO ROOM FOR FILE DEV:FILNAM.TYP

Причина. Размер неиспользуемой области памяти меньше указанного значения [N[ по переключателю DUP/C или указанную область уже занимает файл.

Действие. Удалить файл или файлы с указанной области тома. Повторить команду.

?DUP—F—NO SPACE FOR EXTENSION DEV:FILNAM.TYP

Причина. Размер неиспользуемой области, следующая за расширяемым (по переключателю /T:N) файлом, меньше значения аргумента переключателя.

Действие. Указать меньшее значение аргумента переключателя /T:N или удалить файл или файлы с указанной области тома.

?DUP—F—NO B3 BOOT ON VOLUME

Причина. Попытка загрузить том, не содержащий начального загрузчика системы ФОДОС-2.

**Действие.** Использовать переключатель /U для записи на том начального загрузчика системы ФОДОС-2 или использовать переключатель /Q для загрузки системы ФОДОС.

**?DUP—F—OUTPUT ERROR DEV:FILNAM.TYP**

**Причина.** Аппаратная ошибка во время операции записи.

**Действие.** См. ?DUP—F—INPUT ERROR DEV:FILNAM.TYP.

**?DUP—F—OUTPUT FILE EXISTS DEV:FILNAM.TYP**

**Причина.** Файл с именем, указанным в командной строке с использованием переключателя /C, уже существует на выходном томе.

**Действие.** Удалить существующий на томе файл или указать другое имя выходного файла.

**?DUP—F—SIZE FUNCTION FAILED**

**Причина.** Ошибка при определении размера монтируемого тома. (Монитор, драйвер устройства или DUP могут быть повреждены). Это сообщение возникает, когда привод с указанным номером не поддерживается монитором, созданным при генерации системы.

**Действие.** Перезагрузить систему и повторить операцию. В случае повторения ошибки создать новую копию DUP.SAV и повторить операцию.

**?DUP—F—UNEXPECTED EOF DEV:FILNAM.TYP**

**Причина.** Ошибка при обращении к программе DUP.

**Действие.** Перезагрузить систему и повторить операцию. Если ошибка повторится снова, создать новую копию DUP.SAV и повторить операцию.

**?DUP—F—UNINITIALIZED VOLUME DEV:**

**Причина.** Справочник указанного тома не инициализирован.

**Действие.** Инициализировать справочник тома, используя переключатель /Z.

**?DUP—F—VOLUME NOT FODOS FORMAT DEV:**

**Причина.** Указан том, имеющий недопустимый формат в системе ФОДОС-2.

**Действие.** Отформатировать том, используя команду монитора FORMAT, или использовать том системы ФОДОС-2.

**?DUP—I—NO BAD BLOCKS DETECTED DEV:**

**Причина.** Во время операции сканирования устройства (с использованием переключателя /K или /B) плохих блоков не обнаружено.

**Действие.** Не требуется.

**?DUP—U—CHANNEL NOT OPEN**

Причина. Ошибка при обращении к программе DUP.  
Действие. Перезагрузить систему и повторить операцию. Если ошибка повторится снова, создать новую копию DUP.SAV и повторить операцию.

?DUP—U—SYSTEM ERROR

Причина. См. ?DUP—U—CHANNEL NOT OPEN.

Действие. См. ?DUP—U—CHANNEL NOT OPEN.

?DUP—U—WRONG VERSION

Причина. Произошла попытка обращения системы ФОДОС к программе DUP, входящей в состав ФОДОС-2.

Действие. Использовать DUP системы ФОДОС.

?DUP—W—BAD BLOCKS DETECTED NNNNN.

Причина. Во время операции сканирования устройства (с использованием переключателей /K или /B) обнаружено NNNNN плохих блоков.

Действие. Не требуется.

?DUP—W—DEVICE FULL DEV:FILNAM.TYP

Причина. На выходном томе недостаточно места для размещения передаваемых файлов.

Действие. Использовать другой том, больший или равный входному.

?DUP—W—OUTPUT ERROR DEV:FILNAM.TYP

Причина. Ошибка во время операции сжатия.

Действие. DUP создает файл с типом .BAD на месте плохого блока и продолжает операцию.

?DUP—W—OWNER NAME TRUNCATED AT TEN CHARACTERS

Причина. Имя пользователя для магнитной ленты содержит более десяти символов, DUP «отсекает» лишние.

Действие. Если сокращение не приемлемо, использовать переключатели /Z/V. Указать не более 10-ти символов.

?DUP—W—OWNER NAME TRUNCATED AT TWELVE CHARACTERS

Причина. Имя пользователя для диска содержит более двенадцати символов, DUP «отсекает» лишние символы.

Действие. Если сокращение не приемлемо, использовать переключатели /Z/V. Указать не более 12-ти символов.

?DUP—W—TOO MANY BAD BLOCKS DEV:

Причина. Обнаружено более 128 плохих блоков во время операции сканирования.

Действие. Том не пригоден для использования и подлежит замене.

?DUP—W—VERIFICATION ERROR AT RELATIVE BLOCK NNNNNN

Причина. При использовании переключателей /I/H обнаружено несоответствие входных и выходных данных.  
Действие. Проверить том на плохие блоки. Повторить операцию.

?DUP—W—VOLUME ID TRUNCATED AT SIX CHARACTERS

Причина. В идентификаторе тома магнитной ленты более шести символов. DUP «отсекает» лишние символы.

Действие. Если сокращение не приемлемо, использовать переключатели /Z/V. Указать не более 6-ти символов.

?DUP—W—VOLUME ID TRUNCATED AT TWELVE CHARACTERS

Причина. Идентификатор тома для диска содержит более двенадцати символов, DUP «отсекает» лишние символы.

Действие. Если сокращение не приемлемо, использовать переключатель /V для изменения идентификации тома. Указать не более 12-ти символов.

?EL—W—BUFFER IS FULL, LOGGING SUSPENDED

Причина. Внутренний буфер регистрации полон. Регистрация новых сообщений приостанавливается, статистики сохраняются вместе с подсчетом незарегистрированных сообщений. Состояние не влияет на операцию системы.

Действие. Удалить содержимое внутреннего буфера регистрации ошибок по команде SET EL PURGE.

?EL—W—DEVICE STATISTICS TABLE IS FULL

Причина. Таблица статистик устройств полна. Регистрация ошибок продолжается, но сохраняются статистики только установленных во время генерации устройств. Состояние не влияет на операцию системы.

Действие. Выполнить генерацию системы и установить необходимые устройства.

?ELINIT—F—CHANNEL ERROR

Причина. Ошибка при обращении к каналу.

Действие. Повторить операцию, если ошибка повторяется, создать новую копию программы ELINIT или системы.

?ELINIT—F—DEVICE FULL

Причина. Нет места на указанном устройстве для создания или расширения файла ERRLOG.DAT.

Действие. Использовать команду монитора SQUEEZE или переключатель DUP /S для сжатия тома, или другой том для выполнения операции.

?ELINIT—F—DEVICE NOT AVAILABLE

Причина. Необходимое устройство не загружено.

Действие. Загрузить драйвер устройства и вновь выполнить ELINIT.

?ELINIT—F—ERRLOG.DAT FILE I/O ERROR

Причина. Ошибка чтения или записи файла ERRLOG.DAT.

Действие. Проверить готовность и исправность оборудования.

?ELINIT—F—ERRLOG TASK NOT ACTIVE

Причина. Попытка инициировать регистрацию ошибок перед выполнением задания регистрации ошибок.

Действие. Выполнить EL по команде монитора FRUN или SRUN перед выполнением ELINIT.

?ELINIT—F—INTERNAL ERROR

Причина. Ошибка при обращении к программе ELINIT или сбой программы.

Действие. См. ?ELINIT—F—CHANNEL ERROR.

?ELINIT—F—PROTECTED FILE ALREADY EXISTS ERRLOG.DAT

Причина. Файл ERRLOG.DAT уже защищен.

Действие. Использовать команду UNPROTECT или переключатель PIP/Z для изменения уровня защиты существующего файла, или указать другое имя для создания нового файла. Вновь выполнить программу ELINIT.

?ELINIT—W—INVALID COMMAND

Причина. Введен неправильный ответ на вопрос оператора об инициализации команды. Вопрос повторен вновь.

Действие. Правильно ответить на вопрос.

?ERRLOG—F—CHANNEL ERROR

Причина. См. ?EL—F—CHANNEL ERROR.

Действие. См. ?EL—F—CHANNEL ERROR.

?ERRLOG—F—INTERNAL ERROR

Причина. См. ?ELINIT—F—INTERNAL ERROR.

Действие. См. ?ELINIT—F—INTERNAL ERROR.

?ERRLOG—F—LOG FILE I/O ERROR

Причина. Ошибка чтения или записи файла ERRLOG.DAT.

Действие. Проверить готовность и исправность оборудования.

?ERRLOG—I—REMINDER: TO INITIATE ERROR LOGGING, RUN ELINIT

Причина. Программа ELINIT иницирует регистрацию ошибок.

Действие. Выполнить программу ELINIT.

?ERRLOG—I—TO INITIATE ERROR LOGGING, RUN ELINIT

Причина. Программа ELINIT иницирует регистрацию ошибок.

Действие. Выполнить программу ELINIT.

?ERRLOG—W—INVALID MESSAGE RECEIVED

**Причина.** Попытка связать задачу или фоновое задание, отличное от ERRROUT, с задачей EL. Попытка игнорируется.

**Действие.** Не требуется.

**?ERRLOG—W—LONG FILE FULL, LOGGING SUSPENDED**

**Причина.** Файл ERRLOG.DAT полон. Заголовок блока изменяется, но ошибка записи не регистрируется.

**Действие.** Завершить ERRLOG и вновь инициализировать регистрацию ошибок.

**END BATCH**

**Причина.** Задание BATCH завершено. Управление передается монитору.

**Действие.** Не требуется.

**?ERRROUT—F—CHANNEL ERROR**

**Причина.** См. ?ELINIT—F—CHANNEL ERROR.

**Действие.** См. ?ELINIT—F—CHANNEL ERROR.

**?ERRROUT—F—DEVICE NOT FOUND**

**Причина.** Указанное устройство не найдено.

**Действие.** Установить указанное устройство. Повторить операцию.

**?ERRROUT—F—FILE NOT FOUND**

**Причина.** Входной файл, содержащий статистики зарегистрированных ошибок, не найден.

**Действие.** Ввести правильную командную строку.

**?ERRROUT—F—INPUT ERROR**

**Причина.** Аппаратная ошибка.

**Действие.** Проверить готовность и исправность оборудования.

**?ERRROUT—F—INSUFFICIENT MEMORY**

**Причина.** Недостаточно памяти для обработки файла статистик.

**Действие.** Увеличить область свободной памяти, удалив драйверы неиспользуемых устройств, а также основные и системные задания.

**?ERRROUT—F—INTERNAL ERROR**

**Причина.** См. ?ELINIT—F—INTERNAL ERROR.

**Действие.** См. ?ELINIT—F—INTERNAL ERROR.

**?ERRROUT—F—INVALID COMMAND**

**Причина.** Недопустимый формат команды ERRROUT.

**Действие.** Ввести правильную командную строку.

**?ERRROUT—F—OUTPUT ERROR**

**Причина.** Аппаратная ошибка.

**Действие.** Проверить готовность и исправность оборудования.

**?ERRROUT—F—OUTPUT FILE FULL**

**Причина.** Выходной файл полон.

Действие. Использовать другой том, или указать размер выходного файла с помощью переключателя /ALLOCATE или конструкции [N].

?ERROUT—F—PROTECTED FILE ALREADY EXISTS

Причина. Попытка создать файл с таким же именем, как у имеющегося защищенного файла.

Действие. Использовать команду UNPROTECT или переключатель PIP/Z для изменения уровня защиты существующего файла или другое имя для создания нового файла.

?ERROUT—W—DEVICE FULL

Причина. Нет места на указанном устройстве для создания выходного файла.

Действие. Использовать другое устройство или команду SQUEEZE, или переключатель DUP /S для сжатия тома.

?FORMAT—E—OPERATION NOT COMPLETED

Причина. Выполнение операции форматирования прекращено в ответ на печать пользователем «N» (или любой строки, начинающейся с «N») или CY/C после сообщения системы, вызванного переключателем /W.

Действие. Ответ пользователя на сообщение системы по переключателю /W должен начинаться с «Y» для выполнения операции.

?FORMAT—F—DEVICE ERROR

Причина. Произошла аппаратная ошибка при попытке форматирования, или привод с указанным номером не существует.

Действие. Проверить наличие привода с указанным номером. Если имела место аппаратная ошибка, проверить готовность и исправность оборудования.

?FORMAT—F—DEVICE INVALID OR NOT SUPPORTED BY FORMAT

Причина. Указано недопустимое для программы FORMAT имя устройства.

Действие. Ввести командную строку, указав допустимое имя устройства.

?FORMAT—F—DEVICE NOT READY

Причина. Указанное в командной строке устройство не готово для форматирования или проверки, т. к. устройство выключено или запись на него запрещена; для кассетного диска — скорость вращения меньше допустимой.

Действие. Проверить готовность устройства к операции: устройство должно быть включено и запись на него

разрешена; кассетный диск должен набрать нужную скорость.

**?FORMAT—F—INSUFFICIENT MEMORY**

Причина. Для выполнения указанной операции недостаточно памяти.

Действие. Увеличить область свободной памяти, удалив драйверы неиспользуемых устройств, а также основное и системные задания.

**?FORMAT—F—INVALID COMMAND LINE**

Причина. Недопустимый формат командной строки.

Действие. Ввести командную строку следующего формата: уст: [/прк].

**?FORMAT—F—INVALID OPTION: /X**

Причина. Переключатель, указанный в командной строке, недопустим для указанного устройства.

Действие. Вновь ввести командную строку и использовать только допустимые переключатели.

**?FORMAT—F—INVALID OPTION /X:VAL**

Причина. Для указанного переключателя в командной строке аргумент недопустим.

Действие. Ввести командную строку, указав переключатель без аргумента.

**?FORMAT—F—INVALID VALUE SPECIFIED WITH OPTION: /X**

Причина. С переключателем /X указано недопустимое значение.

Действие. Ввести командную строку, указав допустимое значение для переключателя.

**?FORMAT—F—UNIT NUMBER MUST BE IN RANGE 0—7**

Причина. В командной строке указан недопустимый номер привода для устройства.

Действие. Ввести командную строку, указав номер привода в диапазоне от 0 до 7.

**?FORMAT—F—UNIT NUMBER TOO LARGE**

Причина. В командной строке указан номер привода превышающий допустимый.

Действие. Проверить номер привода, указанный в командной строке, и вновь ввести командную строку.

**?FORMAT—I—FORMATTING ABORTED**

Причина. Форматирование прервано из-за ошибки устройства.

Действие. Не требуется.

**?FORMAT—I—FORMATTING COMPLETE**

Причина. Форматирование завершено.

Действие. Не требуется.

**?FORMAT—I—VERIFICATION ABORTED**

Причина. Проверка прервана из-за ошибки устройства или по желанию пользователя.

Действие. Если необходимо, повторить операцию проверки. Если имела место ошибка устройства, проверить исправность и готовность оборудования.

?FORMAT—I—VERIFICATION COMPLETE

Причина. Программа FORMAT закончила проверку.

Действие. Не требуется.

?FORMAT—U—DEVICE HANDLER FETCH ERROR

Причина. Драйвер устройства не может быть считан с периферийного устройства в основную память.

Действие. Проверить наличие драйвера устройства на системном томе; проверить системный том на наличие плохих блоков. Вновь повторить операцию проверки устройства.

?FORMAT—U—FORMATTING/VERIFYING THE SYSTEM VOLUME IS NOT ALLOWED

Причина. Попытка форматирования или проверки системного тома.

Действие. Указать в командной строке другой привод устройства (отличный от привода системного тома) или использовать переключатель /W (или команду FORMAT/WAIT см. [1]) для замены системного тома томом, который требуется отформатировать.

?FORMAT—U—TOO MANY BAD BLOCKS DEV:

Причина. Число плохих блоков превысило допустимое значение.

Действие. Использовать другой том.

?FORMAT—U—DUPLICATE OPTION: /X

Причина. Переключатель /X дважды указан в командной строке.

Действие. Проверить переключатели в командной строке и вновь ввести командную строку.

## СООБЩЕНИЯ УПРАВЛЯЮЩИХ ФАЙЛОВ

Формат сообщений описан в разделе 1.

?IND—E—INVALID ANSWER OR TERMINATOR

Причина. Недопустимый ответ на подсказку директивы .ASK, .ASKN или .ASKS. Введен ответ AP2<BK> при запрещенном режиме ESCAPE. Подсказка повторяется.

Действие. Ввести правильный ответ на подсказку директивы .ASK, .ASKN или .ASKS.

?IND—E—STRING LENGTH NOT IN RANGE

Причина. В ответ на подсказку директивы .ASKS введена строка символов, значение длины которой не принадлежит диапазону, указанному в директиве .ASKS (см. [1]), подсказка повторяется.

Действие. Ввести правильный ответ на подсказку директивы .ASKS.

?IND—E—VALUE NOT IN RANGE

Причина. В ответ на подсказку директивы .ASKN введено числовое значение, не принадлежащее диапазону, указанному в директиве .ASKN (см. [1]). Подсказка повторяется.

Действие. Ввести правильный ответ на подсказку директивы .ASKN.

?IND—F—BAD RANGE OR DEFAULT SPECIFICATION

Причина. Ответ по умолчанию находится вне диапазона, указанного в директиве .ASKN или .ASKS; в спецификации диапазона использовано недопустимое основание системы счисления; числа в диапазоне указаны в невозрастающем порядке. Значение по умолчанию, присвоенное имени в директиве .ASKS, не является строковым литералом или другим строковым именем.

Действие. Использовать допустимое основание системы счисления в спецификации диапазона. Убедиться, что значение по умолчанию, присвоенное имени в директиве .ASKS, является строковым литералом или другим строковым именем.

?IND—F—DATA FILE OPEN

Причина. Попытка выполнить команду клавиатурного монитора для открытого файла.

Действие. Закрывать открытый файл перед выходом управляющего файла из программы IND.

?IND—F—DELETING SPECIAL SYMBOL

Причина. Попытка уничтожить специальное имя IND.

Действие. Проверить команду. Не уничтожать специальное имя IND.

?IND—F—FILE ALREADY OPEN

Причина. Попытка открыть уже открытый файл с помощью директив типа .OPEN, .OPENA или .OPENR.

Действие. Убедиться, что файл закрыт, прежде чем пытаться открыть его.

?IND—F—FILE NOT FOUND

Причина. Управляющий файл отсутствует в справочнике указанного тома.

Действие. Проверить командную строку.

?IND—F—FILE NOT OPEN

Причина. Попытка доступа к закрытому файлу.

Действие. Открыть файл.

?IND—F—FILE READ ERROR

Причина. Аппаратная ошибка чтения файла.

Действие. Проверить готовность и исправность оборудования.

?IND—F—INVALID ATTEMPT TO ERASE SYMBOL

Причина. Попытка уничтожить имя, не принадлежащее блоку управляющего файла.

Действие. Повторить операцию. Уничтожить имена, находящиеся в пределах текущего блока управляющего файла.

?IND—F—INVALID COMMAND

Причина. Ошибка в командной строке, используемой другим управляющим файлом.

Действие. Ввести правильную командную строку.

?IND—F—INVALID DEVICE OR UNIT

Причина. Указано недопустимое или несуществующее устройство.

Действие. Ввести правильную командную строку.

?IND—F—INVALID FILE NUMBER

Причина. Указан недопустимый номер файла.

Действие. Ввести командную строку, указав номер файла в диапазоне от 0 до 3.

?IND—F—INVALID KEYWORD

Причина. Синтаксическая ошибка в ключевом слове (ключевому слову предшествует точка) в формате директивы условного ветвления.

Действие. Ввести правильную командную строку.

?IND—F—INVALID NESTING

Причина. Обнаружено несоответствие директив .BEGIN и .END.

Действие. Убедиться, что каждой директиве .BEGIN соответствует директива .END.

?IND—F—INVALID OPERATOR FOR OPERATION

Причина. Попытка использовать арифметический оператор ('+', '-', '\*', '/') в логическом выражении директивы .SETL.

Действие. Использовать только логические операторы ('&'(И), 'I'(ИЛИ), '^'(НЕ)) в логическом выражении директивы .SETL.

?IND—F—INVALID OPTION

Причина. Указан недопустимый переключатель.

Действие. Ввести правильную командную строку.

?IND—F—LABEL NOT AT BEGINNING OF LINE

Причина. Метка введена не с первой позиции.

Действие. Убедиться, что метка начинается с первой позиции и в метке отсутствуют пробелы.

?IND—F—MAXIMUM INDIRECT FILES EXCEEDED

Причина. Попытка обращения к управляющему файлу на уровне вложения, превышающем допустимый.

Действие. Ограничить вложение управляющих файлов тремя уровнями.

?IND—F—NULL CONTROL STRING

Причина. Не определены ограничители в «управляющей строке» директивы .PARSE (см. [1]). Ограничители в «управляющей строке» не могут быть опущены.

Действие. Ввести правильную командную строку.

?IND—F—NUMERIC UNDER— OR OVERFLOW

Причина. Числовое значение имени не принадлежит диапазону от 0 до 177777 (восьмеричное) или 65535 (десятичное).

Действие. Убедиться, что числовое значение имени принадлежит указанному диапазону.

?IND—F—PROMPT STRING TOO LARGE

Причина. Длина подсказки в директиве .ASK, .ASKN или .ASKS превышает 80 символов.

Действие. Ввести правильную командную строку.

?IND—F—REDEFINING SPECIAL SYMBOL

Причина. Попытка изменить значение специального имени.

Действие. Не менять значение специального имени IND.

?IND—F—REDEFINING SYMBOL TO DIFFERENT TYPE <SYMBOL>

Причина. Попытка установить в директиве .ASK, .ASKN, .ASKS, .READ, .SETT, .SETF, .SETL, .SETN или .SETS уже определенное имя <SYMBOL> с другим типом. Не определено числовое имя <SYMBOL>, указанное в директиве .SETT, .SETF или .SETL. Числовое имя <SYMBOL>, указанное в директиве .IFT или .IFF определено как логическое или строковое имя.

Действие. Убедиться, что новое имя определено с тем же типом. Определить числовое имя, прежде чем использовать его в директиве .SETT, .SETF или .SETL. Определить имя, указанное в директиве .IFF или .IFT, как числовое.

?IND—F—.RETURN WITHOUT .GOSUB

Причина. Несовпадение директив .RETURN и .GOSUB.  
Действие. Убедиться, что директивы .GOSUB и .RETURN не нарушают логику программы.

?IND—F—STRING EXPRESSION EXCEEDS LIMIT

Причина. Длина строкового выражения превышает 132 символа; длина строки, получившейся в результате соединения строк, превышает 132 символа; отсутствуют кавычки в конце строкового выражения.

Действие. Убедиться, что длина строкового выражения не превышает 132 символов и строковые выражения заключены в кавычки (где необходимо).

?IND—F—STRING SUBSTITUTION ERROR

Причина. Ошибка во время замены имени его значением.

Действие. Убедиться, что все заменяемые имена определены и заключены в апострофы.

?IND—F—SUBROUTINE NESTING TOO DEEP

Причина. Уровень вложения подпрограмм превышает допустимый.

Действие. Ограничить вложение подпрограмм 8 уровнями.

?IND—F—SYMBOL TABLE OVERFLOW <SYMBOL>

Причина. Таблица имен программы IND заполнена: в таблице нет места для размещения имени <SYMBOL>.

Действие. Удалить определения имен из таблицы имен по директиве .ERASE (см. [1]).

?IND—F—SYMBOL TYPE ERROR <SYMBOL>

Причина. Использован недопустимый тип имени в директиве.

Действие. Ввести соответствующий тип имени.

?IND—F—SYNTAX ERROR

Причина. Синтаксическая ошибка в командной строке.

Действие. Ввести правильную командную строку.

?IND—F—UNDEFINED LABEL <.LABEL>

Причина. Не найдена метка <.LABEL>, указанная в директиве .GOTO, .GOSUB или .ONERR.

Действие. Проверить программу и вставить метку <.LABEL>, где необходимо.

?IND—F—UNDEFINED SYMBOL <SYMBOL>

Причина. Не определено имя <SYMBOL>, указанное в командной строке.

Действие. Убедиться, что используемые имена определены.

ПЕРВИЧНЫЕ СООБЩЕНИЯ ПРОГРАММЫ IND  
ОБ ОШИБКАХ ВВОДА-ВЫВОДА

?IND—F—DATA FILE ERROR

Причина. Аппаратная ошибка обработки директивы .OPEN,

.OPENA, .CLOSE или .DATE или доступа к режиму данных в выходном файле.

Действие. Проверить готовность и исправность оборудования.

?IND—F—ERROR READING FROM TERMINAL

Причина. Ошибка чтения данных с терминала.

Действие. См. ?IND—F—DATA FILE ERROR.

?IND—F—SWAP ERROR

Причина. Ошибка во время операции свопинга.

Действие. Проверить готовность устройства к выполнению операции записи.

?IND—F—TIMEOUT SUPPORT NOT AVAILABLE

Причина. Попытка обработки директивы .ENABLE TIMEOUT, если в системе исключена поддержка ожидания готовности устройства и отсутствует таймер.

Действие. Удалить директиву .ENABLE TIMEOUT из программы, установить в системе поддержку ожидания готовности устройства или выполнить программу в системе, которая имеет таймер.

## ВТОРИЧНЫЕ СООБЩЕНИЯ ПРОГРАММЫ IND ОБ ОШИБКАХ ВВОДА-ВЫВОДА

BAD FILE NAME

Причина. Указана недопустимая спецификация файла.

Действие. Ввести правильную командную строку.

BAD RECORD TYPE — NOT ASCII DATA

Причина. Входной файл содержит неиспользуемые в КОИ-7 символы.

Действие. Ввести правильную командную строку.

DATA OVERRUN

Причина. Попытка ввести на терминал или прочитать с терминала запись длиной более 80 символов.

Действие. Убедиться, что длина введенной строки не превышает 80-ти символов.

DEVICE FULL

Причина. Недостаточно места в справочнике выходного устройства для создания указанного выходного файла. Ошибка появляется при попытке выполнить директиву .OPEN или .CLOSE.

Действие. Увеличить область свободной памяти, удалив драйверы неиспользуемых устройств, а также основное и системные задания.

DEVICE READ ERROR

**Причина.** Ошибка чтения файла.

**Действие.** Проверить готовность и исправность оборудования.  
Проверить том на плохие блоки.

#### DEVICE WRITE ERROR

**Причина.** Ошибка записи файла.

**Действие.** Проверить готовность устройства к выполнению операции записи. Проверить том на плохие блоки.

#### FILE ACCESSED FOR READ

**Причина.** Попытка записи открытого в данный момент файла с помощью директивы .OPENR (директива .OPENR используется только для чтения).

**Действие.** Закрыть или очистить файл перед записью.

#### FILE ACCESSED FOR WRITE

**Причина.** Попытка чтения открытого в данный момент файла с помощью директивы .OPEN или .OPENA (директивы .OPEN и .OPENA используются только для записи открытых файлов).

**Действие.** Закрыть или очистить файл перед чтением.

#### FILE ALREADY OPEN

**Причина.** Попытка открыть файл с тем же номером, что у имеющегося открытого файла.

**Действие.** Использовать другой номер для открытия файла или закрыть открытый файл.

#### FILE EXCEEDS SPACE ALLOCATED

**Причина.** На выходном томе нет места для размещения указанного файла.

**Действие.** Использовать команду монитора SQUEEZE или переключатель DUP /S для сжатия тома. Удалить или переписать на другой том ненужные файлы.  
Использовать другой том.

#### FILE PROTECTED ERROR

**Причина.** Попытка создать файл с таким же именем, как у имеющегося защищенного файла, или уничтожить защищенный файл.

**Действие.** Отменить защиту имеющегося файла по команде UNPROTECT или использовать другое имя для открытия нового файла.

#### INVALID DEVICE OR UNIT

**Причина.** Указано недопустимое устройство в директиве .OPEN, .OPENA или .OPENR.

**Действие.** Ввести правильную командную строку.

#### NO FILE ACCESSED ON CHANNEL

**Причина.** Ошибка при обращении к каналу.

- Действие. Повторить операцию. Если ошибка повторяется, создать новую копию программы IND или системы.  
**NO SUCH FILE**
- Причина. Не найден файл, указанный в директиве .OPEN или .DELETE.
- Действие. Ввести правильную командную строку.  
**UNDEFINED ERROR CODE**
- Причина. Обнаружена ошибка, которую программа IND не может классифицировать.
- Действие. См. **NO FILE ACCESSED ON CHANNEL.**  
**?KMON—F—ADDRESS**
- Причина. Недопустимый адрес в команде E или D. Адрес должен быть в пределах от 0 до начала RMON или до начала драйверов, если они были загружены по команде LOAD.
- Действие. Проверить базу перемещения, если ранее была подана команда B.  
**?KMON—F—AMBIGUOUS COMMAND**
- Причина. Поданная команда определяется неоднозначно.
- Действие. Использовать допустимое сокращение команды.  
**?KMON—F—AMBIGUOUS OPTION**
- Причина. Указанный переключатель определяется неоднозначно.
- Действие. Использовать допустимое сокращение переключателя.  
**?KMON—F—BAD FETCH DEV:**
- Причина. Ошибка чтения драйвера с системного тома.
- Действие. Убедиться, что драйверы устройств описаны в системных таблицах и не содержат плохих блоков.  
**?KMON—F—COMMAND FILE NESTING TOO DEEP**
- Причина. Попытка обращения к косвенному командному файлу на четвертом уровне вложения.
- Действие. Ограничить вложение косвенных командных файлов тремя уровнями.  
**?KMON—F—COMMAND STRING TOO COMPLICATED**
- Причина. Команда слишком сложна для анализа. (Команда содержит большое число переключателей).
- Действие. Упростить команду.  
**?KMON—F—CONFLICTING OPTIONS**
- Причина. Указаны несовместимые переключатели.
- Действие. Ввести правильную командную строку.  
**?KMON—F—COFLICTING SYSGEN OPTIONS**
- Причина. Используются несовпадающие варианты генерации драйвера устройства и RMON.

Действие. Проверить соответствие вариантов генерации драйвера устройства и RMON.

?KMON—F—DEVICE FULL DEV:FILNAM.TYP

Причина. Недостаточно места в справочнике выходного устройства для создания указанного выходного файла.

Действие. Освободить место на томе: удалить или переписать на другой том ненужные файлы, сжать том по команде SQUEEZE. Использовать конструкцию [N] для выходного файла. Использовать другой том.

?KMON—F—DEVICE LOADED OR NOT REMOVABLE DEV:

Причина. Указан недопустимый драйвер устройства (TT:, BA: или системного устройства) в команде REMOVE. Драйвер, указанный в команде REMOVE, является резидентным в памяти.

Действие. Использовать команду UNLOAD перед REMOVE. Использовать команду SHOW для получения информации, какие драйверы резидентны в памяти.

?KMON—F—ERROR IN FILE SPEC

Причина. Ошибка в формате спецификации файла, или спецификация файла отсутствует в командной строке.

Действие. Ввести правильную командную строку.

?KMON—F—EXTENDED MEMORY MONITOR REQUIRED FOR DEV:FILNAM.TYP

Причина. Попытка выполнить программу с оверлеями в расширенной памяти в среде SJ или FB. Установлен разряд виртуального задания в слове состояния задания.

Действие. Выполнить программу под управлением монитора XM в системе с диспетчером памяти.

?KMON—F—FILE NOT FOUND

Причина. Не найден файл, указанный в команде R, RUN, FRUN, SRUN, GET, SET, INSTALL, или косвенный командный файл. Файл не найден на указанном устройстве.

Действие. Ввести правильную командную строку.

?KMON—F—FOREGROUND ACTIVE

Причина. Попытка выполнить команду FRUN или UNLOAD F когда выполняется программа в основной области памяти.

Действие. Ждать завершения работы основной программы, перед тем как выполнить команду FRUN или UNLOAD F.

?KMON—F—INPUT ERROR DEV:FILNAM.TYP

**Причина.** Аппаратная ошибка чтения файла, указанного в команде R, RUN, GET, FRUN или SRUN. Аппаратная ошибка чтения нулевого блока драйвера в команде INSTALL. Ошибка выполнения команды SET для драйвера.

**Действие.** Проверить готовность и исправность оборудования.

#### ?KMON—F—INSUFFICIENT MEMORY

**Причина.** Недостаточно памяти для загрузки файла, указанного в команде GET, R или RUN: при загрузке в память он перекрывает монитор. Размер косвенного командного файла слишком велик. Недостаточно памяти для операций ввода-вывода при обработке команд монитора — например, команды SAVE.

**Действие.** Увеличить область свободной памяти, удалив драйверы неиспользуемых устройств, а также основное и системные задания. Разбить косвенный командный файл на две секции, вставив команду CУ/C в середину файла.

#### ?KMON—F—INSUFFICIENT MEMORY FOR REGION

**Причина.** Недостаточно памяти для создания виртуальной оверлейной области программы, выполняющейся в расширенной памяти.

**Действие.** Удалить драйверы неиспользуемых устройств, а также основное и системные задания. Увеличить число оверлеев в программе, сократив их размер.

#### ?KMON—F—INVALID COMMAND

**Причина.** Подана недопустимая команда KMON. Косвенный командный файл должен быть последним (не считая комментария) в командной строке клавиатурного монитора. Имя косвенного командного файла содержит недопустимый(ые) символ(ы). Подана команда, не принадлежащая командному языку системы, и файл с типом .SAV отсутствует на системном томе.

**Действие.** Ввести правильную командную строку. Убедиться, что во время генерации была исключена поддержка введенной команды. Записать на системный том файл с типом .SAV.

#### ?KMON—F—INVALID CONTINUATION

**Причина.** Попытка продолжить строку невложенного косвенного командного файла на системный терминал.

Действие. Убедитесь, что косвенный командный файл вложенный.

?KMON—F—INVALID CONTROL FILE NESTING

Причина. Недопустимое вложение управляющих файлов.

Действие. Не включать в управляющий файл косвенный командный файл, вызывающий в свою очередь другой управляющий файл.

?KMON—F—INVALID DATE

Причина. Недопустимый аргумент в команде DATE.

Действие. Ввести правильную командную строку.

?KMON—F—INVALID DEVICE DEV:

Причина. Указанное устройство недопустимо, операция недопустима для указанного устройства или загружен драйвер устройства для использования основным заданием (DEV=F), когда система работает под управлением монитора SJ. Попытка установить в системе обслуживание TT: или BA: по команде INSTALL. Попытка удалить устройство из памяти во время выполнения основного задания (DEV:—имя задания) под управлением монитора FB или XM. Подана команда LOAD TT: (для монитора SJ) или LOAD BA: (для любого монитора) при отсутствии соответствующего драйвера (TT.SYS, BA.SYS или BAX.SYS) на системном устройстве.

Действие. Ввести правильную командную строку. Скопировать драйверы устройств TT: или BA: на системный том и перезапустить систему. Удалить устройство из памяти после завершения работы основного задания.

?KMON—F—INVALID DEVICE FOR COMMAND FILE

Причина. Косвенный командный файл вызван с устройства последовательного доступа к данным (PC:, MT:).

Действие. Использовать для косвенного командного файла устройство с произвольным доступом к данным (RK:, DX:).

?KMON—F—INVALID DEVICE INSTALLATION DEV:FILNAM.TYP

Причина. Указан недопустимый адрес регистра состояния или устройство не установлено.

Действие. Убедиться, что устройство установлено.

?KMON—F—INVALID FILE FORMAT DEV:FILNAM.TYP

Причина. Указанный в качестве основного задания файл типа .SAV не является виртуальным заданием в режиме XM. Указанный файл не является файлом типа .REL в режиме FB.

**Действие.** Убедиться, что файл типа .SAV, указанный в качестве основного задания, является виртуальным заданием в режиме XM и виртуальный разряд установлен; также убедиться, что в режиме FB указан файл типа .REL.

**?KMON—F—INVALID LOGICAL JOB NAME**

**Причина.** Подана команда SRUN или FRUN для задания с недопустимым именем или с таким же именем, как у другого выполняющегося задания.

**Действие.** Ввести правильную командную строку (логическое имя задания должно иметь длину не более 6-ти символов, не содержать ни F, ни B). Использовать команду SHOW JOBS или переключатель RESORC /J для получения информации об именах текущих заданий.

**?KMON—F—INVALID NO ON OPTION**

**Причина.** Префикс NO не применим для указанного переключателя.

**Действие.** Проверить поданную команду.

**?KMON—F—INVALID OPTION**

**Причина.** Указан недопустимый переключатель.

**Действие.** Ввести правильную командную строку.

**?KMON—F—INVALID OPTION FOR PROGRAM**

**Причина.** Используемый переключатель относится к другой команде.

**Действие.** Ввести правильную командную строку.

**?KMON—F—INVALID PRIORITY LEVEL**

**Причина.** Уровень приоритета, указанный в переключателе SRUN/LEVEL, недопустим или такой же, как у другого выполняющегося задания.

**Действие.** Ввести правильную командную строку. Использовать команду SHOW JOBS или переключатель RESORC /J для получения информации об уровнях приоритета текущих заданий.

**?KMON—F—INVALID SET PARAMETER**

**Причина.** Указан недопустимый параметр в команде SET. Указан несуществующий параметр. Числовое значение или префикс NO используются в команде или переключателе, где параметры недопустимы.

**Действие.** Ввести правильную командную строку.

**?KMON—F—INVALID TERMINAL**

**Причина.** Указан недопустимый номер устройства в команде SET TT CONSOLE=N, FRUN или в переключателе SRUN/TERMINAL.

Номер устройства является недопустимым в следующих случаях:

- номер устройства превышает допустимый;
- назначен несуществующий номер устройства;
- указанный терминал используется другим заданием;
- указан системный терминал.

Действие. Проверить поданную команду. Использовать команду SHOW для получения информации об установленных устройствах.

#### ?KMON—F—INVALID TIME

Причина. Недопустимый аргумент в команде TIME.

Действие. Ввести правильную командную строку.

#### ?KMON-F-INVALID VALUE SPECIFIED WITH OPTION

Причина. Указан недопустимый аргумент в переключателе.

Действие. Ввести правильную командную строку.

#### ?KMON-F-JOB ACTIVE-CANNOT UNLOAD JOBNAME

Причина. Подана команда UNLOAD во время выполнения задания JOBNAME.

Действие. Ждать завершения работы задания, перед тем как выполнить команду UNLOAD или FRUN. Использовать команду SHOW JOB или переключатель RESORC /J для получения информации о текущих заданиях.

#### ?KMON—F—LINE TOO LONG

Причина. Командная строка или строка косвенного командного файла превышает допустимую длину (максимальная длина — 200. знаков).

Действие. Использовать допустимую длину командной строки.

#### ?KMON—F—MUST TYPE «R BATCH», TYPE «/U»

Причина. Попытка выполнить команду UNLOAD BA без предварительной разгрузки драйвера.

Действие. Вызвать BATCH и использовать переключатель /U для разгрузки драйвера. BATCH автоматически выполняет команду UNLOAD BA.

#### ?KMON—F—NO FILE

Причина. Не указано имя файла.

Действие. Проверить поданную команду.

#### ?KMON—F—NO FOREGROUND JOB

Причина. Попытка выполнить команду SUSPEND, RESUME или UNLOAD, когда в памяти отсутствует основное задание.

Действие. Ввести команду, которая не требует наличия в памяти основного задания.

?KMON—F—NO ROOM

Причина. Нет свободного места в таблицах монитора при попытке установить в системе обслуживание нового устройства (по команде INSTALL) или при попытке присвоить устройству логическое имя (по команде ASSIGN).

Действие. Использовать команду REMOVE или DEASSIGN.  
?KMON—F—NO SUCH JOB — JOBNAME

Причина. Попытка выполнить команду UNLOAD, SUSPEND или RESUME для несуществующего задания JOBNAME.

Действие. Ввести правильную командную строку. Использовать команду SHOW JOBS для получения информации о текущих устройствах.

?KMON—F—NO SUCH JOB OR DEVICE — JOBNAME/DEVICENAME

Причина. Попытка удалить из памяти несуществующее задание JOBNAME или устройство DEVICENAME. Это сообщение выдается только в системах с поддержкой системных заданий.

Действие. Ввести правильную командную строку. Использовать команду SHOW DEVICE для получения информации о текущих заданиях и устройствах.

?KMON—F—OUTPUT ERROR DEV:FILNAM.TYP

Причина. Аппаратная ошибка записи файла в формате .REL во время выполнения команды FRUN, SRUN или SAVE.

Действие. Проверить готовность и исправность оборудования. Убедиться, что запись на устройство разрешена.

?KMON—F—PARAMETERS

Причина. Неправильно указаны параметры команды SAVE. Действие. Проверить формат команды SAVE.

?KMON—F—PROTECTED FILE ALREADY EXISTS DEV:FILNAM.TYP

Причина. Попытка открыть файл с таким же именем, как у имеющегося защищенного файла.

Действие. Использовать другое имя для открытия нового файла.

?KMON—F—SIX SYSTEM JOBS ALREADY RUNNING — CANNOT SRUN JOB

Причина. Попытка выполнить команду SRUN во время выполнения максимального числа заданий — шести.

Действие. Ждать завершения работы задания, перед тем как выполнить команду SRUN.

?KMON—F—TOO MANY FILES

- Причина. Число файлов превышает допустимое.  
 Действие. Проверить команду.  
**?KMON—U—COMMAND FILE I/O ERROR**
- Причина. Ошибка при считывании косвенного командного файла. Возможно аппаратная ошибка.  
 Действие. Проверить готовность и исправность оборудования.  
**?KMON—U—OVERLAY READ ERROR**
- Причина. Аппаратная ошибка чтения оверлейного сегмента KMON. Системный файл монитора имеет плохой блок, или отсутствует системный том.  
 Действие. Проверить готовность и исправность оборудования. Вызвать другой файл монитора по команде BOOT.  
**?KMON—U—SYSTEM INPUT ERROR**
- Причина. Аппаратная ошибка на системном устройстве при выполнении операции свопинга.  
 Действие. Проверить готовность и исправность оборудования.  
**?KMON—U—SYSTEM OUTPUT ERROR**
- Причина. Аппаратная ошибка при выполнении операции свопинга.  
 Действие. Проверить установку защиты записи. Проверить готовность и исправность оборудования.  
**?KMON—U—SYSTEM OUTPUT ERROR DEV:FILNAM.TYP**
- Причина. Ошибка во время вывода указанного системного файла в результате выполнения команды SET.  
 Действие. Проверить установку защиты записи. Проверить готовность и исправность оборудования.  
**?KMON—W—ALREADY INSTALLED/ASSIGNED DEV:**
- Причина. Указанное устройство уже описано в системных таблицах.  
 Действие. Удалить имя устройства из таблицы монитора или отменить логическое имя устройства, чтобы записать в таблицу имя нового драйвера.  
**?KMON—W—DEVICE NOT INSTALLED DEV:**
- Причина. Попытка удалить неописанное в системной таблице устройство.  
 Действие. Проверить командную строку. Использовать команду SHOW для получения информации об установленных драйверах.  
**?KMON—W—LOGICAL NAME NOT FOUND DEV:**
- Причина. В системе не определено логическое имя устройства, указанное в команде DEASSIGN.  
 Действие. Проверить командную строку. Использовать команду SHOW для получения информации о логических

- и физических именах устройств.
- ?KMON—W—NO CLOCK**  
 Причина. В системе нет программируемых часов для команды TIME.  
 Действие. Использовать команду TIME нельзя.
- ?KMON—W—NO DATE**  
 Причина. Не установлена дата.  
 Действие. Ввести дату по команде DATE.
- ?LD—F—/C NOT SPECIFIED ALONE**  
 Причина. Переключатель /C указан с другим переключателем в командной строке, что недопустимо.  
 Действие. Ввести правильную командную строку.
- ?LD—F—DEVICE NOT INSTALLED DEV:**  
 Причина. Указанное устройство не установлено.  
 Действие. Установить указанное устройство по команде INSTALL и вновь ввести команду, либо ввести команду, указывая установленное устройство.
- ?LD—F—FETCH ERROR**  
 Причина. Ошибка при обращении к программе LD. Копия LD, файл монитора или указанный драйвер устройства могут быть повреждены на диске, либо повреждены загруженные в память копии LD или монитора.  
 Действие. Перезагрузить программу LD или систему и вновь выполнить операцию. Если ошибка повторяется, создать новую копию LD и указанный драйвер устройства. Повторить операцию.
- ?LD—F—FILE NOT FOUND DEV:FILNAM.TYP**  
 Причина. Файл не найден на указанном устройстве.  
 Действие. Проверить командную строку.
- ?LD—F—INSUFFICIENT MEMORY**  
 Причина. Недостаточно памяти для выполнения операции.  
 Действие. Увеличить область свободной памяти (удалить ненужные драйверы, удалить основное задание, использовать монитор одного задания). Повторить операцию.
- ?LD—F—INVALID COMMAND**  
 Причина. В командной строке обнаружена ошибка.  
 Действие. Ввести правильную командную строку.
- ?LD—F—INVALID DEVICE DEV:**  
 Причина. Указано несуществующее или недопустимое устройство.  
 Действие. Проверить командную строку. Использовать другое устройство.

**?LD—F—LOOKUP FOR LD FAILED**

Причина. Сбой в работе операционной системы или переименован файл LD.SYS.

Действие. Проверить наличие LD.SYS на системном томе. Если файл LD.SYS отсутствует, скопировать его с дистрибутивного тома на рабочий системный том.

**?LD—F—NO FILE SPECIFIED**

Причина. Не указана спецификация файла.

Действие. Ввести правильную командную строку.

**?LD—F—ONLY 1 OPTION PER SET ALLOWED**

Причина. Указано более одного переключателя в команде SET LDN.

Действие. Ввести правильную командную строку, указав один переключатель для каждой команды SET LDN.

**?LD—F—SPFUN TO LD FAILED**

Причина. Сбой в работе операционной системы.

Действие. Перезагрузить программу LD или систему и повторить операцию. Если ошибка повторяется, создать новую копию LD.SYS и вновь повторить операцию.

**?LD—F—SWITCH VALUE OUT OF RANGE (0—7) /VAL**

Причина. Значение переменной, указанной в переключателе, вне интервала 0—7.

Действие. Ввести правильную командную строку, используя значение переменной от 0 до 7.

**?LD—F—SWITCH WITHOUT REQUIRED VALUE**

Причина. Обнаружено недопустимое значение переключателя.

Действие. Ввести правильную командную строку.

**?LD—F—UNABLE TO PROTECT FILE DEV:FILNAM.TYP**

Причина. Попытка создать защищенный файл, когда запись на выходное устройство запрещена.

Действие. Убедиться, что запись на устройство разрешена. Повторить операцию.

**?LD—F—UNIT (N OF LDN) NOT SPECIFIED**

Причина. Не указан номер логического диска в команде SET LDN FREE или SET LDN [NO] WRITE.

Действие. Указать номер логического диска (N) в LDN. Вновь повторить операцию.

**?LD—F—UNKNOWN SWITCH (/C, /L, /R, OR /W ONLY)**

Причина. Указаны переключатели, отличные от /C, /L, /R или /W.

- Действие. Ввести правильную командную строку.  
 ?LD—F—.WRITW TO LD.SYS FAILED
- Причина. Невозможно выполнение операции, т. к. запрещена запись на системное устройство.
- Действие. Убедиться, что запись на устройство разрешена.  
 Повторить операцию.
- ?LD—W—/W AND /R SPECIFIED FOR SAME UNIT, /W IGNORED
- Причина. Переключатели /W:N и /R:N указаны с одним и тем же номером N.
- Действие. Вновь ввести командную строку, указав различные номера N в переключателях /W и /R.
- ?LIBR—F—EOF DURING EXTRACT
- Причина. Конец входного файла был обнаружен раньше конца выбираемого модуля.
- Действие. Вновь создать библиотечный файл, повторить операцию.
- ?LIBR—F—FILE NOT FOUND DEV:FILNAM.TYP
- Причина. Один из входных файлов, указанных в командной строке, не найден.
- Действие. Исправить и заново ввести командную строку.
- ?LIBR—F—INPUT ERROR IN DEV:FILNAM.TYP
- Причина. Ошибка при чтении входного файла.
- Действие. Проверить готовность и исправность оборудования.  
 Повторить операцию.
- ?LIBR—F—INSUFFICIENT MEMORY
- Причина. Недостаточно оперативной памяти для выполнения операции.
- Действие. Освободить часть оперативной памяти (удалить ненужные драйверы, удалить основное задание, использовать монитор одного задания), повторить операцию.
- ?LIBR—F—INTERNAL ERROR
- Причина. Сбой в работе операционной системы. Возможно запорчен текст LIBR.
- Действие. Повторить операцию. При появлении этой же ошибки получить новую копию программы LIBR.
- ?LIBR—F—INVALID DEVICE DEV:
- Причина. В командной строке указано недопустимое устройство.
- Действие. Проверить и заново ввести командную строку.
- ?LIBR—F—INVALID GSD IN DEV:FILNAM.TYP
- Причина. Ошибка в каталоге глобальных имен (GSD). Файл является неправильным объектным модулем.
- Действие. Вновь протранслировать исходную программу, для

получения правильного объектного модуля и повторить операцию.

**?LIBR—F—INVALID INPUT FILE DEV:FILNAM.TYP**

Причина. Входной файл не является библиотечным.

Действие. Указать правильное имя файла и повторить командную строку.

**?LIBR—F—INVALID LIBRARY FOR LISTING OR EXTRACT**

Причина. Входной файл, предназначенный для операции выборки или получения листинга каталога, не является библиотечным.

Действие. Проверить и заново ввести командную строку.

**?LIBR—F—INVALID OPTION:/X**

Причина. Данный переключатель не является переключателем библиотекаря.

Действие. Исправить и заново ввести командную строку.

**?LIBR—F—INVALID OPTION COMBINATION**

Причина. Указаны переключатели, которые выполняют несовместимые операции. Например, если указан /E, нельзя использовать никакой другой переключатель, если указан /M, то за ним может следовать только переключатель продолжения (/C или //).

Действие. Исключить переключатели, вызывающие ошибку, и повторить операцию.

**?LIBR—F—INVALID RECOD TYPE IN DEV:FILNAM.TYP**

Причина. Тип кода двоичной записи объектного файла не в пределах от 1 до 10 (восьмеричное).

Действие. Вновь протранслировать исходную программу для получения правильного объектного файла, повторить операцию.

**?LIBR—F—MACRO NAME TABLE FULL, UZE/M:N**

Причина. Переполнение каталога имен макрокоманд в макробιβотеке.

Действие. Увеличить размер каталога имен макрокоманд с помощью переключателя /M:N.

**?LIBR—F—NO VALUE ALLOWED :/N**

Причина. В командной строке за переключателем следует аргумент, что недопустимо.

Действие. Исправить и заново ввести командную строку.

**?LIBR—F—OUTPUT AND INPUT FILENAMES THE SAME**

Причина. Входной и выходной файлы, предназначенные для создания макробιβотеки, имеют одинаковые спецификации в командной строке.

Действие. Переименовать входной или выходной файлы.

**?LIBR—F—OUTPUT DEVICE FULL DEV:FILNAM.TYP**

Причина. На указанном устройстве недостаточно места для библиотечного файла или файла листинга каталога библиотечного файла.

Действие. Сжать том по команде SQUEEZE. Удалить или записать на другой том ненужные файлы. Использовать другой том.

?LIBR—F—OUTPUT ERROR DEV:FILNAM.TYP

Причина. Ошибка при записи выходного файла, или заблокирована запись на устройстве вывода.

Действие. Проверить готовность и исправность оборудования. Повторить операцию.

?LIBR—F—OUTPUT FILE FULL

Причина. Выходной файл мал для библиотечного файла или файла листинга.

Действие. Увеличить размер выходного файла по команде LIBR/ALLOCATE:N или с помощью конструкции [:N] в спецификации выходного файла (см. [1]).

?LIBR—F—PROTECTED FILE ALREADY EXISTS DEV:FILNAM.TYP

Причина. Попытка включить в защищенный библиотечный файл новый модуль или создать библиотечный файл, когда существует защищенный файл с тем же именем.

Действие. Отменить защиту файла по команде RENAME/NO-PROTECT или R/P/Z или использовать другое имя для нового библиотечного файла.

?LIBR—F—/R OR /U GIVEN ON LIBRARY FILE DEV:FILNAM.TYP

Причина. В командной строке за спецификацией библиотечного файла следует переключатель /R или /U, что недопустимо.

Действие. Исправить и заново ввести командную строку.

?LIBR—F—/U GIVEN ON LIBRARY FILE DEV:FILNAM.TYP

Причина. Переключатель /U указан за спецификацией библиотечного файла, что недопустимо. Переключатель /U можно указывать только за спецификацией входного файла.

Действие. Исправить командную строку и повторить операцию.

?LIBR—W—DUPLICATE FORM NAME OF FORMNM

Причина. Во входной спецификации даны два файла с одинаковыми именами и после второго имени не указан переключатель /U или /UPDATE. Файл, имя которого встречено первым, включается в библиотечный файл. Все дубликаты имен игнорируются.

Действие. Использовать в командной строке переключатель /U или /UPDATE.

**?LIBR—W—DUPLICATE MACRO NAME OF MACNAM**

Причина. Во входном файле имеются две макрокоманды с одним и тем же именем. Макрокоманда, имя которой встречено первым, заносится в выходной файл. Все дубликаты имен игнорируются.

Действие. Присвоить макрокомандам разные имена.

**?LIBR—W—DUPLICATE MODULE NAME OF AAAAAA**

Причина. В библиотечный файл был включен новый модуль, но он имеет то же имя, что и модуль уже существующий в библиотечном файле. Программа библиотекарь вводит повторно имя этого модуля в каталог.

Действие. Не требуется.

**?LIBR—W—INVALID CHARACTER**

Причина. Введенное имя содержит символ, не относящийся к RADIX-50.

Действие. Исправить командную строку, повторить операцию.

**?LIBR—W—INVALID DELETE OF AAAAAA**

Причина. Удаляемый модуль AAAAAA в каталоге библиотечного файла не существует.

Действие. Не требуется.

**?LIBR—W—INVALID EXTRACT OF AAAAAA**

Причина. Указанное глобальное имя не найдено в каталоге библиотечного файла.

Действие. Исправить командную строку, повторить операцию.

**?LIBR—W—INVALID INSERT AAAAAA**

Причина. Включаемый в библиотечный файл модуль AAAAAA содержит ту же точку входа, что и модуль уже существующий в библиотечном файле. Точка входа игнорируется, но модуль включается в библиотечный файл.

Действие. Не требуется.

**?LIBR—W—INVALID REPLACEMENT OF AAAAAA**

Причина. Заменяемый в библиотечном файле модуль AAAAAA не существует. Модуль игнорируется.

Действие. Не требуется.

**?LIBR—W—NULL LIBRARY**

Причина. Создаваемый библиотечный файл не содержит глобальных имен.

Действие. Входной файл должен иметь, по крайней мере, одно глобальное имя.

**?LIBR—W—ONLY CONTINUATION ALLOWED**

Причина. Введена командная строка без переключателя продолжения после строки, использующей переключатель продолжения.

Действие. Подать правильную команду.

?LINK—F—ADDRESS SPACE EXCEEDED

Причина. Размер программы превышает 32К слов.

Действие. Сократить размер программы, используя оверлейную структуру. Повторить редактирование.

?LINK—F—ASECT TOO BIG

Причина. Наложение абсолютного кода программы на относительный код.

Действие. Сократить размер абсолютной или относительных программных секций. Повторить редактирование.

?LINK—F—/B NO VALUE

Причина. Не указан аргумент переключателя /B.

Действие. Ввести правильную командную строку. Повторить редактирование.

?LINK—F—/B ODD VALUE

Причина. Значение аргумента переключателя /B нечетно.

Действие. Ввести правильную командную строку. Повторить редактирование.

?LINK—F—CROSS REFERENCE DEVICE FULL DEV:FILNAM.TYP

Причина. На томе, используемом для вывода таблицы перекрестных ссылок, недостаточно свободного места или полностью заполнен справочник.

Действие. Освободить место на томе или использовать другой том. Повторить редактирование.

?LINK—F—/E NO VALUE

Причина. Не указан аргумент переключателя /E.

Действие. Ввести правильную командную строку. Повторить редактирование.

?LINK—F—/E ODD VALUE

Причина. Значение аргумента переключателя /E нечетно.

Действие. Ввести правильную командную строку. Повторить редактирование.

?LINK—F—/H NO VALUE

Причина. Не указан аргумент переключателя /H.

Действие. Ввести правильную командную строку. Повторить редактирование.

?LINK—F—/H ODD VALUE

Причина. Значение аргумента переключателя /H нечетно.

Действие. Ввести правильную командную строку. Повторить редактирование.

?LINK—F—/H VALUE TOO LOW

Причина. Значение аргумента переключателя /N:N меньше допустимого.

Действие. Получить карту загрузки редактируемой программы и определить объем памяти, требуемый программе. Повторить редактирование, указав переключатель /N:N с допустимым значением аргумента.

?LINK—F—INPUT ERROR DEV:FILNAM.TYP

Причина. Ошибка при чтении входного файла.

Действие. Проверить готовность и исправность оборудования. Повторить редактирование.

?LINK—F—INSUFFICIENT MEMORY

Причина. Недостаточно памяти.

Действие. Освободить часть оперативной памяти (удалить основное задание, использовать монитор одного задания, удалить ненужные драйверы). Повторить редактирование.

?LINK—F—INTERNAL ERROR

Причина. Ошибка при обращении к программе LINK.

Действие. Повторить редактирование. При повторении этой же ошибки использовать новую копию программы редактор связей.

?LINK—F—INVALID CHARACTER

Причина. Введен знак, неиспользуемый в RADIX-50.

Действие. Ввести правильное имя. Повторить редактирование.

?LINK—F—INVALID COMPLEX RELOCATION IN DEV:FILNAM.TYP

Причина. Ошибка в записи составного перемещения в объектном модуле входного файла.

Действие. Проверить правильность командной строки. Вновь получить объектные модули, входящие во входной файл. Повторить редактирование.

?LINK—F—INVALID DEVICE DEV:

Причина. Указанное устройство не обслуживается системой.

Действие. Установить в системе обслуживание требуемого устройства или использовать другое устройство. Повторить редактирование.

?LINK—F—INVALID GSD IN DEV:FILNAM.TYP

Причина. Ошибка в каталоге глобальных имен объектного модуля входного файла.

Действие. Проверить правильность командной строки. Вновь получить объектные модули, входящие во входной файл. Повторить редактирование.

?LINK—F—INVALID RECORD TYPE IN DEV:FILNAM.TYP

- Причина. Недопустимый формат входного файла.  
 Действие. Проверить правильность командной строки. Вновь получить объектные модули, входящие во входной файл. Повторить редактирование.  
 ?LINK—F—INVALID RLD IN DEV:FILNAM.TYP
- Причина. Ошибка в словаре перемещений объектного модуля входного файла.  
 Действие. Проверить правильность командной строки. Вновь получить объектные модули, входящие во входной файл. Повторить редактирование.  
 ?LINK—F—INVALID RLD SYMBOL IN DEV:FILNAM.TYP
- Причина. В словаре перемещений объектного модуля входного файла используется глобальное имя, не указанное в каталоге глобальных имен этого модуля.  
 Действие. Проверить правильность командной строки. Вновь получить объектные модули, входящие во входной файл. Повторить редактирование.  
 ?LINK—F—/K INVALID VALUE
- Причина. Указано недопустимое значение аргумента переключателя /K.  
 Действие. Ввести правильную командную строку. Повторить редактирование.  
 ?LINK—F—/K NO VALUE
- Причина. Не указан аргумент переключателя /K.  
 Действие. Ввести правильную командную строку. Повторить редактирование.  
 ?LINK—F—LIBRARY EPT TOO BIG, INCREASE BUFFER WITH /G
- Причина. Переполнение таблицы точек входа.  
 Действие. Повторить редактирование, указав переключатель /G в первой командной строке.  
 ?LINK—F—LIBRARY LIST OVERFLOW, INCREASE SIZE WITH /P
- Причина. Переполнение таблицы имен библиотечных программ.  
 Действие. Повторить редактирование, увеличив с помощью переключателя /P размер таблицы имен библиотечных программ.  
 ?LINK—F—/M ODD VALUE
- Причина. В качестве адреса стека указано нечетное значение.  
 Действие. Ввести правильную командную строку. Повторить редактирование.  
 ?LINK—F—MAP DEVICE FULL DEV:FILNAM.TYP
- Причина. На томе, используемом для вывода файла карты загрузки, недостаточно свободного места или полностью заполнен справочник.

Действие. Освободить место на томе или использовать другой том.

?LINK—F—OLD LIBRARY FORMAT IN DEV:FILNAM.TYP

Причина. Формат библиотечного файла не соответствует формату, используемому в ФОДОС-2 версии В03.00.

Действие. Вновь сформировать библиотечный файл с помощью программы библиотекарь. Повторить редактирование.

?LINK—F—PROTECTED FILE ALREADY EXISTS DEV:FILNAM.TYP

Причина. Попытка открыть файл с таким же именем, как у имеющегося защищенного файла.

Действие. Отменить защиту имеющегося файла или использовать другое имя для открытия нового файла. Повторить редактирование.

?LINK—F—/R ODD VALUE

Причина. Значение аргумента переключателя /R нечетно.

Действие. Ввести правильную командную строку. Повторить редактирование.

?LINK—F—REL WRITE BEYOND EOF

Причина. Переполнение файла загрузочного модуля в формате REL при записи информации о перемещениях.

Действие. Повторить редактирование, используя в командной строке конструкцию [N] для задания размера файла загрузочного модуля.

?LINK—F—SAV DEVICE FULL DEV:FILNAM.TYP

Причина. На томе, используемом для вывода файла загрузочного модуля, недостаточно свободного места или полностью заполнен справочник.

Действие. Освободить место на томе или использовать другой том. Повторить редактирование.

?LINK—F—SAV READ ERROR

Причина. Ошибка при считывании формируемого файла загрузочного модуля.

Действие. Проверить готовность и исправность оборудования. Повторить редактирование.

?LINK—F—SAV WRITE ERROR

Причина. Ошибка при записи файла загрузочного модуля.

Действие. Проверить готовность и исправность оборудования. Проверить том на плохие блоки. Повторить редактирование.

?LINK—F—SIZE OVERFLOW OF SECTION AAAAAA

Причина. Размер программной секции превышает 32К слов.

- Действие. Сократить размер программной секции AAAAAA или всей программы. Повторить редактирование.
- ?LINK—F—STB DEVICE FULL DEV:FILNAM.TYP
- Причина. На томе, используемом для вывода файла определений имен, недостаточно свободного места или полностью заполнен справочник.
- Действие. Освободить место или использовать другой том. Повторить редактирование.
- ?LINK—F—STB NOT ALLOWED WITH /S AND A MAP
- Причина. Попытка получить файл определений имен и карту загрузки одновременно с указанием переключателя /S.
- Действие. Отдельно получить файл определений имен или карту загрузки. Повторить редактирование.
- ?LINK—F—STB WRITE ERROR
- Причина. Ошибка при записи файла определений имен.
- Действие. Проверить готовность и исправность оборудования. Проверить том на наличие плохих блоков. Повторить редактирование.
- ?LINK—F—STORING TEXT BEYOND HIGH LIMIT
- Причина. Ошибка во входных объектных модулях или на томе, используемом для вывода файла в формате LDA, недостаточно свободного места.
- Действие. Вновь получить объектные модули и повторить редактирование.
- ?LINK—F—SYMBOL TABLE OVERFLOW
- Причина. Переполнение таблицы имен.
- Действие. Повторить редактирование, указав переключатель /S. При появлении этой же ошибки освободить часть оперативной памяти (удалить основное задание, использовать монитор одного задания, удалить ненужные драйверы) и повторить редактирование.
- ?LINK—F—/T ODD VALUE
- Причина. В качестве адреса точки входа редактируемой программы указано нечетное значение.
- Действие. Ввести правильную командную строку. Повторить редактирование.
- ?LINK—F—TOO MANY PROGRAM SEGMENTS
- Причина. Указано более 1777 (восьмеричное) программных сегментов.
- Действие. Сократить число оверлейных сегментов. Повторить редактирование.
- ?LINK—F—TOO MANY VIRTUAL OVERLAY REGIONS

**Причина.** Указано более 8 оверлейных областей (окон) в расширенной памяти, включая корневой сегмент.

**Действие.** Сократить число оверлейных областей в расширенной памяти. Повторить редактирование.

?LINK—F—/U OR /Y VALUE NOT A POWER OF 2

**Причина.** Значение аргумента переключателя /U или /Y не является степенью числа 2.

**Действие.** Ввести правильную командную строку. Повторить редактирование.

?LINK—F—WORD RELOCATION ERROR IN FILNAM

**Причина.** Попытка обратиться к слову по нечетному адресу во время объединения данных программных секций.

**Действие.** Поместить директиву АССЕМБЛЕРА .EVEN в конце программных секций, содержащих данные, чтобы обеспечить правильность обращения к данным.

?LINK—F—/Y NO VALUE

**Причина.** Не указан аргумент переключателя /Y.

**Действие.** Ввести правильную командную строку. Повторить редактирование.

?LINK—F—/Y ODD VALUE

**Причина.** Значение аргумента переключателя /Y нечетно.

**Действие.** Ввести правильную командную строку. Повторить редактирование.

?LINK—F—VIRTUAL OVERLAY LOGICAL ADDRESS SPACE EXCEEDED

**Причина.** Размер оверлеев превышает 96К слов расширенной памяти.

**Действие.** Повторить редактирование, указав раздел M в переключателе/V:N[:M], который совместно использует несколько сегментов.

?LINK—W—ADDITIVE REFERENCE OF NNNNNN AT SEGMENT  
#MMMMMM

**Причина.** Попытка передать управление в оверлейный сегмент MMMMMM не в точку входа этого сегмента; NNNNNN — точка входа. Редактор связей формирует команду передачи управления непосредственно в точку входа сегмента. Редактирование продолжается.

**Действие.** Исправить исходную программу. Повторить редактирование.

?LINK—W—BOUNDARY SECTION NOT FOUND

**Причина.** В корневом сегменте нет программной секции, к которой относится действие переключателя /Y. Пе-

реключатель /Y игнорируется, редактирование продолжается.

**Действие.** Повторить редактирование, указав имя требуемой программной секции.

?LINK—W—BYTE RELOCATION ERROR AT NNNNNN

**Причина.** Ненулевой старший байт перемещаемой величины. NNNNNN — адрес, по которому находится эта величина. Редактор связей усекает перемещаемую величину до 8 разрядов, если формат загрузочного модуля SAV или LDA. Для загрузочного модуля в формате REL усечение не производится. Редактирование продолжается.

**Действие.** Исправить исходную программу и повторить редактирование.

?LINK—W—COMPLEX RELOCATION DIVIDE BY 0 IN DEV:FILNAM.TYP

**Причина.** Деление на 0 в записи составного перемещения в указанном файле. Результат операции равен 0. Редактирование продолжается.

**Действие.** Исправить исходную программу. Повторить редактирование.

?LINK—W—COMPLEX RELOCATION OF AAAAAA

**Причина.** Попытка составного перемещения глобального имени AAAAAA при редактировании основного задания.

**Действие.** Удалить все составные перемещения. Повторить редактирование.

?LINK—W—CONFLICTING SECTION ATTRIBUTES NNNNNN

**Причина.** При повторном описании программной секции NNNNNN значения признаков отличаются от значений из первого описания. Редактор связей использует признаки из первого определения. Редактирование продолжается.

**Действие.** Исправить исходную программу. Повторить редактирование.

?LINK—W—CROSS REFERENCE INPUT ERROR

**Причина.** Ошибка при чтении таблицы перекрестных ссылок. Часть карты загрузки, содержащая таблицу перекрестных ссылок удаляется, редактирование продолжается.

**Действие.** Проверить готовность и исправность оборудования. Повторить редактирование.

?LINK—W—CROSS REFERENCE OUTPUT ERROR

**Причина.** Ошибка при записи таблицы перекрестных ссылок. Часть карты загрузки, содержащая таблицу пере-

крестных ссылок, удаляется, редактирование продолжается.

Действие. Проверить готовность и исправность оборудования. Повторить редактирование.

?LINK—W—DEFAULT SYSTEM LIBRARY NOT FOUND SYSLIB.OBJ

Причина. Не найден файл системной библиотеки SY:SYSLIB.OBJ.

Действие. Сформировать файл системной библиотеки или исправить исходную программу. Повторить редактирование.

?LINK—W—DUPLICATE SYMBOL 'SYMBOL' DEFINED IN  
DEV:FILNAM.TYP

Причина. Попытка дублировать небиблиотечный файл с помощью переключателя /D.

Действие. Поместить модуль, содержащий глобальное имя 'SYMBOL', в библиотеку объектных модулей. Повторить редактирование.

?LINK—W—DUPLICATE SYMBOL 'SYMBOL' IS FORCED TO THE ROOT

Причина. Модуль, содержащий глобальное имя 'SYMBOL', помещен в корневого сегмент.

Действие. Устранить все обращения к глобальному имени SYMBOL. Повторить редактирование.

?LINK—W—EXTEND SECTION NOT FOUND

Причина. Не найдена программная секция, к которой относится действие переключателя /E. Переключатель /E игнорируется, редактирование продолжается.

Действие. Повторить редактирование, указав имя требуемой программной секции.

?LINK—W—FILE NOT FOUND DEV:FILNAM.TYP

Причина. Не найден указанный входной файл.

Действие. Проверить, существует ли файл с указанным именем. Ввести правильную командную строку. Повторить редактирование.

?LINK—W—INVALID OPTION: /X

Причина. Переключатель /X недопустим или используется в недопустимой комбинации. Если ошибка допущена в первой командной строке, то редактор связей игнорирует эту командную строку. Если ошибка допущена в последующих командных строках, то редактор связей игнорирует только переключатель /X.

Действие. Ввести правильную командную строку. Повторить редактирование.

?LINK—W—LOAD ADDRESS ODD

**Причина.** Указан нечетный адрес загрузки программной секции, к которой относится действие переключателя /Q.

**Действие.** Ввести правильную командную строку. Повторить редактирование.

?LINK—W—LOAD ADDRESS TOO LOW PSECT

**Причина.** Значение адреса загрузки программной секции меньше допустимого.

**Действие.** Повторить редактирование, указав больший адрес загрузки программной секции.

?LINK—W—LOAD SECTION NOT FOUND PSECT

**Причина.** Не найдена указанная секция загрузки в корневом сегменте. Секция игнорируется, редактирование продолжается.

**Действие.** Повторить редактирование, поместив секцию загрузки в корневой сегмент.

?LINK—W—MAP WRITE ERROR

**Причина.** Ошибка при записи файла карты загрузки. Файл карты загрузки не выводится. Редактирование продолжается.

**Действие.** Проверить готовность и исправность оборудования. Проверить том на наличие плохих блоков. Повторить редактирование.

?LINK—W—MULTIPLE DEFINITION OF 'SYMBOL'

**Причина.** Многократное определение глобального имени 'SYMBOL'. Редактор связей использует первое встреченное определение имени 'SYMBOL'. Редактирование продолжается.

**Действие.** Исправить исходную программу. Повторить редактирование.

?LINK—W—NO LOAD ADDRESS

**Причина.** Не указан адрес загрузки программной секции, к которой относится действие переключателя /Q.

**Действие.** Ввести правильную командную строку. Повторить редактирование.

?LINK—W—/O OR /V OPTION ERROR RE—ENTER LINE

**Причина.** Переключатели /O или /V указаны в неправильном порядке или используются в недопустимой комбинации.

**Действие.** Ввести правильную командную строку. Повторить редактирование.

?LINK—W—ROUND SECTION NOT FOUND AAAAAA

**Причина.** В корневом сегменте нет программной секции, к которой относится действие переключателя /U. Пе-

реключатель /U игнорируется, редактирование продолжается.

Действие. Повторить редактирование, указав имя требуемой программной секции.

?LINK—W—STACK ADDRESS UNDEFINED OR IN OVERLAY

Причина. В корневом сегменте не определено глобальное имя, к которому относится действие переключателя /M. Переключатель /M игнорируется, редактирование продолжается.

Действие. Повторить редактирование, указав требуемое глобальное имя.

?LINK—W—TRANSFER ADDRESS UNDEFINED OR IN OVERLAY

Причина. В корневом сегменте не определено глобальное имя, к которому относится действие переключателя /T. Переключатель /T игнорируется, редактирование продолжается.

Действие. Повторить редактирование, указав требуемое глобальное имя.

?LINK—W—UNDEFINED GLOBALS:

AAAAAA

BBBBBB

.  
. .  
. . .  
. . . .

Причина. Глобальные имена AAAAAA, BBBBBB .... не определены.

Действие. Проверить правильность командной строки. Исправить исходную программу. Повторить редактирование.

?MDUP—F—BAD BLOCK IN SYSTEM AREA DEV:

Причина. Найден плохой блок в системной области диска.

Действие. Отформатировать диск и повторить операцию. Если ошибка повторяется, заменить диск.

?MDUP—F—CHANNEL IN USE DEV:FILNAM.TYP

Причина. Ошибка при обращении к каналу.

Действие. Повторить операцию, если ошибка повторяется, создать новую копию программы MDUP или системы.

?MDUP—F—CONFLICTING SYSGEN OPTIONS

Причина. Используются несовпадающие варианты генерации драйвера устройства и монитора.

Действие. Проверить соответствие вариантов генерации.

?MDUP—F—DIRECTORY I/O ERROR DEV:FILNAM.TYP

Причина. Ошибка чтения программой MDUP справочника указанного устройства.

Действие. Проверить готовность и исправность оборудования. Вновь повторить операцию, если ошибка повторяется, отформатировать том.

?MDUP—F—DIRECTORY INPUT ERROR DEV:

Причина. Ошибка чтения справочника указанного устройства.

Действие. Проверить готовность и исправность оборудования. Повторить операцию, если ошибка повторяется, отформатировать том.

?MDUP—F—DIRECTORY OUTPUT ERROR DEV:

Причина. Ошибка записи справочника указанного устройства.

Действие. Проверить готовность и исправность оборудования. Повторить операцию, если ошибка повторяется, отформатировать том.

?MDUP—F—ERROR READING BAD BLOCK REPLACEMENT TABLE DEV:

Причина. Ошибка чтения программой MDUP таблицы замены плохих блоков, расположенной в блоке 1, или нулевого блока указанного тома.

Действие. Проверить готовность и исправность оборудования. Проверить том на плохие блоки. Если нулевой блок плохой, использовать другой том.

?MDUP—F—FILE NOT FOUND

Причина. Указанный файл не найден.

Действие. Ввести правильную командную строку.

?MDUP—F—INPUT ERROR DEV:FILNAM.TYP

Причина. Аппаратная ошибка чтения.

Действие. Проверить готовность и исправность оборудования.

?MDUP—F—INSUFFICIENT MEMORY

Причина. Недостаточно памяти для завершения операции.

Действие. Отвести не меньше 12К слов памяти для требуемой операции (см. [3]).

?MDUP—F—INVALID COMMAND

Причина. Подана недопустимая команда; формат командной строки недопустим; указана недопустимая комбинация переключателей; указано недопустимое для требуемой операции устройство.

Действие. Ввести правильную командную строку.

?MDUP—F—INVALID DEVICE DEV:

Причина. Указанное устройство не поддерживается программой MDUP.

Действие. Ввести правильную командную строку.

?MDUP—F—INVALID DIRECTORY DEV:

Причина. Недопустимая структура справочника для указанного устройства.

Действие. Инициализировать том перед использованием его первый раз. Более подробно об использовании MDUP.SAV см. [3].

?MDUP—F—INVALID OPTION: /X

Причина. Указан недопустимый переключатель.

Действие. Ввести правильную командную строку.

?MDUP—F—INVALID VALUE SPECIFIED WITH OPTION: /X

Причина. С переключателем /X указано недопустимое значение.

Действие. Ввести командную строку, указав допустимое значение для переключателя.

?MDUP—F—NON—BOOTABLE DRIVER DEV: FILNAM. TYP

Причина. Драйвер устройства содержит недопустимый начальный загрузчик, поскольку активная магнитная лента или MDUP.MX построены неверно.

Действие. Обеспечить правильность создания активной ленты или MDUP.SAV (см. [3]).

?MDUP—F—NO ROOM FOR FILE DEV: FILNAM. TYP

Причина. Нет места на выходном томе для размещения указанного файла.

Действие. Использовать команду INITIALIZE или переключатель DUP /Z для инициализации выходного тома перед копированием на него системы.

?MDUP—F—OUTPUT ERROR DEV: FILNAM. TYP

Причина. Аппаратная ошибка во время операции записи.

Действие. Проверить готовность и исправность оборудования.

?MDUP—F—SIZE FUNCTION FAILED

Причина. Ошибка определения размера монтируемого тома. Монитор, драйвер устройства или программа DUP могут быть повреждены.

Действие. Проверить готовность и исправность оборудования. Перезапустить систему и повторить операцию.

?MDUP—F—TRAP TO 4

Причина. Ошибка при выполнении программы MDUP.

Действие. См.

?MDUP—F—CHANNEL IN USE DEV: FILNAM. TYP.

?MDUP—F—TRAP TO 10

Причина. Ошибка при выполнении программы MDUP.

Действие. См.

?MDUP—F—CHANNEL IN USE DEV: FILNAM. TYP.

?MDUP—I—BAD BLOCKS DETECTED NNNNN.

**Причина.** Обнаружено NNNNN. Плохих блоков во время операции сканирования устройства по команде монитора INITIALIZE/BADBLOCKS или переключателю DUP /Z/B.

**Действие.** Не требуется.

?MDUP—I—NO BAD BLOCKS DETECTED DEV:

**Причина.** Не обнаружено плохих блоков во время операции сканирования устройства по команде монитора INITIALIZE/BADBLOCKS или переключателю DUP /Z/B.

**Действие.** Не требуется.

?MDUP—U—SYSTEM ERROR

**Причина.** Ошибка при выполнении программы MDUP.

**Действие.** См.

?MDUP—F—CHANNEL IN USE DEV:FILNAM.TYP.

?MDUP—W—TOO MANY BAD BLOCKS DEV:

**Причина.** Обнаружено более 128 плохих блоков во время сканирования.

**Действие.** Том непригоден для использования и подлежит замене.

?MON—F—BAD FETCH

**Причина.** Ошибка чтения драйвера с системного устройства; недопустимый адрес загрузки драйвера; используются несовпадающие варианты генерации драйвера устройства и RMON.

**Действие.** Убедиться, что адрес загрузки драйвера находится в пределах программы. Проверить соответствие вариантов генерации.

?MON—F—DIR IO ERR NNNNNN

**Причина.** Ошибка при выполнении операции ввода-вывода справочника устройства. NNNNNN — адрес команды, следующей за командой, которая вызвала ошибку, или EMT. Сообщение выдается только мониторами FB и XM.

**Действие.** Проверить готовность и исправность оборудования. Проверить диск на плохие блоки.

?MON—F—DIR OVFLO NNNNNN

**Причина.** Перевыполнение справочника устройства при выполнении запроса .ENTER. NNNNNN — адрес команды, следующей за командой, которая вызвала ошибку, или EMT. Сообщение выдается только мониторами FB и XM.

Действие. Использовать команду, INITIALIZE/SEGMENTS для увеличения числа сегментов справочника. Стереть или переписать на другой том ненужные файлы.

?MON—F—DIRECTORY I/O ERROR NNNNNN

Причина. См. ?MON—F—DIR IO ERR NNNNNN. Сообщение выдается только монитором SJ.

Действие. См. ?MON—F—DIR IO ERR NNNNNN.

?MON—F—DIRECTORY OVERFLOW NNNNNN

Причина. См. ?MON—F—DIR OVFLO NNNNNN. Сообщение выдается только монитором SJ.

Действие. См. ?MON—F—DIR OVFLO NNNNNN.

?MON—F—DIRECTORY UNSAFE

Причина. Ошибка при корректировке справочника устройства программой USR. Монитор пытается завершить операцию со справочником, прежде чем удалить программу из памяти. Возможна потеря информации. Сообщение выдается только мониторами FB и XM.

Действие. Убедиться, что в справочнике нет потери информации. Проверить готовность и исправность оборудования.

?MON—F—FP TRAP NNNNNN

Причина. Произошло прерывание от БПЗ, но программой пользователя не установлен адрес подпрограммы обработки этого прерывания (по .SFPA). Задание удаляется. NNNNNN — адрес команды, следующей за командой, которая вызвала ошибку, или EMT. Сообщение выдается только мониторами SJ.

Действие. Проверить переполнение и антипереполнение данных в формате с плавающей запятой.

?MON—F—FPU TRAP NNNNNN

Причина. См. ?MON—F—FP TRAP NNNNNN. Сообщение выдается только мониторами FB и XM.

Действие. См. ?MON—F—FP TRAP NNNNNN.

?MON—F—INV ADDR NNNNNN

Причина. Адрес, указанный в запросе, нечетный или вне границ задания. NNNNNN — адрес команды, следующей за командой, которая вызвала ошибку, или EMT. Сообщение выдается только мониторами FB и XM.

Действие. Исправить адрес.

?MON—F—INV CHAN NNNNNN

**Причина.** Указан недопустимый номер канала. NNNNNN — адрес команды, следующей за командой, которая вызвала ошибку, или EMT. Сообщение выдается только мониторами FB и XM.

**Действие.** Использовать запрос .CDFN для определения дополнительных каналов ввода-вывода (максимально 225).

?MON—F—INV EMT NNNNNN

**Причина.** Недопустимая команда EMT. NNNNNN — адрес команды, следующей за командой, которая вызвала ошибку, или EMT. Сообщение выдается только мониторами FB и XM.

**Действие.** Проверить функциональный код в команде EMT.

?MON—F—INV SST NNNNNN

**Причина.** Недопустимый адрес прерывания для синхронных системных прерываний. Неправильная инициализация векторов прерываний перед выполнением команды прерывания (BPT, IOT или TRAP), из-за чего монитор помешал нормальному выполнению команды. NNNNNN — адрес команды, следующей за командой, которая вызвала ошибку, или EMT. Драйвер внутренних отображений неправильно восстановил PASC/PPC внутреннего режима или выдал запрос ожидания готовности устройства без запроса .FORK. Сообщение выдается только монитором XM.

**Действие.** Указать допустимый адрес прерывания для синхронных системных прерываний. Правильно инициализировать вектор прерывания перед выполнением команд прерывания. Если в программе отсутствуют команды прерывания, убедиться, что в программе нет логических ошибок, вызывающих непредусмотренные прерывания. Проверить логику драйвера.

?MON—F—INV USR NNNNNN

**Причина.** Попытка загрузить .USR с адреса, находящегося вне границ задания. NNNNNN — адрес команды, следующей за командой, которая вызвала ошибку, или EMT.

**Действие.** Убедиться, что задания, которые выдают программные запросы, требующие USR, обеспечивают область для свопинга USR. Эта область, адрес которой указан в ячейке 46 должна содержать, по

крайней мере, 2К слов памяти и полностью находиться в пределах задания.

?MON—F—INVALID ADDRESS NNNNNN

Причина. См. ?MON—F—INV ADDR NNNNNN. Сообщение выдается только монитором SJ.

Действие. См. ?MON—F—INV ADDR NNNNNN.

?MON—F—INVALID CALL TO USR NNNNNN

Причина. Попытка вызвать USR из подпрограммы завершения. Запрос .SERR этого сообщения не подавляет. NNNNNN — адрес команды, следующей за командой, которая вызвала ошибку, или EMT. Сообщение выдается только монитором SJ.

Действие. Исправить программу.

?MON—F—INVALID CHANNEL NNNNNN

Причина. См. ?MON—F—INV CHAN NNNNNN. Сообщение выдается только монитором SJ.

Действие. См. ?MON—F—INV CHAN NNNNNN.

Причина. См. ?MON—F—INV EMT NNNNNN. Сообщение выдается только монитором SJ.

Действие. См. ?MON—F—INV EMT NNNNNN.

?MON—F—MEM ERR NNNNNN

Причина. Ошибка паритета памяти. Монитор удаляет программу пользователя. NNNNNN — адрес команды, которая вызвала чтение ячейки с плохим паритетом. (Поддержка паритета памяти устанавливается во время генерации системы). Если таких ошибок в программе несколько, память повреждена. Ошибка паритета в системе с КЭШ памятью указывает на сбой основной памяти. Сообщение выдается только мониторами FB и XM.

Действие. Проверить основную память с помощью тест-программ.

?MON—F—MEMORY ERROR NNNNNN

Причина. См. ?MON—F—MEM ERR NNNNNN. Сообщение выдается только монитором SJ.

Действие. См. ?MON—F—MEM ERR NNNNNN.

?MON—F—MMU FAULT NNNNNN

Причина. Обращение к адресу, который находится вне границ отображенной программы. NNNNNN — адрес команды, следующей за командой, которая вызвала ошибку, или EMT. Сообщение выдается только монитором XM.

Действие. Проверить команду, предшествующую команде с адресом NNNNNN и исправить ее.

?MON—F—NO DEV NNNNNN

Причина. Попытка выполнить запросы типа .READ/.WRITE или .LOOKUP при отсутствии в памяти драйвера указанного устройства. NNNNNN — адрес команды, следующей, за командой, которая вызвала ошибку, или EMT. Сообщение выдается только мониторами FB и XM.

Действие. Убедиться, что программный запрос .RELEASE не был выполнен перед запросами типа .READ/.WRITE. Использовать команду монитора LOAD или программный запрос .FETCH для загрузки соответствующего драйвера.

?MON—F—NO DEVICE NNNNNN

Причина. См. ?MON—F—NO DEV NNNNNN. Сообщение выдается только монитором SJ.

Действие. См. ?MON—F—NO DEV NNNNNN.

?MON—F—OVERLAY ERROR NNNNNN

Причина. Ошибка чтения оверлея программой оверлейной структуры. NNNNNN — адрес команды, следующей за командой, которая вызвала ошибку, или EMT. Сообщение выдается только монитором SJ.

Действие. Убедиться, что случайного выполнения запросов .CLOSE и .PURGE по 15 (десятичное) каналу не происходит. Если оверлейная программа выполняется на устройстве, тип которого отличается от типа системного устройства, убедиться, что драйвер устройства загружен, и оно подключено к линии.

?MON—F—OVLY ERR NNNNNN

Причина. См. ?MON—F—OVERLAY ERROR NNNNNN. Сообщение выдается только монитором SJ.

Действие. См. ?MON—F—OVERLAY ERROR NNNNNN.

?MON—F—STACK OVERFLOW

Причина. Прерывание по 4 или 10 векторам. Переполнение стека (указатель стека ниже 400 (восьмеричное)). Сообщение выдается только монитором SJ.

Действие. См. ?MON—F—TRAP TO 4.

?MON—F—SWAP ERROR

Причина. Аппаратная ошибка при записи программы пользователя на диск во время операции свопинга. Сообщение выдается только мониторами FB и XM.

Действие. Проверить готовность устройства к выполнению операции записи.

?MON—F—SYSTEM HALT

**Причина.** Попытка прочитать в память KMON или USR, когда системный том разгружен или файл монитора, находящийся на системном устройстве, имеет плохие блоки. Сообщение выдается только мониторами FB и XM.

**Действие.** Использовать новую копию файла монитора.

?MON—F—SYSTEM READ FAILURE HALT

**Причина.** См. ?MON—F—SYSTEM HALT. Сообщение вы-  
дается только монитором SJ.

**Действие.** См. ?MON—F—SYSTEM HALT.

?MON—F—SYSTEM WRITE ERROR

**Причина.** Попытка записи на системное устройство при ус-  
тановленной защите записи. Возможна аппаратная  
ошибка. Сообщение выдается только монитором  
SJ.

**Действие.** Разрешить запись на диск. Проверить готовность  
и исправность оборудования.

?MON—F—TRAP TO 4 NNNNNN

?MON—F—TRAP TO 10 NNNNNN

**Причина.** Прерывание происходит в следующих случаях:  
задание обращается к недопустимой памяти; ус-  
ройство неспособно передать или принять инфор-  
мацию за отведенное ему время; использована не-  
допустимая команда. Прерывание по 4 вектору  
происходит в следующих случаях: переполнение  
стека; попытка выполнить команду с нечетного  
адреса; произошла аппаратная ошибка; содержи-  
мое счетчика команд (СК) равно 1 или отсутству-  
ет точка входа в директиве .END в программе,  
написанной на языке АССЕМБЛЕР, команде  
монитора LINK/TRANSFER или переключателе /T  
программы LINK. NNNNNN — адрес команды, сле-  
дующей за командой, которая вызвала ошибку,  
или EMT.

**Действие.** Определить границы программы пользователя по  
карте загрузки или по содержимому ячеек 40, 46,  
50 и 54. Если ошибка произошла в пределах про-  
граммы пользователя, проверить программу. Про-  
верить, что программа пользователя не повредила  
области стека, элементов очереди ввода — вывода  
или самого монитора. Проверить правильность  
выполнения свопинга USR. Проверить, нет ли об-  
ращения к несуществующему устройству. Переза-  
грузить систему, если ошибка повторяется, создать

новую копию системы. Убедиться, что точка входа указана в директиве .END, в команде монитора LINK/TRANSFER или переключателе /T.

?MON—F—UNLOADED DRIVER NNNNNN

Причина. Попытка обращения в режиме FB к устройству, драйвер которого отсутствует в памяти и не может быть загружен. Нет места для устройства в таблице монитора в режиме XM. NNNNNN— адрес команды, следующей за командой, которая вызвала ошибку, или EMT.

Действие. Загрузить драйверы устройств по команде LOAD перед выполнением программы. Не загружать драйвер в область памяти, отображенную регистрами PAC1 и PAC3 внутреннего режима, или выполнить генерацию системы, увеличив свободную область в таблице периферийных устройств.

?MSBOOT—F—FILE NOT FOUND

Причина. Указанный файл отсутствует на магнитной ленте.

Действие. Ввести правильную командную строку.

?MSBOOT—F—INVALID FILE NAME

Причина. Указано недопустимое имя файла.

Действие. Ввести правильную командную строку.

?MSBOOT—F—I/O ERROR

Причина. Аппаратная ошибка при выполнении операции загрузки с магнитной ленты.

Действие. Проверить правильность установки тома магнитной ленты.

?MSBOOT—F—LINE TOO LONG

Причина. Длина командной строки превышает 80 символов.

Действие. Проверить поданную командную строку.

?PAT—F—CORRECTION FILE HAS INVALID RECORD

Причина. Формат файла изменений отличен от объектного формата, используемого в ФОДОС.

Действие. Проверить правильность введенной командной строки. Вновь получить файл изменений и повторить операцию.

?PAT—F—CORRECTION FILE MISSING

Причина. Не указана спецификация файла изменений.

Действие. Ввести правильную командную строку.

?PAT—F—CORRECTION FILE MISSING RLD RECORD

Причина. В файле изменений отсутствует каталог перемещений перед первой записью TXT.

Действие. Вновь протранслировать файл изменений и повторить операцию.

#### ?PAT—F—CORRECTION FILE READ ERROR

Причина. Аппаратная ошибка чтения файла изменений.

Действие. Проверить готовность и исправность оборудования.  
Повторить операцию.

#### ?PAT—F—DEVICE FULL DEV:FILNAM. TYP

Причина. На томе, используемом для вывода, нет свободной области для размещения выходного файла.

Действие. Указать в командной строке размер выходного файла; освободить место на томе; использовать для вывода другой том.

#### ?PAT—F—INCOMPATIBLE REFERENCE TO GLOBAL AAAAAA

Причина. В файле изменений неправильно определено глобальное имя AAAAAA.

Действие. Исправить файл изменений, определив глобальное имя AAAAAA в соответствии с определением этого имени в основном модуле. Повторить операцию.

#### ?PAT—F—INCOMPATIBLE REFERENCE TO SECTION AAAAAA

Причина. В файле изменений неправильно определена программная секция AAAAAA.

Действие. Исправить файл изменений, определив программную секцию AAAAAA в соответствии с определением этой секции в основном модуле. Повторить операцию.

#### ?PAT—F—INPUT FILE HAS INVALID RECORD

Причина. Формат основного файла отличен от объектного формата, используемого в ФОДОС.

Действие. Проверить правильность введенной командной строки. Вновь получить основной файл и повторить операцию.

#### ?PAT—F—INPUT FILE MISSING

Причина. Не указана спецификация основного файла.

Действие. Ввести правильную командную строку.

#### ?PAT—F—INSUFFICIENT MEMORY

Причина. Недостаточно оперативной памяти для выполнения операции.

Действие. Освободить часть оперативной памяти (удалить ненужные драйверы, удалить основное задание, использовать монитор одного задания). Повторить операцию.

#### ?PAT—F—INTERNAL ERROR

Причина. Сбой в работе операционной системы. Возможна ошибка в PAT.

Действие. Повторить операцию. При появлении этой же ошибки получить новую копию программы PAT.

**?PAT—F—INVALID COMMAND**

Причина. Ошибка в командной строке.

Действие. Ввести правильную командную строку.

**?PAT—F—INVALID GSD SYMBOL IN DEV:FILNAM. TYP**

Причина. Ошибка в каталоге глобальных имен основного модуля.

Действие. Проверить правильность введенной командной строки. Вновь получить модуль и повторить операцию.

**?PAT—F—INVALID RLD SYMBOL IN DEV:FILNAM. TYP**

Причина. Ошибка в каталоге перемещений основного модуля.

Действие. Проверить правильность введенной командной строки. Вновь получить модуль и повторить операцию.

**?PAT—F—ONLY /C ALLOWED**

Причина. Задан переключатель, отличный от /C[:N].

Действие. Ввести правильную командную строку.

**?PAT—F—OUTPUT ERROR DEV:FILNAM. TYP**

Причина. Заблокирована запись на устройстве вывода; аппаратная ошибка записи выходного файла.

Действие. Проверить готовность и исправность оборудования. Повторить операцию.

**?PAT—F—UNABLE TO LOCATE MODULE AAAAAA**

Причина. В основном файле нет модуля с именем AAAAAA. AAAAAA — имя модуля изменений.

Действие. Определить в файле изменений правильное имя модуля. Повторить выполнение.

**?PAT—F—WRONG VERSION**

Причина. Попытка обращения системы ФОДОС к программе PAT, принадлежащей системе ФОДОС-2.

Действие. Вызвать программу PAT системы ФОДОС.

**?PAT—W—ADDITIONAL INPUT FILE IGNORED**

Причина. Указано более двух входных файлов.

Действие. Не требуется. PAT игнорирует все входные файлы, кроме двух первых.

**?PAT—W—ADDITIONAL OUTPUT FILE ERROR**

Причина. Указано более одного выходного файла.

Действие. Не требуется. PAT игнорирует все выходные файлы, кроме первого.

**?PAT—W—CORRECTION FILE CHECKSUM ERROR**

Причина. Вычисленное значение контрольной суммы модуля изменений не совпадает со значением, указанным в командной строке.

Действие. Не требуется.

?PAT—W—CORRECTION FILE CHECKSUM IS NNNNNN

Причина. Вычисленное значение контрольной суммы модуля изменений равно NNNNNN.

Действие. Не требуется.

?PAT—W—INPUT FILE CHECKSUM ERROR

Причина. Вычисленное значение контрольной суммы основного модуля не совпадает со значением, указанным в командной строке.

Действие. Не требуется.

?PAT—F—INPUT MODULE CHECKSUM IS NNNNNN

Причина. Вычисленное значение контрольной суммы основного модуля равно NNNNNN.

Действие. Не требуется.

?PIP—E—OPERATION NOT COMPLETED

Причина. Прервана операция с переключателем /F, потому что ответ на запрос начинался с «N» или нажато СУ/С.

Действие. Для продолжения операции на тот же самый запрос необходимо напечатать «Y» и нажать клавишу «BK».

?PIP—E—PROTECTED FILE ALREADY EXISTS DEV FILNAM.TYP

Причина. На выходном томе уже существует защищенный файл с именем, указанным в командной строке. Операция создания защищенного выходного файла невозможна. PIP выполнит любую другую операцию, указанную в той же самой командной строке, кроме создания защищенного выходного файла.

Действие. Использовать переключатель /Z для снятия защиты уже существующего файла или воспользоваться другим именем для создания нового файла.

?PIP—F—CHANNEL IN USE

Причина. Ошибка при обращении к программе PIP.

Действие. Перезагрузить систему и повторить операцию. Если ошибка повторится снова, создать новую копию PIP. SAV и повторить операцию.

?PIP—F—CHANNEL NOT OPEN

Причина. См. ?PIP—F—CHANNEL IN USE.

Действие. См. ?PIP—F—CHANNEL IN USE.

?PIP—F—CHECKSUM ERROR DEV: FILNAM.TYP

Причина. Ошибка контрольной суммы во время копирования, используя переключатель /B.

Действие. Проверить введенную командную строку. Повторить операцию по переключателю /G.

**?PIP—F—DEVICE FULL DEV:FILNAM. TYP**

**Причина.** На томе недостаточно места для размещения копируемого файла.

**Действие.** Увеличить размер свободной области на томе; удалить или переписать на другой том ненужные файлы, сжать том по команде SQUEEZE. Использовать конструкцию [N] для выходного файла. Использовать другой том.

**?PIP—F—DEVICE IN USE**

**Причина.** Требуемое устройство (обычно MT:) используется другим заданием.

**Действие.** Повторить операцию после выполнения другого задания.

**?PIP—F—DIRECTORY FULL DEV:**

**Причина.** В справочнике выходного тома нет места для записи выходного файла.

**Действие.** Использовать команду SQUEEZE для объединения неиспользуемых областей выходного тома или скопировать выходной том на том с большим числом сегментов справочника. Повторить команду.

**?PIP—F—DIRECTORY INPUT ERROR DEV:**

**Причина.** Аппаратная ошибка во время чтения справочника.

**Действие.** Проверить исправность и правильность включения аппаратуры. Проверить том на плохие блоки. Повторить операцию. Если ошибка повторяется, скопировать максимально возможное число файлов на другой том. Отформатировать том и повторить операцию. Если ошибка повторяется снова, заменить том другим.

**?PIP—F—FETCH ERROR DEV:**

**Причина.** Ошибка при обращении к программе PIP.

**Действие.** См. ?PIP—F—CHANNEL IN USE.

**?PIP—F—FILE NOT FOUND DEV:FILNAM. TYP**

**Причина.** Указанный входной файл не найден или нет файлов с указанной датой создания, когда используются переключатели /I[:дата], J[:дата] или T[:дата].

**Действие.** Проверить командную строку и сравнить имеющиеся файлы с указанными в командной строке. Повторить операцию.

**?PIP—F—FILE SEQUENCE NUMBER NOT FOUND**

**Причина.** Входной файл с указанным последовательным номером не найден на томе.

**Действие.** Проверить введенную командную строку. Получить

- справочник магнитной ленты по команде монитора DIRECTORY/POSITION и повторить операцию.
- ?PIP—F—INPUT ERROR DEV:FILNAM.TYP
- Причина.** Аппаратная ошибка во время чтения файла.
- Действие.** См. ?PIP—F—DIRECTORY INPUT ERROR DEV:.
- ?PIP—F—INSUFFICIENT MEMORY
- Причина.** Недостаточно памяти для выполнения требуемой операции (обычно, если спецификация файла содержит конструкцию «\*»).
- Действие.** Удалить ненужные драйверы устройств. Удалить основное задание. Использовать монитор SJ. Разрешить свопинг USR по команде монитора SET USR SWAP. Копировать файлы отдельно.
- ?PIP—F—INVALID COMMAND
- Причина.** Ошибка в командной строке.
- Действие.** Ввести правильную командную строку.
- ?PIP—F—INVALID DELETE DEV:FILNAM. TYP
- Причина.** Операция удаления недопустима для магнитных лент.
- Действие.** Не требуется.
- ?PIP—F—INVALID DEVICE DEV:
- Причина.** Указано недопустимое или несуществующее устройство.
- Действие.** Проверить введенную командную строку. Использовать команду монитора SHOW для получения списка устройств, обслуживаемых системой.
- ?PIP—F—INVALID DIRECTORY
- Причина.** Структура справочника тома устройства (метки тома для магнитной ленты) не является структурой ФОДОС-2.
- Действие.** Инициализировать том по команде монитора INITIALIZE.
- ?PIP—F—INVALID OPTION: /X
- Причина.** В командной строке обнаружен недопустимый переключатель.
- Действие.** Ввести правильную командную строку.
- ?PIP—F—INVALID OUTPUT FILE
- Причина.** Недопустимое имя выходного файла для указанной операции.
- Действие.** Проверить и повторить командную строку.
- ?PIP—F—INVALID PROTECT OR UNPROTECT DEV:FILNAM.TYP
- Причина.** Ошибка при обращении (вызове) к программе PIP.

Действие. См. ?PIP—F—CHANNEL IN USE.

?PIP—F—INVALID RENAME DEV:FILNAM. TYP

Причина. Недопустимая операция переименования.

Действие. Проверить введенную командную строку (входной и выходной тома должны совпадать).

?PIP—F—INVALID SET DATE DEV:FILNAM. TYP

Причина. Ошибка при вызове программы PIP.

Действие. См. ?PIP—F—CHANNEL IN USE.

?PIP—F—INVALID USE OF WILDCARDS DEV:FILNAM.TYP

Причина. Спецификация выходного файла содержит встроенную конструкцию «\*» и не соответствует входной спецификации.

Действие. Проверить введенную командную строку.

?PIP—F—INVALID VALUE SPECIFIED WITH OPTION: /X

Причина. Во время копирования с магнитной ленты значение последовательного номера файла в аргументе переключателя /M:N выходит за пределы от —2 до 999.

Действие. Проверить командную строку.

?PIP—F—LIBRARY FILE NOT COPIED DEV:FILNAM.TYP

Причина. Копирование библиотечных файлов по переключателю /B.

Действие. Копировать в режиме отображения памяти.

?PIP—F—OUTPUT ERROR DEV:FILNAM. TYP

Причина. Аппаратная ошибка во время операции записи. На томе недостаточно места для создания файла.

Действие. Проверить исправность и правильность включения аппаратуры. Проверить защиту записи. Проверить диск на плохие блоки. Сжать том по команде монитора SQUEEZE. Удалить или записать на другой том ненужные файлы. Использовать конструкцию [N] для выходного файла. Использовать другой том.

?PIP—F—OUTPUT FILE FULL DEV:FILNAM. TYP

Причина. Выходной файл заполнен; на томе нет места для файла. Обнаружен маркер физического конца тома (для магнитной ленты).

Действие. Сжать том по команде монитора SQUEEZE. Удалить или записать на другой том ненужные файлы. Использовать конструкцию [N] для выходного файла. Использовать другой том.

?PIP—F—PROTECTED FILE DEV:FILNAM.TYP

Причина. Попытка удаления защищенного файла.

Действие. Отменить защиту файла, используя переключатель /Z, и повторить команду.

?PIP—F—SIZE FUNCTION FAILED

Причина. Ошибка при определении размера тома на приводе, который не поддерживается монитором.

Действие. Установить том на привод с номером, который поддерживается монитором. Выполнить генерацию системы, увеличив количество поддерживаемых монитором приводов требуемого устройства.

?PIP—F—SYSTEM ERROR

Причина. Ошибка в системе.

Действие. Перезагрузить систему.

?PIP—F—TOO MANY OUTPUT FILES

Причина. В командной строке определено несколько выходных спецификаций.

Действие. Проверить введенную командную строку.

?PIP—F—VERIFICATION ERROR DEV:FILNAM.TYP

Причина. При использовании переключателя /V после операции копирования обнаружены различия между входными и соответствующими выходными данными.

Действие. Проверить том на плохие блоки. Повторить операцию копирования, используя переключатель /V. Если различия повторяются, заменить выходной том.

?PIP—F—WRONG VERSION

Причина. Попытка обращения системы ФОДОС к программе PIP, принадлежащей системе ФОДОС-2.

Действие. Вызвать программу PIP системы ФОДОС.

?PIP—I—DEVICE FULL DEV:FILNAM. TYP

Причина. При использовании переключателя /V установленный выходной том заполнен, не имеет участка памяти для размещения входных файлов. PIP продолжает копирование и запрещает установку следующего выходного тома.

Действие. Снять заполненный выходной том и установить новый выходной том.

?PIP—I—FILE NOT FOUND

Причина. Указанный файл не найден.

Действие. Не требуется.

?PIP—I—PROTECTED FILE NOT FOUND DEV:FILNAM.TYP

Причина. Указанный файл найден, однако, он уже незащищен.

Действие. Не требуется.

- ?PIP—I—UNPROTECTED FILE NOT FOUND DEV:FILNAM.TYP  
 Причина. Указанный файл найден, однако он уже защищен.  
 Действие. Не требуется.
- ?PIP—W—FILE LARGER THAN OUTPUT DEVICE DEV:FILNAM.TYP  
 Причина. При использовании переключателя /V размер указанного файла больше выходного устройства.  
 Файл не скопирован, операция продолжается.  
 Действие. Использовать для копирования файла тома большего размера
- ?PIP—W—INPUT ERROR DEV:FILNAM. TYP  
 Причина. Аппаратная ошибка во время операции чтения по переключателю /G.  
 Действие. Не требуется. Предупреждающее сообщение.
- ?PIP—W—NO .SYS ACTION  
 Причина. Не указан переключатель /Y. Операция выполняется для всех файлов, кроме файлов с типом .SYS.  
 Действие. Использовать переключатель /Y, если необходимо включить в операцию файлы с типом .SYS. Предупреждающее сообщение.
- ?PIP—W—OUTPUT FILE FOUND, NO OPERATION PERFORMED DEV:FILNAM.TYP  
 Причина. На устройстве вывода уже существует файл с тем же именем, что и выходной файл.  
 Действие. Переименовать файл. Использовать другой том. Предупреждающее сообщение.
- ?PIP—W—REBOOT  
 Причина. Файлы с типом .SYS перемещены, переименованы или удалены на системном томе.  
 Действие. Если это файлы, с которыми в данный момент работает система, перезагрузить систему, иначе — сообщение игнорировать. Предупреждающее сообщение.
- ?PATCH—F—FILE PROTECTED  
 Причина. Попытка модифицировать защищенный файл.  
 Действие. Использовать команду RENAME/NOPROTECT или переключатель PIP /Z для отмены защиты файла.
- ?PATCH—F—INSUFFICIENT MEMORY  
 Причина. Недостаточно оперативной памяти для выполнения операции.  
 Действие. Освободить часть оперативной памяти (удалить ненужные драйверы, удалить основное задание, использовать монитор одного задания). Повторить операцию.
- ?PATCH—F—READ ERROR

Причина. Ошибка чтения.  
 Действие. Проверить готовность и исправность оборудования.  
 Повторить операцию.  
 ?PATCH—F—WRITE ERROR

Причина. Ошибка записи.  
 Действие. Проверить готовность и исправность оборудования.  
 Повторить операцию.  
 ?PATCH—I—[+2K CORE]

Причина. Свопинг программы обслуживания пользователя (USR).  
 Действие. Не требуется.  
 ?PATCH—I—CHECKSUM=NNNNNN

Причина. Вычисленное значение контрольной суммы равно NNNNNN.  
 Действие. Не требуется.  
 ?PATCH—I—SEARCH FAILURE FOR NNNNNN

Причина. Программы PATCH не может обнаружить адрес NNNNNN.  
 Действие. Подать команду правильно.  
 ?PATCH—W—ADDRESS NOT IN SEGMENT

Причина. Указанный в команде адрес не принадлежит сегменту.  
 Действие. Определить границы сегмента по карте загрузки.  
 Подать команду правильно.  
 ?PATCH—W—BOTTOM ADDRESS WRONG

Причина. Содержимое указанного адреса не соответствует первому слову оверлейного драйвера.  
 Действие. Установить соответствующий адрес по команде «X; B».  
 ?PATCH—W—CHECKSUM ERROR

Причина. Вычисленное значение контрольной суммы не совпадает со значением, введенным на запрос CHECKSUM?  
 Действие. Удалить неправильно модифицированный файл.  
 Повторно создать копию файла. Повторить модификацию.  
 ?PATCH—W—ILLEGAL COMMAND

Причина. В командной строке обнаружена ошибка.  
 Действие. Ввести правильную командную строку.  
 ?PATCH—W—ILLEGAL OPTION

Причина. В командной строке обнаружен недопустимый переключатель.  
 Действие. Ввести правильную командную строку.  
 ?PATCH—W—INVALID OVERLAY HANDLER MODIFICATION

- Причина. Попытка ввести нулевое значение в таблицы оверлейного драйвера.
- Действие. Подать команду «X;O» с ненулевым значением X.  
**?PATCH—W—INVALID RELOCATION REGISTER**
- Причина. В команде указан недопустимый номер регистра перемещения.
- Действие. Указать в команде допустимый номер регистра перемещения (от 0 до 7).  
**?PATCH—W—INVALID SEGMENT NUMBER**
- Причина. В модифицированном файле нет сегмента с указанным в команде номером.
- Действие. Определить номер сегмента по карте загрузки. Подать команду правильно.  
**?PATCH—W—MUST OPEN WORD**
- Причина. Не открыто слово перед подачей команды «@», «P» или «X».
- Действие. Открыть слово перед подачей указанных команд.  
**?PATCH—W—MUST SPECIFY SEGMENT NUMBER**
- Причина. Указанный в команде адрес не принадлежит корневому сегменту.
- Действие. Определить по карте загрузки, какому сегменту принадлежит адрес. Подать команду правильно.  
**?PATCH—W—NO ADDRESS OPEN**
- Причина. Не открыто слово или байт перед подачей команды <PC>, «^», «@», «X», «P», «C» или «A».
- Действие. Открыть слово или байт перед подачей указанных команд.  
**?PATCH—W—NOT IN PROGRAM BOUNDS**
- Причина. Адрес, указанный в команде, или начальное значение указателя стека превосходит старший адрес программы, записанный по адресу 50 в нулевом блоке файла.
- Действие. Определить по карте загрузки старший адрес программы и подать команду правильно.  
**?PATCH—W—ODD ADDRESS**
- Причина. Попытка открыть байт с нечетным адресом по команде «/».
- Действие. Открыть байт с нечетным адресом по команде «\».  
**?PATCH—W—ODD BOTTOM ADDRESS**
- Причина. Адрес, указанный в команде «X;B» или содержащийся по адресу 42 в нулевом блоке оверлейного файла, нечетный.
- Действие. Требуемый адрес указать четным.  
**?PATCH—W—PROGRAM HAS NO SEGMENT**

**Причина.** Обращение к сегменту в файле, который не определен в командной строке как оверлейный, или обращение к оверлейному сегменту в файле, который их не имеет.

**Действие.** Определить по карте загрузки структуру файла. Ввести правильную командную строку.

?RESORC—F—ERROR READING DIRECTORY

**Причина.** Ошибка чтения справочника системного устройства.

**Действие.** Проверить готовность и исправность оборудования. Проверить системный том на плохие блоки, используя команды DIRECTORY/BAD BLOCKS или переключатель /K программы DUP.

?RESORC—F—INPUT ERROR FILNAM. TYP

**Причина.** Ошибка чтения указанного файла.

**Действие.** Проверить системный том на плохие блоки.

?RESORC—F—INSUFFICIENT MEMORY

**Причина.** Недостаточно памяти для работы программы RESORC.

**Действие.** Разгрузить основное или системные задания и неиспользуемые в данный момент драйверы.

?RESORC—F—INVALID OPTION

**Причина.** Переключатель, указанный в командной строке, не является переключателем RESORC.

**Действие.** Исправить и вновь ввести командную строку.

?RESORC—F—SYSTEM ERROR

**Причина.** Сбой при работе запроса .DASTATUS или .LOOKUP. Либо устройство не готово к работе, либо монитор, загруженный в память, поврежден.

**Действие.** Проверить готовность и исправность оборудования, перезагрузить монитор.

?RESORC—F—WRONG VERSION

**Причина.** Попытка обращения системы ФОДОС к программе RESORC, принадлежащей к системе ФОДОС-2.

**Действие.** Использовать программу RESORC с системой ФОДОС-2.

?SIPP—E—BELOW SEGMENT BOUNDARY

**Причина.** Попытка открыть ячейку, адрес которой меньше младшего адреса сегмента; SIPP печатает подсказку OFFSET?.

**Действие.** Проверить, чтобы указанный адрес был не меньше младшего адреса сегмента.

?SIPP—E—CHECKSUM ERROR

**Причина.** Указана неправильная контрольная сумма; SIPP печатает подсказку SEGMENT? или BASE? для нахождения и исправления ошибки.

**Действие.** Если указана неправильная контрольная сумма, следует подать СУ/С и <ВК> для возврата к подсказке CHECKSUM?, и ввести правильную контрольную сумму. Если была сделана ошибка во время модификации, следует подать команду «;V» для воспроизведения на терминале внесенных изменений. Исправить ошибку, напечатать СУ/С и <ВК>, а затем ввести правильную контрольную сумму.

#### ?SIPP—E—END OF FILE

**Причина.** Попытка модификации ячейки, адрес которой больше старшего адреса файла.

**Действие.** SIPP печатает подсказку OFFSET?, ранее сделанные изменения не отменяются. Проверить, чтобы указанный адрес ячейки был не больше старшего адреса файла.

#### ?SIPP—E—EXCEEDS PROGRAM LIMIT

**Причина.** Адрес ячейки, которая модифицируется или проверяется, больше старшего адреса программы.

**Действие.** SIPP печатает подсказку OFFSET?, ранее сделанные изменения не отменяются. Ячейка, открытая в данный момент, не модифицируется.

#### ?SIPP—E—EXCEEDS SEGMENT BOUNDARY

**Причина.** Попытка проверить или модифицировать ячейку, адрес которой вне границ сегмента.

**Действие.** SIPP печатает подсказку OFFSET?, ранее сделанные изменения не отменяются.

#### ?SIPP—E—INVALID ADDRESS

**Причина.** Указан недопустимый адрес в качестве адреса сегмента, базы или смещения.

**Действие.** SIPP печатает подсказку OFFSET?. Ввести правильный адрес.

#### ?SIPP—E—INVALID BOUNDARY SIZE

**Причина.** Область поиска превышает размер файла.

**Действие.** Повторно ввести команду поиска и ответить на подсказку, указав правильную область поиска.

#### ?SIPP—E—INVALID COMMAND

**Причина.** Введена недопустимая командная строка.

**Действие.** Ввести правильную командную строку.

#### ?SIPP—E—INVALID EXTENSION OF ROOT SEGMENT

**Причина.** Попытка модификации корневого сегмента вне границ этого сегмента. SIPP печатает подсказку OFFSET?.

**Действие.** Если файл содержит оверлеи в расширенной памяти, размер корневого сегмента может быть увеличен до старшего адреса последнего блока области. Нельзя расширять корневой сегмент, если файл содержит оверлеи в нижней памяти.

?SIPP—E—INVALID INPUT

**Причина.** При вводе данных имела место ошибка. SIPP печатает подсказки.

**Действие.** Проверить введенные данные и повторить операцию.

?SIPP—E—INVALID OPTION: /X

**Причина.** Указан недопустимый переключатель.

**Действие.** Ввести правильную командную строку.

?SIPP—E—INVALID OPTION COMBINATION

**Причина.** Переключатели /C и /D использованы одновременно.

**Действие.** Повторно ввести командную строку, указав лишь один из этих переключателей.

?SIPP—E—INVALID SEARCH COMMAND

**Причина.** Команда или данные, введенные на подсказку SIPP для поиска, недопустимы.

**Действие.** Проверить введенную команду или данные. Повторить операцию.

?SIPP—E—INVALID SEGMENT NUMBER

**Причина.** Указан несуществующий номер сегмента. SIPP повторно печатает подсказку SEGMENT? или START?, если находится в режиме поиска.

**Действие.** Проверить карту загрузки программы для определения и исправления номера оверлейного сегмента.

?SIPP—E—NOT B3 OVERLAID FILE

**Причина.** Модифицируемый файл — файл оверлейной структуры, созданный редактором связей ФОДОС.

**Действие.** Повторно отредактировать программу редактором связей ФОДОС-2 или модифицировать программу по PATCH.

?SIPP—E—OLD BASE ADDRESS

**Причина.** Указан нечетный адрес базы. SIPP вновь печатает подсказку BASE?.

**Действие.** Указать четный адрес базы. Нечетный адрес может быть указан в ответ на подсказку OFFSET?.

?SIPP—E—PATCH BUFFER FULL

**Причина.** Вся память, предназначенная для хранения изменений, использована.

**Действие.** Подать **СУ/С** и **<ВК>** для внесения изменений, напечатанных ранее. Если это сообщение имело место во время расширения файла, сегмента или программы, следует повторно открыть файл и ввести ноль (0) в ячейку, следующую за той, которая вызвала печать этого сообщения об ошибке.

**?SIPP—E—REGION SIZE EXCEEDS**

**Причина.** Модифицируемая программа расширена сверх максимального адреса оверлейной области.

**Действие.** Остановить текущую операцию по команде **СУ/С** и повторно выполнить модификацию, или продолжить текущую операцию, исправив размер этого сегмента до размера указанного в оверлейной таблице и два слова размера для этого сегмента в блоке определения окна драйвера.

**?SIPP—F—DEVICE FULL DEV:FILNAM.TYP**

**Причина.** Устройство вывода не имеет достаточно свободной области для создания выходного файла.

**Действие.** Использовать команду **SQUEEZE** для сжатия диска или использовать другое устройство для этой операции.

**?SIPP—F—FILE NOT FOUND DEV:FILNAM.TYP**

**Причина.** Входного файла нет на указанном устройстве.

**Действие.** Проверить наличие файла на указанном устройстве и ввести повторно командную строку.

**?SIPP—F—INPUT ERROR DEV:FILNAM.TYP**

**Причина.** При чтении модифицируемого файла произошла аппаратная ошибка.

**Действие.** Проверить исправность и готовность оборудования. Повторить операцию.

**?SIPP—F—INSUFFICIENT MEMORY**

**Причина.** Недостаточно памяти для работы программы **SIPP**.

**Действие.** Разгрузить драйверы временно неиспользуемых устройств, удалить основное и системные задания.

**?SIPP—F—INVALID DEVICE DEV:**

**Причина.** Указанное устройство не является устройством прямого доступа.

**Действие.** Использовать допустимое устройство.

**?SIPP—F—OUTPUT ERROR DEV:FILNAM.TYP**

**Причина.** При модификации файла по программе **SIPP** произошла аппаратная ошибка.

Действие. Проверить исправность и готовность оборудования.  
Повторить операцию.

?SIPP—F—PATCH ABORTED, NO MODIFICATION MADE  
Причина. Произошла аппаратная ошибка. Изменения в файл не введены.

Действие. Проверить исправность и готовность оборудования.  
Повторить операцию.

?SIPP—F—PROTECTED FILE ALREADY EXISTS DEV:FILNAM.TYP  
Причина. Попытка создать косвенный командный файл с именем уже существующего защищенного файла.

Действие. Использовать команду монитора UNPROTECT или переключатель /Z программы PIP для отмены защиты файла.

?SIPP—F—SIZE FUNCTION FAILED

Причина. Аппаратная ошибка произошла в то время, когда программа SIPP определяла размер тома.

Действие. Проверить готовность и исправность оборудования.  
Повторить операцию.

?SIPP—I—APPROACHING SEGMENT BOUNDARY

Причина. Программа SIPP превысила верхнюю границу (или старший адрес блока) оверлейного сегмента не более чем на 5 слов.

Действие. Не требуется. Операция продолжается.

?SIPP—I—END OF FILE

Причина. В ответ на подсказку OFFSET? введен адрес больше старшего адреса файла.

Действие. SIPP вновь печатает подсказку OFFSET?. Ввести правильный адрес.

?SIPP—I—EXTENDING HIGH LIMIT

Причина. Попытка изменить код ячейки, адрес которой больше старшего адреса программы.

Действие. Не требуется.

?SIPP—I—EXTENDING OVERLAY SEGMENT

Причина. Попытка изменить код ячейки, адрес которой больше старшего адреса сегмента, но меньше старшего адреса последнего блока оверлейной области.

Действие. Не требуется.

?SIPP—I—NO MODIFICATION MADE

Причина. После создания «Заплаты» или после команды «;V» изменения не были внесены в файл.

Действие. Не требуется.

?SIPP—W—PATCH BUFFER APPROACHING LIMIT

Причина. Переполнение памяти для хранения изменений.

Действие. Можно внести не более пяти изменений. Для рас-

ширения файла следует подать команду `CV/Y` и ввести оставшиеся изменения.

**?SLP—F—AUDIT TRAIL EXCEEDS LINE LIMIT**

Причина. Указана недопустимая комбинация переключателей `/P`, `/S` или `/L`.

Действие. Проверить командную строку.

**?SLP—F—AUDIT TRAIL OVERWRITES LINE**

Причина. Недопустимое расположение признака модификации: он «забывает» текстовую строку.

Действие. Использовать переключатель `/P` для изменения позиции признака модификации.

**?SLP—F—BACKUP FILE SUPPRESSION UNSUCCESSFUL**

Причина. Попытка использовать переключатель `/N` для защиты файла.

Действие. Отменить защиту файла по команде монитора `UNPROTECT` или использовать переключатель `/Z` программы `PIP`.

**?SLP—F—CHARACTERS BEFORE THE «—» IN DEV:FILNAM.TYP**

Причина. В командном файле изменений перед символом дефис (-) находятся символы, которые игнорируются.

Действие. Установить дефис в начало командного файла изменений.

**?SLP—F—COMMAND SYNTAX ERROR**

Причина. Ошибка в командной строке.

Действие. Проверить командную строку. Подать команду правильно.

**?SLP—F—CORRECTION FILE CHECKSUM ERROR**

Причина. Вычисленная контрольная сумма исходного файла, который обновляется, не совпадает со значением указанным в переключателе `/C:N` командной строки программы `SLP`.

Действие. Проверить командную строку. Проверить значение контрольной суммы в переключателе `/C:N` и имя указанного файла. Повторить командную строку.

**?SLP—F—DEVICE FULL DEV:FILNAM.TYP**

Причина. На устройстве недостаточно места для выходного файла.

Действие. Увеличить размер свободной области на томе: удалить ненужные файлы или переписать их на другой том, сжать том по команде `SQUEEZE`.

**?SLP—F—EXTRA FILE(S) SPECIFIED**

Причина. В командной строке указано более двух выходных и двух входных файлов.

Действие. Проверить командную строку.

**?SLP—F—FILE NOT FOUND DEV:FILNAM.TYP**

Причина. Входной файл не найден.  
Действие. Проверить, существует ли файл с указанным именем. Подать команду правильно.

?SLP—F—FILE PROTECTED

Причина. Попытка модификации защищенного файла.  
Действие. Отменить защиту файла по команде монитора RENAME/NOPROTECT или использовать переключатель /Z программы PIP.

?SLP—F—HARD ERROR ON DEVICE

Причина. Ошибка во время выполнения операции ввода-вывода.

Действие. Проверить готовность и исправность оборудования. Повторить операцию.

?SLP—F—INSUFFICIENT MEMORY

Причина. Недостаточно оперативной памяти для выполнения операции.

Действие. Освободить часть оперативной памяти (удалить ненужные драйверы, удалить основное задание, использовать монитор одного задания). Повторить операцию.

?SLP—F—/C INVALID OPTION

Причина. Переключатель /C был указан с выходным файлом в командной строке программы SLP.

Действие. Убедитесь в том, что переключатель /C указан в командной строке программы SLP с входным текстовым файлом программы SLP или с командным файлом.

?SLP—F—INVALID OPTION: /X

Причина. В команде указан недопустимый переключатель.

Действие. Проверить командную строку.

?SLP—F—INVALID RENAME

Причина. Попытка переименовать файл, находящийся на магнитной ленте.

Действие. Операция переименования недопустима для магнитной ленты.

?SLP—F—INVALID VALUE SPECIFIED WITH OPTION: /X

Причина. Недопустимое значение переключателя.

Действие. Подать команду правильно.

?SLP—F—LINE NUMBER ERROR

Причина. Неверный номер строки.

Действие. Проверить командную строку. Подать команду правильно.

?SLP—F—LINE TOO LONG

Причина. Командная строка имеет длину более 80 символов.

Действие. Используйте переключатель /L, чтобы увеличить буфер строки.

?SLP—F—PROTECTED FILE ALREADY EXISTS DEV:FILNAM.TYP

Причина. Попытка создать файл с тем же именем, что и у имеющегося защищенного файла.

Действие. Отменить защиту имеющегося файла по команде монитора RENAME/NOPROTECT или использовать переключатель /Z программы PIP, либо создать файл с другим именем.

?SLP—F—SEARCH FAILURE IN SOURCE FILE

Причина. Не найдена указанная строка во входном текстовом файле.

Действие. Проверить указанную строку и, если необходимо, указать другую строку.

?SLP—F—SOURCE FILE CHECKSUM ERROR

Причина. Вычисленная контрольная сумма исходного файла, который обновляется, не совпадает со значением, указанным в переключателе /C:N командной строки программы SLP.

Действие. Проверить командную строку. Проверить значение контрольной суммы в переключателе /C:N и имя указанного файла. Повторить командную строку.

?SLP—F—SOURCE FILE NOT SPECIFIED

Причина. В командной строке не указан входной текстовый файл.

Действие. Подать команду правильно.

?SLP—F—SYSTEM ERROR

Причина. Системный файл монитора имеет плохие блоки.

Действие. Использовать новую копию файла монитора.

?SLP—F—UNEXPECTED END OF CORRECTION FILE

Причина. Командный файл не содержит строки с признаком конца файла («/»).

Действие. Изменить командный файл, включив строку с признаком конца файла («/»).

?SLP—F—VALUE REQUIRED: /X

Причина. Не указано значение аргумента для переключателя.

Действие. Указать значение аргумента и подать команду правильно.

?SLP—I—DEV:FILNAM.TYP CHECKSUM IS N

Причина. Указан переключатель /C в командной строке программы SLP для определения контрольной суммы входного файла.

Действие. Не требуется.

?SLP—W—AUDIT TRAIL OVERWRITES EXISTING AUDIT TRAIL

Причина. Попытка записи признака модификации в конце строки, уже имеющийся признак модификации. Новый признак модификации «забывает» старый.

Действие. Не требуется.

?SLP—W—NO UPDATE TEXT FOUND

Причина. Командный файл изменений не содержит символ дефис (-). Модификация файла не выполняется и управление передается монитору.

Действие. Установить дефис в командном файле изменений.

?SRCCOM—F—CHANNEL IN USE DEV:FILNAM.TYP

Причина. Сбой в работе операционной системы.

Действие. Перезагрузить программу SRCCOM или систему и повторить операцию. Если ошибка повторится снова, создать новую копию SRCCOM.SAV и повторить операцию.

?SRCCOM—F—CHANNEL NOT OPEN DEV:FILNAM.TYP

Причина. Сбой в работе операционной системы.

Действие. Перезагрузить программу SRCCOM или систему и повторить операцию. Если ошибка повторится снова, создать новую копию SRCCOM.SAV и повторить операцию.

?SRCCOM—F—DEVICE FULL DEV:FILNAM.TYP

Причина. В справочнике выходного тома нет места для записи указанного файла.

Действие. Сжать том по команде SQUEEZE. Удалить или записать на другой том ненужные файлы. Использовать конструкцию [N] для выходного файла. Использовать другой том.

?SRCCOM—F—DEVICE NOT ACTIVE DEV:

Причина. Операция ввода-вывода невозможна на указанном устройстве или томе; устройство отключено; нет тома на устройстве; запрещена запись; устройство не описано в таблицах монитора.

Действие. Проверить правильность включения устройства. Установить обслуживание устройства по команде INSTALL.

?SRCCOM—F—DIRECTORY FULL DEV:

Причина. В справочнике выходного тома нет места для записи файла.

Действие. Сжать том по команде SQUEEZE. Удалить или записать на другой том ненужные файлы. Использовать другой том.

?SRCCOM—F—DIRECTORY I/O ERROR DEV:FILNAM.TYP

Причина. Ошибка чтения справочника указанного устройства.

Действие. Проверить готовность и исправность оборудования.  
Проверить диск на плохие блоки.

?SRCCOM—F—ERROR READING DIRECTORY DEV:

Причина. Аппаратная ошибка чтения справочника указанного устройства.

Действие. Проверить готовность и исправность оборудования.  
Проверить диск на плохие блоки.

?SRCCOM—F—FETCH ERROR DEV:

Причина. Сбой в работе операционной системы.

Действие. Перезагрузить программу SRCCOM или систему и повторить операцию. Если ошибка повторится снова, создать новую копию SRCCOM.SAV и повторить операцию.

?SRCCOM—F—FILE NOT FOUND DEV:FILNAM.TYP

Причина. Файл не найден на указанном устройстве.

Действие. Проверить командную строку.

?SRCCOM—F—ILLEGAL DEVICE DEV:FILNAM.TYP

Причина. В командной строке указано несуществующее или недопустимое устройство.

Действие. Проверить командную строку. Использовать другое устройство.

?SRCCOM—F—INPUT ERROR DEV:FILNAM.TYP

Причина. Аппаратная ошибка доступа к входному устройству.

Действие. Проверить готовность и исправность оборудования.  
Проверить диск на плохие блоки.

?SRCCOM—F—INSUFFICIENT MEMORY

Причина. Недостаточно оперативной памяти для выполнения операции.

Действие. Освободить часть оперативной памяти (удалить ненужные драйверы, удалить основное задание, использовать монитор одного задания). Повторить операцию.

?SRCCOM—F—INVALID COMMAND

Причина. В командной строке обнаружена ошибка.

Действие. Ввести правильную командную строку.

?SRCCOM—F—INVALID DIRECTORY DEV:

Причина. Структура справочника тома устройства не является структурой ФОДОС.

Действие. Использовать справочник тома устройства со структурой ФОДОС.

?SRCCOM—F—INVALID OPTION: /X

Причина. В командной строке обнаружен недопустимый переключатель.

Действие. Ввести правильную командную строку.

?SRCCOM—F—INVALID VALUE SPECIFIED WITH OPTION: /X

Причина. В командной строке обнаружено недопустимое значение переключателя.

Действие. Ввести правильную командную строку, проверив границы допустимых значений переключателя.

?SRCCOM—F—OUTPUT ERROR DEV:FILNAM.TYP

Причина. Аппаратная ошибка записи. На томе недостаточно места для записи файла.

Действие. Проверить командную строку. Проверить готовность и исправность оборудования. Проверить диск на плохие блоки. Сжать том по команде SQUEEZE. Удалить или записать на другой том ненужные файлы.

?SRCCOM—F—OUTPUT FILE FULL DEV:FILNAM.TYP

Причина. На томе нет места для записи файла. Обнаружен маркер физического конца тома (для магнитной ленты).

Действие. Сжать том по команде SQUEEZE. Удалить или записать на другой том ненужные файлы. Использовать конструкцию [N] для выходного файла. Использовать другой том.

?SRCCOM—F—PROTECTED FILE ALREADY EXISTS DEV:FILNAM.TYP

Причина. Попытка создать файл с именем уже существующего защищенного файла.

Действие. Использовать команду UNPROTECT или переключатель /Z программы PIP для отмены защиты существующего файла или использовать другое имя для создания нового файла.

?SRCCOM—F—READ ERROR

Причина. Аппаратная ошибка чтения.

Действие. Проверить готовность и исправность оборудования. Повторить операцию.

?SRCCOM—F—SYSTEM ERROR

Причина. Ошибка в системе.

Действие. Перезагрузить систему.

?SRCCOM—F—RECORD TOO LONG DEV:FILNAM.TYP

Причина. На томе недостаточно места для записи файла.

Действие. Сжать том по команде SQUEEZE. Удалить или записать на другой том ненужные файлы. Использовать конструкцию [N] для выходного файла. Использовать другой том.

?SRCCOM—F—TOO MANY DIFFERENCES

**Причина.** Программа встретила более 310 (восьмеричное) строк различий.

**Действие.** Убедиться, что указана правильная командная строка. Разбить файлы на части для сравнения.

**?SRCCOM—F—UNEXPECTED EOF DEV:FILNAM.TYP**

**Причина.** Ошибка при обращении к программе SRCCOM.

**Действие.** Перезагрузить систему и повторить операцию. Создать новую копию с SRCCOM.SAV и повторить операцию.

**?SRCCOM—F—WRITE ERROR**

**Причина.** Запрещена запись на устройство ввода-вывода; переполнен том устройства вывода; аппаратная ошибка записи.

**Действие.** Разрешить запись на устройство вывода; использовать для вывода другой том или другое устройство; проверить диск на плохие блоки. Повторить операцию.

**?SRCCOM—F—WRONG VERSION**

**Причина.** Попытка обращения системы ФОДОС к программе SRCCOM, принадлежащей системе ФОДОС-2.

**Действие.** Вызвать программу SRCCOM системы ФОДОС.

**?SRCCOM—I—NO DIFFERENCES FOUND**

**Причина.** Различий в файлах не найдено.

**Действие.** Не требуется.

**?SRCCOM—W—AUDIT TRAIL TRUNCATED TO ELEVEN CHARACTERS**

**Причина.** Признак модификации превысил одиннадцать символов и был усечен.

**Действие.** Ввести признак модификации верно, если усеченный признак модификации не подходит.

**?SRCCOM—W—FILES ARE DIFFERENT**

**Причина.** В файлах найдены различия.

**Действие.** Не требуется.

**?VHANDL—F—WINDOW ERROR**

**Причина.** Файл, созданный с помощью переключателя /V, использует слишком много окон.

**Действие.** Повторить редактирование, сократив число окон.

## 5. СООБЩЕНИЯ ПРОГРАММЫ АССЕМБЛЕР

Программа АССЕМБЛЕР печатает коды диагностических сообщений об ошибках во время второго прохода. Код, обозначающий тип ошибки, располагается в начале строки, в которой обнаружена ошибка. Эти коды описаны в начале

данного раздела (см. табл. 3). Остальные сообщения располагаются в соответствии с правилами, приведенными в разделе 1.

Таблица 3

Сообщение	Причина
1	2
<p>A</p> <p>B</p> <p>D</p> <p>E</p> <p>I</p> <p>L</p>	<p>Ошибка адресации или перемещения:</p> <ul style="list-style-type: none"> <li>— превышена допустимая величина смещения в команде условного ветвления (т. е. от —128 (десятичное) до 127 (десятичное))</li> <li>— оператор неправильно изменяет счётчик текущего адреса (например, оператор превысил счётчик текущего адреса за границы директивы .PSECT)</li> <li>— оператор содержит недопустимое выражение, определяющее адрес (например, абсолютное выражение содержит глобальное имя, перемещаемый терм или составной перемещаемый терм).</li> </ul> <p>Директивы АССЕМБЛЕРА .BLKB, .BLKW, и .REPT должны содержать абсолютное значение или выражение</p> <ul style="list-style-type: none"> <li>— использование нескольких выражений, не отделенных друг от друга запятыми</li> <li>— ошибка определения глобального имени</li> </ul> <p>Если исходная программа содержит директиву .ENABLE, АССЕМБЛЕР просматривает в конце первого прохода таблицу имен и предполагает все неопределенные имена глобальными. Если какое-нибудь из этих имен определяется при втором проходе, происходит общая ошибка адресации</p> <ul style="list-style-type: none"> <li>— недопустимые ссылки вперед:             <ol style="list-style-type: none"> <li>1) оператор прямого присваивания имя-выражение содержит ссылку вперед</li> <li>2) выражение, определяющее счётчик адреса, содержит ссылку вперед</li> </ol> </li> <li>— директива АССЕМБЛЕРА в исходной программе содержит недопустимый аргумент, недопустимый ограничитель или недопустимую конструкцию аргументов</li> </ul> <p>Значение счётчика адреса нечетно; во время трансляции значение счётчика адреса увеличивается на 1</p> <p>Обращение к многократно определенному имени</p> <p>Отсутствует директива .END в конце исходной программы. Система завершает текущий проход трансляции</p> <p>Недопустимый символ. Недопустимый символ в листинге заменяется знаком вопроса (?); символ игнорируется, трансляция выполняется</p> <p>Исходная строка содержит более 132 символов (например, при замене формальных параметров фактическими в макрорасширениях)</p>

1	2
M	Множественное определение метки. Метка эквивалентна по первым шести символам ранее встреченной метке
N	Константа содержит цифру, которая не входит в текущую систему счисления программы, константа рассматривается как десятичная
O	Ошибка в поле операции: — директива вне контекста — превышен допустимый уровень вложения для директив условной трансляции — не найдено макроопределение, заданное директивой .MCALL
P	— значение метки меняется от прохода к проходу — в блоке локальных имен появляется множественное определение локального имени
Q	Синтаксическая ошибка в операторе: — пропущен аргумент — указан лишний аргумент — не закончен просмотр оператора
R	Ошибка регистра: — недопустимое обращение к регистру — попытка переопределить стандартное имя регистра без использования директивы .DSABL REG
T	Ошибка усечения: — сформированное число занимает более 16 разрядов — значение выражения содержит более 8 разрядов для директивы .BYTE или команды прерывания (EMT или TRAP)
U	Неопределенное имя. Неопределенному имени присваивается нулевое значение
Z	Выполнение команды различно в ЭВМ «Электроника-60», «ЭЛЕКТРОНИКА-100-25», «ЭЛЕКТРОНИКА-79». (Например, двухадресные команды, имеющие адресацию RN, (RN)+; RN, — (RN); команды JMP и JSR с автоинкрементным методом адресации)

Сообщения, приведенные ниже, указывают на преждевременное прекращение трансляции.

?MACRO—F—DEVICE FULL DEV

Причина. На томе, используемом для вывода, недостаточно свободного места для размещения выходного файла.

Действие. Освободить место на томе или использовать для вывода другой том.

?MACRO—F—FILE NOT FOUND DEV:FILNAM.TYP

Причина. Файл, указанный в командной строке, не найден.  
Действие. Проверить, существует ли файл с указанным именем. Ввести правильную командную строку.

### ?MACRO—F—INCLUDE DIRECTIVE FILE ERROR

Причина. Файл, указанный в директиве .INCLUDE, не существует, или в директиве указана недопустимая спецификация файла. В командной строке указано недопустимое имя устройства. Уровень вложения исходных файлов по директиве .INCLUDE превышает 5.

Действие. Проверить введенную командную строку и ввести допустимую для директивы .INCLUDE спецификацию файла. Вновь ввести командную строку, указав допустимое имя устройства. Убедиться, что уровень вложения исходных файлов не превышает 5.

### ?MACRO—F—INSUFFICIENT MEMORY

Причина. Конфигурация вычислительной системы имеет объем памяти менее 32 Кбайт, необходимый для выполнения программы.

Действие. Увеличить объем свободной памяти (выгрузить драйверы неиспользуемых в данный момент устройств; завершить выполнение основного или системного задания и удалить его; использовать монитор одного задания SJ; разрешить свопинг по команде SET USR SWAP; уменьшить размеры программы, уменьшив максимальное число каналов, открытых одновременно, или разбив программу на небольшие модули для увеличения оверлейного эффекта, или использовав алгоритм, требующий минимальный объем памяти, или записав данные на периферийные устройства).

### ?MACRO—F—INVALID COMMAND

Причина. Командная строка содержит синтаксическую ошибку или более шести спецификаций файлов.

Действие. Ввести правильную командную строку.

### ?MACRO—F—INVALID DEVICE DEV:

Причина. Указанное устройство не обслуживается системой.

Действие. Установить в системе обслуживание требуемого устройства, либо использовать другое устройство.

### ?MACRO—F—INVALID MACRO LIBRARY

Причина. Файл макробιβлиотеки поврежден или создан не бιβлиотекарем системы ФОДОС-2.

Действие. Использовать программу LIBR для создания новой копии SYSMAC.SML.

### ?MACRO—F—INVALID OPTION: /X

Причина. Указан недопустимый переключатель /X.

Действие. Проверить переключатель, указанный в командной строке, и ввести правильную командную строку.

?MACRO—F—I/O ERROR ON DEV:FILNAM.TYP

Причина. Ошибка при считывании или записи указанного файла.

Действие. Проверить исправность и готовность оборудования.

?MACRO—F—I/O ERROR ON WORKFILE

Причина. Ошибка при считывании или записи рабочего файла WRK.TMP. Эта ошибка может произойти при недостаточной области на диске для размещения рабочего файла.

Действие. Проверить исправность и готовность оборудования, и использовать команду SQUEEZE или переключатель /S программы DUP для сжатия тома и размещения рабочего файла.

?MACRO—F—.LIBRARY DIRECTIVE FILE ERROR

Причина. Файл, указанный в директиве .LIBRARY, не существует, или в директиве указана недопустимая спецификация файла. Спецификация файла, указанная в директиве .LIBRARY, содержит имя устройства последовательного доступа. Конфигурация вычислительной системы не содержит устройство, указанное в командной строке. Уровень вложения файлов по директиве .LIBRARY превысил допустимый.

Действие. Проверить введенную командную строку, и ввести допустимую для директивы .LIBRARY спецификацию файла. Убедиться, что спецификация файла, указанная в директиве .LIBRARY, содержит имя устройства произвольного доступа. Ввести новую командную строку, содержащую имя устройства конфигурации данной вычислительной системы. Проверить, чтобы уровень вложения файлов по директиве .LIBRARY не превысил допустимое значение.

?MACRO—F—PROTECTED FILE ALREADY EXISTS DEV-FILNAM TYP

Причина. Попытка создать файл с таким же именем, как у имеющегося защищенного файла.

Действие. Отменить защиту существующего файла по команде монитора UNPROTECT или по переключателю /Z программы PIP, или указать другое имя для нового файла.

?MACRO—F—STORAGE LIMIT EXCEEDED (64K)

**Причина.** Программа требует более 128 Кбайт памяти для хранения таблицы виртуальных имен, в то время как система отводит данной таблице не более 128 Кбайт.

**Действие.** Проверить причины, вызывающие переполнение таблицы виртуальных имен. Разделить исходную программу на независимые модули, и транслировать каждый модуль отдельно.

?MACRO—W—I/O ERROR ON CREF FILE:CREF ABORTED

**Причина.** На выходном томе недостаточно свободного места для выполнения операции, или во время записи на том рабочего файла CREF произошла ошибка ввода-вывода. Вывод файла CREF прерывается, но трансляция продолжается.

**Действие.** Увеличить на томе объем свободной памяти (удалить с тома ненужные файлы; использовать переключатель /ALLOCATE для резервирования нужного количества блоков для выходного файла; сжать том по команде монитора SQUEEZE или по переключателю /S программы DUP; переписать часть сегментов справочника на другой том). Создать несколько логических дисков на томе, используя команды MOUNT и DISMOUNT. Если переполнение тома все еще сохраняется после принятых мер, использовать том большей емкости.

## 6. СООБЩЕНИЯ РЕДАКТОРОВ

Данный раздел содержит сообщения об ошибках редактора текста (EDIT), экранного редактора (K13) и редактора командной строки (SL).

### 6.1. Сообщения редактора текста

Ниже приводятся сообщения, выдаваемые редактором текста (EDIT). Формат сообщений описан в разделе 1.

?EDIT—F—«< >» ERROR; NO COMMAND(S) EXECUTED

**Причина.** Несовпадение количества открывающих и закрывающих угловых скобок, или глубина вложения конструкции «< >» превышает 20.

**Действие.** Подать команду правильно.

?EDIT—F—COMMAND ABORTED

**Причина.** Выполнение команды прервано оператором.

**Действие.** Проверить, не привело ли преждевременное завершение работы команды к нежелательным результатам.

- ?EDIT—F—COMMAND BUFFER FULL; NO COMMAND(S) EXECUTED  
 Причина. Переполнение буфера команд.  
 Действие. Очистить буферы сохранения и МАКРО, если это возможно, или записать часть буфера текста в выходной файл, или представить командную строку в виде нескольких коротких командных строк.
- ?EDIT—F—DIRECTORY FULL  
 Причина. В справочнике тома не хватает места для имени выходного файла. (Сообщение выдается после команд EB и EW, а также команды монитора EDIT).  
 Действие. Удалить ненужные файлы или использовать для вывода другой том.
- ?EDIT—F—END INPUT FILE  
 Причина. Весь входной файл прочитан.  
 Действие. Закрыть файл по команде EX или EF.
- ?EDIT—F—FILE NOT FOUND  
 Причина. Входной файл, указанный в команде, не найден.  
 Действие. Проверить, существует ли файл с указанным именем. Подать команду правильно.
- ?EDIT—F—ILLEGAL ARGUMENT; NO COMMAND(S) EXECUTED  
 Причина. В команде обнаружен недопустимый аргумент.  
 Действие. Подать команду правильно.
- ?EDIT—F—ILLEGAL COMMAND; NO COMMAND(S) EXECUTED  
 Причина. В командной строке обнаружена недопустимая команда.  
 Действие. Проверить формат каждой команды в командной строке. Подать командную строку правильно
- ?EDIT—F—ILLEGAL DEVICE  
 Причина. В системе не установлено обслуживание указанного устройства.  
 Действие. Проверить введенную командную строку. Использовать другое устройство.
- ?EDIT—F—ILLEGAL FILE NAME  
 Причина. В команде обнаружена недопустимая спецификация файла.  
 Действие. Проверить правильность задания спецификации файла. В записи «спф [N]» не должно быть более 19 знаков. Подать команду правильно.
- ?EDIT—F—ILLEGAL MACRO; NO COMMAND(S) EXECUTED  
 Причина. В текстовом аргументе команды M обнаружена команда M или EM, или неправильно использованы ограничители в команде M.  
 Действие. Подать команду M правильно.
- ?EDIT—F—INSUFFICIENT MEMORY

Причина. Не хватает памяти для выполнения команд I, S, U, R, N, C или X.

Действие. Очистить буферы, если это возможно, или записать часть текста в выходной файл.

?EDIT—F—NO FILE OPEN FOR INPUT

Причина. Подана команда R, N, F или P, а входной файл не открыт.

Действие. Проверить командную строку. Подать команду ER или EB.

?EDIT—F—NO FILE OPEN FOR OUTPUT

Причина. Подана команда EF, EX, F, N или W, а выходной файл не открыт.

Действие. Проверить командную строку. Подать команду EW.

?EDIT—F—NOT ENOUGH FREE BLOCKS

Причина. На томе не хватает места для файла указанной величины.

Действие. Выполнить сжатие диска по команде монитора SQUEEZE, если это возможно, или использовать другой том.

?EDIT—F—OUTPUT FILE FULL

Причина. Выходной файл заполнен.

Действие. Закрыть текущий и открытый новый выходной файл.

?EDIT—F—PROTECTED FILE ALREADY EXISTS

Причина. Попытка корректировать защищенный файл с таким же именем как у имеющегося защищенного файла.

Действие. Снять защиту имеющегося файла по команде монитора RENAME/NOPROTECT (переключатель /Z программы PIP), или создать файл с другим именем.

?EDIT—F—PROTECTED .BAK FILE EXISTS

Причина. На томе уже имеется защищенный файл с таким же именем и типом .BAK.

Действие. Выйти из программы EDIT и снять защиту этого файла по команде монитора RENAME/NOPROTECT (переключатель /Z программы PIP). Повторить операцию создания копии файла по команде EB.

?EDIT—F—READ ERROR

Причина. Ошибка чтения.

Действие. Проверить готовность и исправность оборудования. Повторить операцию.

### ?EDIT—F—SEARCH FAILED

**Причина.** Не найден текстовый аргумент, указанный в команде поиска.

**Действие.** Проверить текстовый аргумент. Если была подана команда G, поместить указатель в начало буфера и повторить команду. Если были поданы команды F или P, закрыть файл, открыть его повторно и подать команду вновь.

### ?EDIT—F—SYSTEM I/O ERROR

**Причина.** Сбой аппаратуры во время операции ввода-вывода. Возможно, защита записи на томе.

**Действие.** Проверить готовность и исправность оборудования. Повторить операцию.

### ?EDIT—F—WRITE ERROR

**Причина.** Ошибка записи.

**Действие.** Проверить готовность и исправность оборудования. Проверить диск на наличие плохих блоков, устранить их, если это возможно, или использовать другой том. Повторить операцию.

### ?EDIT—W—COMMAND BUFFER ALMOST FULL

**Причина.** До переполнения буфера команд осталось не более 10 знаков.

**Действие.** Завершить командную строку, используя допустимое число знаков. Выполнить введенную часть командной строки.

### ?EDIT—W—SUPERSEDING EXISTING FILE

**Причина.** Файл с указанным именем уже имеется на устройстве. Если новый файл закрыть, старый будет стерт.

**Действие.** Подать <CY/C><AP2><AP2>, после чего воспользоваться командой монитора REENTER.

**ПРИМЕЧАНИЕ.** Если ошибка обнаружена во время выполнения команды M, редактор выдает сообщения вида <сообщение> IN MACRO

или

<сообщение> IN MACRO; NO COMMAND(S) EXECUTED где <сообщение> — одно из приведенных выше сообщений.

#### 6.2. Сообщения экранного редактора

Сообщения об ошибках экранного редактора (K13) делятся на три группы:

- 1) сообщения, выдаваемые в командном режиме;
- 2) диагностические;
- 3) информационные сообщения и подсказки.

Сообщения приведены в том порядке, в каком они перечислены.

#### ?K13—F—INSUFFICIENT MEMORY

**Причина.** Недостаточно памяти для работы программы — экранному редактору требуется около 16К слов памяти.

**Действие.** Освободить часть оперативной памяти (снять основное задание, использовать монитор SJ, удалить ненужные драйверы). Повторить сеанс редактирования.

#### ?K13—F—I/O OR DEVICE ERROR

**Причина.** Ошибка ввода-вывода или устройства. Возможно, отсутствие готовности или защита записи, или плохие блоки на устройстве ввода-вывода.

**Действие.** Проверить исправность и готовность устройства. Проверить диск на плохие блоки.

#### ?K13—F—OUTPUT FILE SHORTER THAN INPUT FILE

**Причина.** Размер свободной области, отведенной под выходной файл, меньше чем размер входного файла.

**Действие.** Сжать диск по команде монитора SQUEEZE, или использовать другой том выходного устройства.

#### ?K13—F—UNABLE TO ACCESS INPUT DEVICE

**Причина.** Нет доступа к входному устройству. Возможны следующие причины:

- недопустимое устройство;
- нет готовности устройства;
- устройство не включено в систему,
- много плохих блоков на диске;
- не хватает памяти для драйвера устройства

**Действие.** Проверить допустимость, готовность и исправность устройства. Проверить диск на плохие блоки и заменить в случае необходимости. Если устройство не включено в систему, включить его. При необходимости, освободить часть оперативной памяти, удалив ненужные драйверы, сняв основное задание или воспользовавшись монитором SJ.

#### ?K13—F—UNABLE TO ACCESS OUTPUT DEVICE

**Причина.** Возможно, недопустимое устройство, или не загружен драйвер выходного устройства.

**Действие.** Проверить допустимость, готовность и исправность устройства. Проверить диск на плохие блоки. Включить устройство в систему, если оно не включено. При необходимости освободить часть оперативной памяти, удалив ненужные драйверы, сняв

основное задание или воспользовавшись монитором SJ.

ПК13—F—UNABLE TO OPEN INPUT FILE

Причина. Редактор не может открыть входной файл по следующим причинам:

- входного файла нет на томе;
- нет готовности входного устройства.

Действие. Проверить готовность и исправность устройства. Проверить, существует ли файл с указанным именем. Повторить сеанс редактирования.

ПК13—F—UNABLE TO OPEN OUTPUT FILE

Причина. Редактор не может открыть выходной файл по следующим причинам:

- нет готовности устройства (возможно, защита записи);
- свободной памяти для выходного файла недостаточно;
- имеется защищенный файл с таким же именем, как и у входного файла.

Действие. Проверить готовность и исправность устройства. При необходимости, сжать диск по команде SQUEEZE монитора или использовать другое устройство. Указать другое имя для выходного файла.

ПК13—W—ONLY XXXXXX BLOCKS AVAILABLE FOR INSERTIOUS — CONTINUE (Y,N)?

Причина. Свободная область на выходном устройстве позволяет создать выходной файл, который менее чем на 10 блоков (а именно на XXXXXX) превосходит входной файл.

Действие. Напечатать «Y» и <BK>, если сеанс редактирования решено продолжить. Любой другой ответ вызовет прекращение сеанса редактирования и печать «\*» — Редактор ждет команды. При необходимости сжать диск по команде SQUEEZE или использовать другой том.

ПК13—W—OUTPUT FILE EXISTS—CONTINUE (Y, N)?

Причина. На выходном устройстве уже имеется файл с таким же именем, как и у выходного файла.

Действие. Напечатать «Y» и <BK>, если имеющийся файл решено стереть и создать новый. Любой другой ответ сохраняет имеющийся файл. Редактор печатает «\*» и ждет команды.

ПК13—W—OUTPUT FILES PURGED

Причина. Это сообщение выдается после выполнения команды QUIT. Ни основной, ни дополнительный выходной файл, если он был открыт, не сохраняется.

Действие. Не требуется.

PK13—W—UNABLE TO RENAME INPUT FILE TO BAK FILE TYPE

Причина. Редактор не сохранил входной файл с типом .BAK, однако, попытался сохранить выходной файл.

Действие. Не требуется. Если копия входного файла отсутствует, то восстановить файл уже не удастся. Действия запрещены: просмотр.

Причина. Попытка выполнения в режиме просмотра команды или функции, изменяющей файл.

Действия. Воспользоваться командой или функцией, допустимой в режиме просмотра.

Конец файла

Причина. Попытка перемещения курсора, находящегося в конце файла, в результате выполнения функций перемещения курсора, стирания, смены регистров, а также команды WRITE.

Действие. Воспользоваться допустимой командой или функцией.

Начало файла

Причина. Попытка перемещения курсора, находящегося в начале файла, в результате выполнения функций стирания или перемещения курсора.

Действие. Воспользоваться допустимой функцией.

Граница файла

Причина. Попытка перемещения курсора за пределы файла в результате выполнения функций перемещения курсора, в частности, <АБЗАЦ> или <СТРАНИЦА>.

Действие. Воспользоваться допустимой функцией.

Доп. входной файл не открыт

Причина. Команда INCLUDE не выполнена, поскольку не был открыт дополнительный входной файл.

Действие. Открыть дополнительный входной файл по команде OPEN INPUT.

Доп. выходной файл уже есть — заменить?

Причина. На выходном устройстве уже имеется файл с таким же именем, что и у дополнительного выходного файла.

Действие. Если необходимо создать новый файл, уничтожив имеющийся, следует подать «Y» и <BK>. Если

имеющийся файл необходимо сохранить, следует подать любой другой ответ.

Доп. выходной файл уже есть

Причина. Информационное сообщение. Выдается в случае отрицательного ответа на предыдущее сообщение.

Действие. Продолжить сеанс редактирования.

Доп. выходной файл полон

Причина. Переполнение дополнительного выходного файла.

Действие. Закрывать дополнительный выходной файл по команде CLOSE и открыть новый. Объединить полученные выходные файлы можно по команде монитора COPY или во время отдельного сеанса редактирования.

Доп. выходной файл не открыт

Причина. Команда WRITE не выполнена, поскольку не был открыт дополнительный выходной файл.

Действие. Открыть дополнительный выходной файл по команде OPEN OUTPUT.

Граница поиска

Причина. При заданном режиме поиска на странице (SET SEARCH BOUNDED) не найдена указанная для поиска строка текста (<ПОИСК> или <ДАЛЕЕ>).

Действие. Воспользоваться любой допустимой функцией или командой.

Отмена команды

Причина. Пользователь подал СУ/С или СУ/З для отмены команды редактора.

Действие. Воспользоваться любой допустимой функцией или командой.

Команда

Причина. Эта подсказка выдается при использовании функции <КОМАНДА>.

Действие. Подать требуемую команду, завершив ввод клавишей <ВВОД>.

Прекращение операции из-за СУ/С

Причина. Во время обработки команды или функции пользователь подал СУ/С или СУ/З для ее отмены. Редактор обновляет экран, указывая, куда перестился курсор. Команды или функции, поданные во время сообщения «Ждите...», игнорируются.

Действие. Продолжить сеанс редактирования.

СУ/С, СУ/З не идут — подай QUIT

**Причина.** Редактор игнорирует СУ/С и СУ/З во время работы с файлом.

**Действие.** Вставить СУ/С или СУ/З в файл можно с помощью функции <СПЕЦ>. Прекратить сеанс редактирования, не сохраняя открытые выходные файлы, можно по команде QUIT.

Курсор не у цели.

**Причина.** Функции <ОБНОВИТЬ>, <ДОБАВИТЬ>, <БУФЕР> или <ВМЕСТО> не выполнены, поскольку не определен интервал выбора и курсор расположен не на строке, указанной для поиска.

**Действие.** Воспользоваться любой допустимой командой или функцией.

Нет интервала выбора

**Причина.** Интервал выбора не определен или не содержит текста.

**Действие.** Определить интервал выбора правильно. Продолжить сеанс редактирования.

Конец дополнительного входного файла.

**Причина.** При выполнении команды SKIP или INCLUDE дополнительный входной файл прочитан до конца. K13 автоматически закрывает этот файл.

**Действие.** Продолжить сеанс редактирования.

Ошибка чтения доп. входного файла.

**Причина.** Аппаратная ошибка во время чтения дополнительного входного файла

**Действие.** Проверить готовность и исправность входного устройства.

Доп. выходной файл: ошибка записи

**Причина.** Аппаратная ошибка во время записи в дополнительный выходной файл. Возможно наличие плохих блоков на выходном устройстве.

**Действие.** Проверить готовность и исправность устройства. Проверить диск на плохие блоки.

EXIT прервана по СУ/С

**Причина.** Пользователь подал СУ/С во время выполнения команды EXIT.

**Действие.** Если дважды поданные СУ/С не успели отменить EXIT, то K13 выходит в командный режим. Выходные файлы закрыты, переименование файлов (если оно требовалось) выполнено. Если дважды поданные СУ/С отменили EXIT, то на экран выдается часть файла с текущим положением курсора. Продолжить сеанс редактирования.

**FILL** — файл полон

**Причина.** Переполнение выходного или временного файла во время выполнения команды **FILL**.

**Действие.** Закрыть выходной файл и открыть дополнительный выходной файл. Выполнить операцию сжатия диска по команде монитора **SQUEEZE**. Воспользоваться другим выходным устройством.

**INCLUDE** — файл полон

**Причина.** Переполнение файла во время выполнения команды **INCLUDE**. Курсор располагается правее последнего считанного знака.

**Действие.** Закрыть выходной файл и открыть дополнительный выходной файл. Выполнить операцию сжатия диска по команде монитора **SQUEEZE**. Воспользоваться другим выходным устройством.

Ошибка закрытия доп. выходного файла

**Причина.** Ошибка во время закрытия дополнительного выходного файла — возможно, обнаружен плохой блок.

**Действие.** Проверить диск на плохие блоки. Воспользоваться другим выходным устройством для дополнительного выходного файла.

Это не команда

**Причина.** Недопустимая команда.

**Действие.** Воспользоваться любой допустимой командой или функцией.

Страница или абзац заданы неверно.

**Причина.** В команде **SET ENTITY PAGE** или **SET ENTITY SECTION** допущена ошибка задания параметра.

**Действие.** Подать команду правильно. Продолжить сеанс редактирования.

Имя файла задано неверно

**Причина.** Недопустимая спецификация файла в команде **OPEN INPUT** или **OPEN OUTPUT**.

**Действие.** Подать команду правильно. Продолжить сеанс редактирования.

Это не функция

**Причина.** Указанная функция не является допустимой функцией **K13**; вместе с тем, непосредственно вставить введенные символы в буфер нельзя.

**Действие.** Символы, которые нельзя вставлять непосредственно, надо вставлять с помощью функции **<СПЕЦ>**. Продолжить сеанс редактирования, подавая допустимые команды и функции.

- Недопустимая правая граница  
Причина. Указанное в команде SET WRAP значение превышает 256.  
Действие. Подать команду правильно.
- Отступ задан неверно  
Причина. В команде SET TABS задано нулевое или отрицательное значение абзацного отступа.  
Действие. Подать команду правильно.
- Поиск — неверный ввод  
Причина. Недопустимые символы в образце поиска, или ввод образца поиска завершается нажатием клавиш, отличных от <ВПЕРЕД> или <НАЗАД>.  
Действие. Указать образец для поиска правильно.
- Ввод команды завершен неверно  
Причина. Недопустимый символ в команде, или ввод команды завершен нажатием клавиши, отличной от клавиши <ВВОД>.  
Действие. Подать команду правильно.
- Вставка 0 запрещена  
Причина. Попытка вставить нулевой код КОИ-7 с помощью функции <СПЕЦ>.  
Действие. Воспользоваться любой допустимой командой или функцией.
- INCLUDE — конец доп. входного файла  
Причина. Во время выполнения команды INCLUDE дополнительный входной файл прочитан до конца. K13 закрывает его.  
Действие. Продолжить сеанс редактирования.
- Вставка — файл полон  
Причина. Переполнение файла в процессе вставки текста.  
Действие. Закрывать выходной файл и открыть новый, или удалить из файла ненужный текст, если он есть. Сжать диск по команде монитора SQUEEZE, или воспользоваться другим выходным устройством.
- Работа прервана.  
Причина. Выполнение команды LOCAL прервано по СУ/С или из-за сбоя программы.  
Действие. Продолжить сеанс редактирования.
- ADJUST — ошибка параметра  
Причина. Указанный в команде TABS ADJUST параметр выходит за пределы диапазона [— 50, +50].  
Действие. Подать команду правильно.
- LOCAL — ошибка параметра

**Причина.** Указанное в команде LOCAL начальное значение меньше 0 или больше 32767.

**Действие.** Подать команду правильно.

Сбой программы K13

?K13—F—сбой программы K13

**Причина.** Возможно, аппаратная ошибка, или заперчен загрузочный модуль программы K13.

**Действие.** Проверить готовность и исправность оборудования. Использовать другой экземпляр загрузочного модуля.

Буфер МАКРО полон

**Причина.** Функции и команды, введенные после команды LEARN, вызвали переполнение буфера МАКРО. Действие команды LEARN немедленно прекращается и буфер МАКРО очищается.

**Действие.** При определении макрокоманд следует:

— использовать функции <вперед> и <назад> лишь там, где это необходимо;

— использовать допустимые сокращения команд;

— образцы для поиска задавать как можно более короткими.

Локальная метка: более 6 цифр

**Причина.** Во время выполнения команды LOCAL обнаружена локальная метка, содержащая более 6 цифр.

**Действие.** Проверить локальные метки.

Ищу:

**Причина.** Редактор выдает эту подсказку, когда оператор использовал функцию <ПОИСК>.

**Действие.** Указать образец поиска — строку текста, содержащую от 1 до 46 знаков, — и направление поиска (функция <ВПЕРЕД> или <НАЗАД>).

МАКРО не задана

**Причина.** Буфер МАКРО пуст, т. к. пользователь не подавал команду LEARN, или при определении МАКРО произошло переполнение буфера МАКРО.

**Действие.** Подать команду LEARN для определения МАКРО.

Цель не указана

**Причина.** Функции <ДАЛЕЕ>, <ЗАМЕНИТЬ> или <ВМЕСТО> не были выполнены, т. к. пользователь не указал образец для поиска.

**Действие.** Указать допустимый образец для поиска в функции <ПОИСК>, завершив ввод указанием направления поиска — <ВПЕРЕД> или <НАЗАД>.

### Файл расширить нельзя

**Причина.** Свободная память, отведенная под выходной файл, не позволяет завершить выполнение функций, которые увеличивают выходной файл — <БУФЕР>, <ВОСТЗНАК>, <ВОСТСЛОВ> или <ВОСТРК>. Курсор располагается правее последнего символа, вставка которого завершилась успешно.

**Действие.** Вывести часть информации в дополнительные выходные файлы, если это возможно, удалить ненужную информацию и завершить формирование выходного файла. Сжать диск по команде монитора SQUEEZE или воспользоваться другим томом выходного устройства.

### Стираний не было

**Причина.** Функция восстановления текста <ВОСТРК>, <ВОСТСЛОВ> или <ВОСТЗНАК> — не была выполнена, т. к. соответствующий буфер пуст.

**Действие.** Продолжить сеанс редактирования.

### Повтор

**Причина.** Это сообщение выдается при задании многократного повторения функции (<СЛУЖ>NNN) или во время выполнения функции <СПЕЦ>.

**Действие.** Указать необходимое число повторений или код вставляемого символа (десятичное значение).

### Отмена поиска

**Причина.** Пользователь указал СУ/С или СУ/З для отмены функции <ПОИСК> и сообщения «Ищу:».

**Действие.** Продолжить сеанс редактирования.

### Интервал выбора очень велик

**Причина.** Буфер сбора не может вместить весь текст, входящий в интервал выбора. Функция <ОБНОВИТЬ> или <ДОБАВИТЬ> не выполнена.

**Действие.** Задавать более короткие интервалы выбора, а также использовать дополнительные файлы для временного хранения текста.

### TAB — недопустимый аргумент

**Причина.** Сообщение выдается при выполнении функции <СЛУЖ>А. Номер знакоместа, на котором расположен курсор, должен быть кратен величине базового абзацного отступа.

**Действие.** Проверить правильность задания величины абзацного отступа и расположения курсора. Повторить функцию <СЛУЖ>А.

Не было SET TABS

**Причина.** Команда TABS ADJUST подана раньше, чем команда SET TABS.

**Действие.** Подать команды в другом порядке — сначала SET TABS, затем TABS ADJUST.

Цель не найдена

**Причина.** При выполнении функции <ПОИСК> или <ДАЛЕЕ> курсор дошел до границы файла, однако, искомая строка текста не найдена.

**Действие.** Продолжить сеанс редактирования.

Лишние аргументы в команде

**Причина.** Пользователь указал в команде редактора лишние аргументы.

**Действие.** Подать команду правильно.

Нет доступа к доп. устройству

**Причина.** Возможны следующие причины:

- недопустимое устройство;
- нет готовности устройства;
- устройство не включено в систему;
- много плохих блоков на диске;
- не хватает памяти для драйвера устройства.

**Действие.** Проверить допустимость, готовность и исправность устройства. Проверить диск на плохие блоки. Включить устройство в систему, если оно не включено. При необходимости освободить часть оперативной памяти, удалив ненужные драйверы, сняв основное задание или воспользовавшись монитором SJ.

Не закрывается доп. файл

**Причина.** Редактор не может закрыть дополнительный выходной файл. Возможна неисправность выходного устройства или отсутствие его готовности. Возможно, сбой программы K13.

**Действие.** Проверить готовность и исправность выходного устройства. Проверить диск на плохие блоки. Освободить часть оперативной памяти. Воспользоваться другим экземпляром программы K13.

Не открывается доп. входной файл.

**Причина.** Отсутствует готовность входного устройства, или файл с указанным именем отсутствует на устройстве.

**Действие.** Проверить наличие файла с указанным именем на выходном устройстве. Проверить готовность устройства, а также, включено ли оно в систему.

Не открывается доп. выходной файл

Причина. Отсутствие готовности устройства (возможно, за-  
щита записи).

Действие. Проверить готовность и исправность дополнитель-  
ного выходного устройства.

Буфер стирания полон

Причина. Выполнение функции <СТИРСТ> или <СТИР-  
СЛОВ> не закончено, поскольку буфер строки  
(132 знака) или буфер слова (80 знаков) не вме-  
щает, соответственно, стираемую строку или слово.

Действие. Длинные строки или слова можно удалять повтор-  
ным применением функций <СТИРСТ> или  
<СТИРСЛОВ>, игнорируя сигналы об ошибках.  
Для этого можно также использовать буфер сбо-  
ра или дополнительные файлы.

LOCAL — курсор не на месте

Причина. Между текущим положением курсора и началом  
файла (или знаком перевода формата) нет гло-  
бальных имен или директивы .ENABLE LSB.

Действие. Проверить, находится ли курсор на первой страни-  
це блока глобальных имен. Подать команду  
LOCAL.

Ждите...

Причина. Это сообщение выдается в том случае, когда функ-  
ция или команда выполняется дольше, чем обык-  
но. Время выполнения операции зависит от харак-  
тера операции, величины файла, быстродействия  
устройств и т. п.

Действие. Не требуется.

Переполнение файла

Причина. Выполнение функции <ВСТАВИТЬ СТРОКУ>  
вызывает переполнение файла.

Действие. Вывести часть информации в дополнительные вы-  
ходные файлы, если это возможно, удалить ненуж-  
ную информацию и завершить формирование вы-  
ходного файла. Сжать диск по команде монитора  
SQUEEZE или воспользоваться другим томом вы-  
ходного устройства.

Недопустимый аргумент

Причина. Отрицательный аргумент в команде INCLUDE,  
SKIP или WRITE.

Действие. Подать команду правильно.

### 6.3. Сообщения редактора командной строки

Ниже приводятся сообщения, выдаваемые редактором командной строки (SL).

Начало строки

Причина. Попытка переслать курсор влево, когда курсор находится в начале строки.

Действие. Ввести правильную командную строку.

Конец строки

Причина. Попытка переслать курсор вправо, когда курсор находится в конце строки.

Действие. Ввести правильную командную строку.

Запрещенный символ

Причина. Использован недопустимый управляющий символ.

Действие. Ввести допустимый управляющий символ.

Запрещено

Причина. Использована недопустимая клавиша.

Действие. Нажать допустимую клавишу.

Строки не было

Причина. Попытка использовать клавишу «стрелка вверх», когда введенной строки не существует.

Действие. Использовать допустимую функцию или команду.

Граница строки

Причина. Попытка ввести командную строку, длина которой превышает допустимую.

Действие. Ввести командную строку допустимой длины.

Стираний не было

Причина. Попытка восстановить строку, когда строка отсутствует или не была уничтожена.

Действие. Использовать допустимую функцию или команду.

Команда запрещена

Причина. Введенная команда не поддерживается редактором командной строки.

Действие. Ввести командную строку.

## ПЕРЕЧЕНЬ ССЫЛОЧНЫХ ДОКУМЕНТОВ

1. Операционная система ФОДОС-2  
Командный язык системы
2. Операционная система ФОДОС-2  
Программы работы с файлами.  
Руководство оператора
3. Операционная система ФОДОС-2  
Установка и генерация системы  
Руководство системного программиста

## СОДЕРЖАНИЕ

### СИСТЕМНАЯ МАКРОБИБЛИОТЕКА

	Стр.
1. Назначение и условия применения . . . . .	3
2. Обращение к системной макробиблитеке . . . . .	4
2.1. Адресация аргументов . . . . .	7
2.2. Ключевой аргумент CODE . . . . .	8
3. Входные и выходные данные . . . . .	9
3.1. Номер канала . . . . .	9
3.2. Блок устройства (DBLK) . . . . .	10
3.3. Блок аргументов EMT . . . . .	10
3.4. Подпрограммы завершения . . . . .	10
3.5. Требование USR . . . . .	11
4. Сообщения . . . . .	12
5. Программные запросы . . . . .	12
5.1. Программные запросы для операций с файлами . . . . .	18
5.1.1. .CHCOPY (режим FB, XM) . . . . .	18
5.1.2. .CLOSE . . . . .	20
5.1.3. .DELETE . . . . .	21
5.1.4. .ENTER . . . . .	21
5.1.5. .LOOKUP . . . . .	24
5.1.6. .PURGE . . . . .	26
5.1.7. .RENAME . . . . .	27
5.1.8. .REOPEN . . . . .	28
5.1.9. .SAVESTATUS . . . . .	29
5.2. Программные запросы для организации ввода-вывода данных. . . . .	30
5.2.1. .READ, .READC, .READW . . . . .	30
5.2.2. .WRITE, .WRITC, .WRITW . . . . .	35
5.2.3. .RCVD, .RCVDC, .RCVDW (режим FB, XM) . . . . .	37
5.2.4. .SDAT, .SDATC, .SDATW (режим FB, XM) . . . . .	38
5.2.5. .TTYIN, .TTINR . . . . .	39
5.2.6. .TTYOUT, .TTOUTR . . . . .	41
5.2.7. .PRINT . . . . .	42
5.2.8. .MTIN . . . . .	43
5.2.9. .MTOUT . . . . .	44
5.2.10. .MTPRNT . . . . .	44
5.2.11. .SPFUN . . . . .	45
5.3. Запросы общего назначения . . . . .	46
5.3.1. .CDFN . . . . .	46
5.3.2. .CHAIN . . . . .	48
5.3.3. .CMKT . . . . .	49
5.3.4. .CNTXSW (режим FB, XM) . . . . .	50
5.3.5. .CSIGEN . . . . .	51
5.3.6. .CSISPC . . . . .	54
5.3.6.1. Передача информации о переключателях программе пользо- вателя . . . . .	56
5.3.7. .ABTIO . . . . .	57
5.3.8. .CSTAT (режим FB, XM) . . . . .	57
5.3.9. .DATE . . . . .	58
5.3.10. .DEVICE (режим FB, XM) . . . . .	59
5.3.11. .DSTATUS . . . . .	60
5.3.12. .EXIT . . . . .	61
5.3.13. .FETCH, .RELEASES . . . . .	63

5.3.14.	.FORK	64
5.3.15.	.FPROT	65
5.3.16.	.GTIM	66
5.3.17.	.GTJB	67
5.3.18.	.GTLIN	68
5.3.19.	.GVAL, .PVAL	69
5.3.20.	.HERR, .SERR	70
5.3.21.	.HRESET	72
5.3.22.	.INTEN	72
5.3.23.	.LOCK, .UNLOCK	72
5.3.24.	.MFPS, .MTPS	74
5.3.25.	.MRKT	75
5.3.26.	.MTATCH	77
5.3.27.	.MTDTCB	77
5.3.28.	.MTSET	77
5.3.29.	.MTGET	79
5.3.30.	.MTSTAT	80
5.3.31.	.MTRCTD	80
5.3.32.	.MWAIT (режим FB, XM)	80
5.3.33.	.PEEK, .POKE	82
5.3.34.	.PROTECT, .UNPROTECT (режим FB, XM)	82
5.3.35.	.QSET	83
5.3.36.	.RCTRLD	84
5.3.37.	.SPND, .RSUM (режим FB, XM)	85
5.3.38.	.SCCA	86
5.3.39.	.SDTTM	87
5.3.40.	.SETTOP	88
5.3.41.	.SPCPS (режим FB, XM)	88
5.3.42.	.SFDAI	90
5.3.43.	.SFPA	90
5.3.44.	.SRESET	91
5.3.45.	.SYNCH	91
5.3.46.	.TLOCK	92
5.3.47.	.TRPSET	92
5.3.48.	.TWAIT	93
5.3.49.	.WAIT	94
ПРИЛОЖЕНИЕ 1. Коды ошибок программных запросов		95
ПРИЛОЖЕНИЕ 2. Структура блока CBLK		99
Перечень ссылочных документов		99

## БИБЛИОТЕКА СИСТЕМНЫХ ПОДПРОГРАММ

1.	Назначение и условия применения библиотеки системных подпрограмм	100
1.1.	Назначение	100
1.2.	Условия применения	100
1.2.1.	Требования к техническим средствам	100
1.2.2.	Требования к программным средствам	101
1.2.3.	Порядок использования библиотеки системных подпрограмм	101
1.2.4.	Требования к оперативной памяти для ФОРТРАН-программ в основном режиме	102
1.2.5.	Особенности использования USSR	103
1.2.6.	Требования к подпрограммам завершения	103
2.	Обращение к подпрограммам. Входные и выходные данные	104

2.1.	Обращение на ФОРТРАНе . . . . .	105
2.2.	Обращение на АССЕМБЛЕРе . . . . .	105
2.3.	Входные и выходные данные . . . . .	107
3.	Характеристики подпрограмм . . . . .	107
3.1.	Операции с файлами . . . . .	108
3.1.1.	CLOSEC/ICLOSE . . . . .	108
3.1.2.	IDELET . . . . .	109
3.1.3.	IENTER . . . . .	110
3.1.4.	IFPROT . . . . .	111
3.1.5.	IRENAM . . . . .	112
3.1.6.	ISFDAT . . . . .	112
3.1.7.	LOOKUP . . . . .	113
3.2.	Передача данных . . . . .	114
3.2.1.	IABTIO . . . . .	114
3.2.2.	GTLIN . . . . .	114
3.2.3.	IRCV D/IRCVDC/IRCVDF/IRCVDW (режим FB) . . . . .	115
3.2.3.1.	IRCV D . . . . .	115
3.2.3.2.	IRCVDC . . . . .	116
3.2.3.3.	IRCVDF . . . . .	116
3.2.3.4.	IRCVDW . . . . .	116
3.2.4.	IREAD/IREADC/IREADF/IREADW . . . . .	117
3.2.4.1.	IREAD . . . . .	117
3.2.4.2.	IREADC . . . . .	117
3.2.4.3.	IREADF . . . . .	118
3.2.4.4.	IREADW . . . . .	118
3.2.5.	ISDAT/ISDATC/ISDATF/ISDATW (режим FB) . . . . .	119
3.2.5.1.	ISDAT . . . . .	119
3.2.5.2.	ISDATC . . . . .	119
3.2.5.3.	ISDATF . . . . .	120
3.2.5.4.	ISDATW . . . . .	120
3.2.6.	ITTINR . . . . .	120
3.2.7.	ITTOUR . . . . .	122
3.2.8.	IWAIT . . . . .	122
3.2.9.	IWRITE/IWRITC/IWRITF/IWRITW . . . . .	122
3.2.9.1.	IWRITE . . . . .	122
3.2.9.2.	IWRITC . . . . .	123
3.2.9.3.	IWRITF . . . . .	123
3.2.9.4.	IWRITW . . . . .	123
3.2.10.	MWAIT (режим FB) . . . . .	124
3.2.11.	PRINT . . . . .	124
3.3.	Операции с каналами . . . . .	125
3.3.1.	ICDFN . . . . .	125
3.3.2.	ICHCPY (режим FB) . . . . .	126
3.3.3.	ICSTAT (режим FB) . . . . .	126
3.3.4.	IFREEC . . . . .	127
3.3.5.	IGETC . . . . .	127
3.3.6.	ILUN . . . . .	128
3.3.7.	IREOPN . . . . .	128
3.3.8.	ISAVES . . . . .	129
3.3.9.	PURGE . . . . .	130
3.4.	Описание устройств и файлов . . . . .	130
3.4.1.	IASIGN . . . . .	130
3.4.2.	ICSI . . . . .	132

<b>3.5. Работа с таймером . . . . .</b>	<b>134</b>
<b>3.5.1. CVTTIM . . . . .</b>	<b>134</b>
<b>3.5.2. GTIM . . . . .</b>	<b>135</b>
<b>3.5.3. ICMKT . . . . .</b>	<b>135</b>
<b>3.5.4. ISCHED (режим FB) . . . . .</b>	<b>136</b>
<b>3.5.5. ISDTTM . . . . .</b>	<b>137</b>
<b>3.5.6. ISLEEP (режим SJ) . . . . .</b>	<b>137</b>
<b>3.5.7. ITIMER . . . . .</b>	<b>138</b>
<b>3.5.8. ITWAIT (вариант SJ) . . . . .</b>	<b>139</b>
<b>3.5.9. IUNTIL (вариант SJ) . . . . .</b>	<b>140</b>
<b>3.5.10. JTIME . . . . .</b>	<b>140</b>
<b>3.5.11. MRKT (вариант SJ) . . . . .</b>	<b>140</b>
<b>3.5.12. SECNDS . . . . .</b>	<b>141</b>
<b>3.5.13. TIMASC . . . . .</b>	<b>141</b>
<b>3.5.14. TIME . . . . .</b>	<b>142</b>
<b>3.6. Системные операции . . . . .</b>	<b>142</b>
<b>3.6.1. CHAIN . . . . .</b>	<b>142</b>
<b>3.6.2. DEVICE (режим FB) . . . . .</b>	<b>143</b>
<b>3.6.3. GTJB/IGTJB . . . . .</b>	<b>144</b>
<b>3.6.4. IDSTAT . . . . .</b>	<b>146</b>
<b>3.6.5. IFETCH . . . . .</b>	<b>146</b>
<b>3.6.6. IQSET . . . . .</b>	<b>147</b>
<b>3.6.7. ISPFN/ISPFNC/ISPFNF/ISPFNW . . . . .</b>	<b>148</b>
<b>3.6.7.1. ISPFN . . . . .</b>	<b>148</b>
<b>3.6.7.2. ISPFNC . . . . .</b>	<b>150</b>
<b>3.6.7.3. ISPFNF . . . . .</b>	<b>150</b>
<b>3.6.7.4. ISPFNW . . . . .</b>	<b>151</b>
<b>3.6.8. ITLOCK (режим FB/XM) . . . . .</b>	<b>151</b>
<b>3.6.9. LOCK . . . . .</b>	<b>152</b>
<b>3.6.10. RCHAIN . . . . .</b>	<b>153</b>
<b>3.6.11. RCTRLO . . . . .</b>	<b>153</b>
<b>3.6.12. RESUME (режим FB) . . . . .</b>	<b>154</b>
<b>3.6.13. SCCA . . . . .</b>	<b>154</b>
<b>3.6.14. SETCMD . . . . .</b>	<b>155</b>
<b>3.6.15. SUSPND (режим FB) . . . . .</b>	<b>155</b>
<b>3.6.16. UNLOCK . . . . .</b>	<b>156</b>
<b>3.7. Операции с данными INTEGER*4 . . . . .</b>	<b>157</b>
<b>3.7.1. AJELT . . . . .</b>	<b>158</b>
<b>3.7.2. DJFLT . . . . .</b>	<b>158</b>
<b>3.7.3. IAJFLT . . . . .</b>	<b>158</b>
<b>3.7.4. IDJFLT . . . . .</b>	<b>159</b>
<b>3.7.5. JCVT . . . . .</b>	<b>159</b>
<b>3.7.6. JADD . . . . .</b>	<b>160</b>
<b>3.7.7. JAFIX . . . . .</b>	<b>160</b>
<b>3.7.8. JCMP . . . . .</b>	<b>161</b>
<b>3.7.9. JDFIX . . . . .</b>	<b>161</b>
<b>3.7.10. JDIV . . . . .</b>	<b>162</b>
<b>3.7.11. JICVT . . . . .</b>	<b>162</b>
<b>3.7.12. JJCVT . . . . .</b>	<b>163</b>
<b>3.7.13. JMOV . . . . .</b>	<b>163</b>
<b>3.7.14. JMUL . . . . .</b>	<b>164</b>
<b>3.7.15. JSUB . . . . .</b>	<b>164</b>
<b>3.8. Операции с текстовыми данными . . . . .</b>	<b>165</b>
<b>3.8.1. Размещение текстового данного в массиве . . . . .</b>	<b>165</b>

3.8.2.	Передача текстовых данных	166
3.8.3.	CONCAT	167
3.8.4.	GETSTR	168
3.8.5.	INDEX	168
3.8.6.	INSERT	169
3.8.7.	ISCOMP/SCOMP	170
3.8.8.	VERIFY/IVERIF	170
3.8.9.	LEN	171
3.8.10.	PUTSTR	171
3.8.11.	REPEAT	172
3.8.12.	SCOPY	172
3.8.13.	STRPAD	173
3.8.14.	SUBSTR	174
3.8.15.	TRANSL	174
3.8.16.	TRIM	175
3.9.	Преобразования RADIX-50	175
3.9.1.	IRAD50	175
3.9.2.	R50ASC	176
3.9.3.	RAD50	176
3.10.	Работа в мультитерминальном режиме	176
3.10.1.	MTATCH (режим FB/XM)	177
3.10.2.	MTDTCH (режим FB/XM)	179
3.10.3.	MTGET (режим FB/XM)	179
3.10.4.	MTIN (режим FB/XM)	179
3.10.5.	MTOUT (режим FB/XM)	180
3.10.6.	MTPRNT (режим FB/XM)	180
3.10.7.	MTRCTO (режим FB/XM)	181
3.10.8.	MTSET (режим FB/XM)	181
3.10.9.	MTSTAT (режим FB/XM)	181
3.11.	Вспомогательные операции	182
3.11.1.	IADDR	182
3.11.2.	IGETSP	182
3.11.3.	INTSET	182
3.11.4.	IPEEK	184
3.11.5.	IPEKB	184
3.11.6.	IPOKE	184
3.11.7.	IPOKEB	185
3.11.8.	ISPY	185
3.11.9.	IPUT	186
4.	Сообщения	186
ПРИЛОЖЕНИЕ 1.	Список системных подпрограмм	186
ПРИЛОЖЕНИЕ 2.	Особенности выполнения подпрограмм	191
ПРИЛОЖЕНИЕ 3.	Формат слова состояния канала	193
ПРИЛОЖЕНИЕ 4.	Формат слова асинхронного состояния терминала	193
Перечень ссылочных документов		193

## СООБЩЕНИЯ СИСТЕМЫ

1.	Сокращения и условные обозначения	195
2.	Сообщения транслятора с языка ФОРТРАН и библиотеки ФОРТРАН	198
2.1.	Сообщения транслятора с языка ФОРТРАН	198
2.1.1.	Сообщения предварительного контроля	198

2.1.2. Сообщения синтаксического и семантического контроля . . . . .	199
2.1.2.1. Сообщения, предупреждающие об ошибках . . . . .	209
2.1.2.2. Сообщения о неустранимых ошибках . . . . .	209
2.2. Сообщения библиотеки ФОРТРАН . . . . .	211
3. Сообщения интерпретатора с языка БЕЙСИК . . . . .	222
4. Сообщения операционной системы ФОДОС-2 . . . . .	237
5. Сообщения программы АССЕМБЛЕР . . . . .	326
6. Сообщения редакторов . . . . .	331
6.1. Сообщения редактора текста . . . . .	331
6.2. Сообщения экранного редактора . . . . .	334
6.3. Сообщения редактора командной строки . . . . .	346
Перечень ссылочных документов . . . . .	346

Ответственный за выпуск **М. Г. Бойкова**

Редактор **Т. А. Савельева**

Корректор **В. Н. Лыткина**

Сдано в набор 19.03.90 г. Подписано в печать 19.07.90 г.

Формат 60×84<sup>1</sup>/<sub>16</sub>. Бумага писчая № 1. Гарнитура литературная.

Печать высокая. Усл. печ. л. 20,54. Тираж 20 000 экз. Зак. 801.

Бесплатно.

Ленинградское отделение РГПО «Союзбланкоиздат».

Великолукская городская типография управления издательств,  
полиграфии и книжной торговли Псковского облисполкома,  
182100, г. Великие Луки, ул. Полиграфистов, 78/12