

Министерство высшего и среднего специального образования
Р С Ф С Р

Куйбышевский ордена Трудового Красного Знамени авиационный
институт имени академика С.П.Королева

ИСПОЛЬЗОВАНИЕ ВАЖНЕЙШИХ КОМАНД И КЛЮЧЕЙ
СЛУЖЕБНЫХ ПРОГРАММ ОСРВ СМ ЭВМ

Утверждено редакционно-издательским
советом института в качестве
методических указаний для студентов

Куйбышев 1985

УДК 681.3.06

В методических указаниях дано краткое описание важнейших команд и ключей служебных программ *MCR*, *EDI*, *FOR*, *TKB*, *PIP*, *PLX* операционной системы реального времени СМ ЭВМ. Приводятся соответствующие примеры. Содержащихся в указаниях сведений достаточно для подготовки и отладки простых программ в среде ОСРВ.

Методические указания предназначены для студентов всех специальностей и инженеров, выполняющих работы, связанные с прикладным программированием в среде ОСРВ СМ ЭВМ

Составитель С.В.Смирнов

Рецензенты: канд. техн. наук, доц. О.К.Риттер,
канд. техн. наук С.В.Суханов

Предисловие

Ц е л ь данных методических указаний – познакомить студента с элементарными основами подготовки и отладки программ в рамках операционной системы реального времени СМ ЭВМ.

Пособие предполагает знание читателем программирования в целом; сведения же об ОС РВ вводятся за исключением специально оговариваемых случаев постепенно, по мере потребности с надлежащими примерами и, по-возможности, рафинируются от многочисленных деталей. Процедуры подготовки и отладки программ разделены на три группы:

р а б о т а з а т е р м и н а л о м , где обсуждаются возможности клавиатуры, взаимодействие с программой связи с оператором MCR (в том числе регистрация пользователя в системе), приводится список основных команд программы связи с оператором;

п о д г о т о в к а п р о г р а м м ы , где описывается использование служебных программ для создания, компиляции, компоновки и выполнения пользовательских программ;

р а б о т а с ф а й л а м и , где приводятся правила описания файлов и работы с ними, а также сведения об использовании косвенных команд файлов.

1. РАБОТА ЗА ТЕРМИНАЛОМ

Средством связи между пользователем и ОСРВ обычно служит алфавитно-цифровой дисплей, называемый здесь т е р м и н а л о м . Пользователь дает задание системе, набирая команды на клавиатуре терминала. Система либо принимает эти команды, либо выводит сообщение о том, что не понимает введенные данные или не может ответить на них.

1.1. И с п о л ь з о в а н и е т е р м и н а л а

Система ОСРВ может обслуживать видеотерминалы различных марок и производителей: ВТА 2000-3, VT 340, VT 52100-C, MEPA 7953. Все эти и им подобные терминалы имеют в целом одинаковую клавиатуру, совпадающую в основном с клавиатурой пишущей машинки, где строчные буквы заменены буквами латинского алфавита и некоторыми специальными знаками. Кроме того, клавиатура терминала имеет дополнитель-

ные ("функциональные") клавиши специального назначения, дающие возможность выполнять разнообразные операции по обработке данных на ЭВМ. Начиная работу с терминалом, следует внимательно изучить расположение и набор клавиш на клавиатуре.

Далее дается описание наиболее важных функциональных клавиш, имеющих на упомянутых выше терминалах.

ВК, или *CR* , или *RETURN*

— "возврат каретки", завершает вводимую строку и продвигает каретку (т.е. курсор) на первую позицию следующей строки.*

УС, или *CTRL*

"управляющий символ", является частью многочисленных двухклавишных комбинаций (УС и буквенная клавиша), выполняющих различные функции. Управляющим символом в данном пособии называется именно эта комбинация, обозначаемая далее УС/Х, где Х — соответствующая буква.

ЗБ, или *DEL* , или *RUBOUT*

— "забей", удаляет последний набранный в строке символ, а при повторных нажатиях — примыкающие слева символы. Обычно терминалы стирают каждый удаленный символ с экрана и перемещают курсор на одну позицию влево.

ТАБ, или *TAB*

— "табулирование", каждая восьмая позиция строки является позицией останова при табуляции. Нажатие клавиши ТАБ вызывает перемещение курсора на следующую такую позицию.

У п р а в л я ю щ и й с и м в о л вводится одновременным нажатием соответствующей буквенной клавиши и клавиши УС. Ниже приведены наиболее часто используемые управляющие символы.

УС/С

— вызывает прерывание выполняемого на терминале ввода или вывода данных и передает управление программе связи с оператором *MCR*, которая отвечает подсказкой *MCR>* и ждет ввода команды.

УС/5 и УС/О

— управляющий символ УС/5 приостанавливает вывод на терминал

* Далее завершение строки нажатием клавиши "возврат каретки" условно будем изображать как <ВК>

до момента введения управляющего символа UC/Q . Потери данных при этом не происходит.

UC/U

удаляет набранную строку и выполняет возврат каретки. Это позволяет вновь набрать строку, если отдельные исправления делать не разумно. Управляющий символ выполняет свою функцию, если строка не завершена нажатием $\langle BK \rangle$.

UC/Z

— используется для завершения работы многих служебных программ и передачи управления программе *MCR* .

1.2. Программа связи с оператором

Управление работой системы с терминала осуществляется путем выдачи команд специальной служебной программе *MCR* — посредник у между пользователем и системой:

вызов программы : UC/C , или UC/Z

ответ программы : $MCR >$ или $>$

Обе подсказки ($MCR >$ и $>$) свидетельствуют, что программа *MCR* готова к приему команд с терминала. Командная строка имеет следующий формат:

$MCR >$ ИМЯ_КОМАНДЫ ПАРАМЕТР/КЛЮЧИ $\langle BK \rangle$

Ниже кратко прокомментируем компоненты командной строки.

ИМЯ_КОМАНДЫ состоит из трех или более букв, ограниченных пробелом, и определяет подлежащую выполнению функцию программы *MCR*. Программа *MCR* читает только три первые буквы, остальные буквы лишь помогают пользователю идентифицировать команду (в данном пособии эти необязательные части команд не приводятся). Приведем несколько примеров имен команд:

MOU — монтировать файловый том;

RUN — пустить задачу;

ABO — снять выполняемую в данный момент задачу, т.е. прекратить выполнение программы.

ПАРАМЕТР обычно описывает объект, над которым производятся действия, определяемые данной командой. Чаще всего объект — это устройство или задача (т.е. готовая к выполнению программа). Параметр от параметра и от имени команды должны отделять один или более пробелов.

> *MODU DK5* : < *BK* >

- монтировать файловый том на диске с подвижными головками №5;

> *RUN Zθ* < *BK* >

- пустить задачу с именем *Zθ*.

Ключи модифицируют либо параметр, либо функцию команды. Каждый ключ начинается знаком /, за которым следует алфавитно-цифровая последовательность, идентифицирующая данный ключ. Затем может следовать знак равенства и значение ключа (/КЛЮЧ=ЗНАЧЕНИЕ). Ключ следует непосредственно за параметром, который он определяет (ПАРАМЕТР/КЛЮЧ). Однако, если ключ модифицирует функцию команды, надо отделить имя команды от ключа хотя бы одним пробелом (ИМЯ КОМАНДЫ /КЛЮЧ). Пробелы между последовательно идущими ключами не требуются.

> *RUN Z1* / *RSI=15M* < *BK* >

- запуск задачи *Z1* и повторение запуска через 15 минут;

> *SET* / *UIC* [... , ...] < *BK* >

- устанавливает новый код идентификации пользователя (здесь квадратные скобки являются частью кода идентификации, а вместо многоточий указываются цифровые коды, например, [200, 253]). Каждая строка завершается нажатием клавиши < *BK* > .

1.3. Р е г и с т р а ц и я в с и с т е м е

Для работы в среде ОСРВ пользователю необходимо вначале зарегистрировать вход в систему с терминала, а по окончании работы - выход из системы.

Далее предполагается, что пользователь имеет доступ к системе (т.е. имеет право работать с системой), терминал включен и находится на связи с системой.

Для регистрации входа в систему пользователь должен набрать команду

> *HEL* ИМЯ-ПОЛЬЗОВАТЕЛЯ

или

> *HEL* КОД-ИДЕНТИФИКАЦИИ-ПОЛЬЗОВАТЕЛЯ

ИМЯ-ПОЛЬЗОВАТЕЛЯ - параметр, устанавливаемый индивидуально для каждого имеющего доступ к системе пользователя. КОД-ИДЕНТИФИКАЦИИ-ПОЛЬЗОВАТЕЛЯ - числовой эквивалент имени пользователя (или, сокращенно, *UIC* пользователя), который состоит из двух восьмеричных чисел, разделенных косой чертой. Например,

>HEL IVANOV <BK>

>HEL 2/25 <BK>

Отвечая на команду *HEL* программа *MCR* запрашивает пароль пользователя, выдавая на экран во-прос

PASSWORD :

В ответ пользователь должен набрать индивидуально установленный для него пароль и <BK> . П а р о л ь - некоторая алфавитно-цифровая последовательность не длиннее шести символов, которая совместно с именем пользователя или его *UIC* служит для опознавания пользователя системой и допуска его к работе.

Внимание! Система не печатает символы, набираемые в ответ на *PASSWORD* :.

Регистрация выхода из системы осуществляется командой

>BYE <BK>

1.4. С о о б щ е н и я о б о ш и б к а х

Когда программа *MCR* получает от пользователя недопустимые или не поддающиеся распознаванию данные, она выдает на терминал сообщение об ошибке. Сообщение начинается с первых букв имени связанной с ним команды или задачи, а затем следует краткое пояснение на английском языке. Все виды сообщений и более подробная их расшифровка приводятся в штатной документации системы. Например, при выдаче команды *HEL* система может вывести сообщение:

HEL- OTHER USER LOGGED ON

Это означает, что данный терминал занят; следует попытаться зарегистрироваться с другого терминала, либо попросить освободить данный терминал.

1.5. О с н о в н ы е к о м а н д ы п р о г р а м м ы с в я з и с о п е р а т о р о м *

Опишем наиболее употребляемые командные строки программы *MCR*.
ABO ИМЯ_ЗАДАЧИ

- прекратить выполнение задачи;

ASN

- установить, стереть, выдать логическую идентификацию устройств;

* При первом прочтении этот раздел рекомендуется пропустить.

CAN ИМЯ_ЗАДАЧИ

- аннулировать запланированный (в будущем) запуск задачи;

INS СПЕЦИФИКАЦИЯ _ ФАЙЛА [КЛЮЧ]

- установить задачу (занести информацию о задаче в таблицу монитора системы), образ которой содержит указанный файл (типа *.TSK*).

Приведем варианты ключей команды *INS* :

/PAR = ИМЯ _ РАЗДЕЛА

/PRI = ЗНАЧЕНИЕ _ ПРИОРИТЕТА (от 1 до 250)

/TASK ИМЯ_ЗАДАЧИ

/UIC [...,...]

REM ИМЯ_ЗАДАЧИ

- отменить установку задачи;

FIX ИМЯ_ЗАДАЧИ

- фиксировать задачу в основной памяти (после завершения задачи остается в основной памяти; задача должна быть установлена);

LUN ИМЯ_ЗАДАЧИ

- выдать назначение логических устройств задачи (задача должна быть установлена);

REA ИМЯ_ЗАДАЧИ *N* ИМЯ_УСТРОЙСТВА :

- назначить в задаче логический номер *N* указанному устройству (задача должна быть установлена);

RES ИМЯ_ЗАДАЧИ

- продолжить выполнение ранее приостановленной задачи (задача должна быть установлена).

Далее приведены пять вариантов команды *RUN* .

I. Запустить задачу:

RUN ИМЯ_ЗАДАЧИ [КЛЮЧ] [КЛЮЧ]

Ключи:

IRSI = *NNNVU* - интервал повторения запуска ,
где *NNNV* - число

U - обозначение единицы времени , а именно:

T - такт (до 32767);

S - секунда (до 32767);

M - минута (до 1440);

H - час (до 24)

/UIC = [...,...] - код идентификации пользователя.

II. Запустить задачу по истечении заданного временного промежутка *ИНТЕРВАЛ*:

RUN ИМЯ — ЗАДАЧИ ИНТЕРВАЛ [/КЛЮЧ] [/КЛЮЧ]

Параметр ИНТЕРВАЛ определяется в форме *NNNVU* (см. вариант I).
Ключи: см. вариант I.

III. Запустить задачу по истечении единицы времени ЕДВР:
RUN ИМЯ — ЗАДАЧИ ЕДВР [ИНТЕРВАЛ] [/КЛЮЧ] [/КЛЮЧ]
Параметр ЕДВР может принимать значения *H* — час, *M* — минута,
S — секунда, *T* — такт. ИНТЕРВАЛ и ключи имеют тот же смысл и
задание как в вариантах I и 2.

IV. Запустить задачу в заданное абсолютное время АБСВР:
RUN ИМЯ — ЗАДАЧИ АБСВР [/КЛЮЧ] [/ КЛЮЧ]
Параметр АБСВР задается в форме ЧАСЫ : МИНУТЫ : СЕКУНДЫ.
Ключи: см. вариант I.

V. Установить, запустить и после завершения задачи выдать
сообщение:

RUN СПЕЦИФИКАЦИЯ — ФАЙЛА [/КЛЮЧ] [/КЛЮЧ]
Параметр и ключи те же, что для команды *INS*. Сообщение выдается
в форме

ИМЯ — ЗАДАЧИ : -- STOP

TIM

- выдать либо установить время и дату в форме ЧАСЫ : МИНУТЫ
[:СЕКУНДЫ] МЕСЯЦ /ДЕНЬ /ГОД .

UFD ИМЯ — УСТРОЙСТВА : [.....]

- выдать каталог файлов пользователя, имеющего код иденти-
фикации [.....] ; подлежащие перечислению файлы располагаются
на носителе указанного устройства.

UNF ИМЯ — ЗАДАЧИ

- отменить фиксирование задачи в основной памяти.

2. ПОДГОТОВКА ПРОГРАММЫ

Подготовка программы для выполнения под управлением ОСРВ
предполагает осуществление следующих операций:

создание и с х о д н о й программы;

трансляция исходной программы для получения объектного кода
(о б ъ е к т н о й программы).

компоновка в общем случае нескольких объектных программ для получения готовой к выполнению программы, называемой *з а д а ч е й*. Задача может быть запущена по команде программы *MCR* (см. п. I.5).

Каждый из перечисленных шагов включает в себя работу с файлом соответственно исходной программы, объектной программы и образца задачи. В разделе описываются основные возможности служебных программ ОСРВ, поддерживающих выполнение каждого из шагов подготовки программы.

Всюду в разделе предполагаем, что пользователь работает (по умолчанию) с системным диском.

2.1. Т е к с т о в ы й р е д а к т о р

Текстовый редактор *EDI* служит для создания и редактирования файла исходной программы.

Вызов программы : `>RUN *EDI<BK>` или `>EDI<BK>`

Ответ программы : `EDI>`

Командная строка : `EDI> СПЕЦИФИКАЦИЯ _ ФАЙЛА < BK >`

Компонента СПЕЦИФИКАЦИЯ _ ФАЙЛА в целом будет рассмотрена в п.

3.1. Используя принцип умолчания, указываем лишь две составляющие полной спецификации имя файла и его тип в виде

`ИМЯ _ ФАЙЛА . ТИП`

где `ИМЯ _ ФАЙЛА` - алфавитно-цифровая строка длиной от 1 до 9 символов, `ТИП` - трехбуквенный «немокод» с предшествующей ему точкой, говорящий о содержимом файла. Стандартными для файлов исходных программ на языках Бейсик, Кобол, Макроассемблер и Фортран являются соответственно типы `.BAS`, `.CBL`, `.MAC` и `.FTN`. Мультированы следующим образом.

1. Если в командной строке редактор *EDI* получает имя несуществующего файла, то он создает пустой файл с указанными в спецификации атрибутами и переходит в режим ввода, выдавая на терминал сообщение

`[CREATING NEW FILE]`

`INPUT`

После этого пользователь может сразу же начать набирать исходную программу. После завершения ввода каждой строки (обычно с помощью `<BK>`) редактор помещает ее в рабочий буфер, содержимое которого в свою очередь периодически переносится в новый файл. До заверше-

ния ввода строки ее можно исправить, используя клавиатуру (см. п. I.1); после ввода - только с помощью команд редактора в режиме редактирования.

Переход из режима ввода в режим редактирования производится набором <BK> как первого символа строки, на что редактор отвечает выводом подсказки: символа * .

2. Для внесения изменений существующий файл специфицируется в командной строке *EDI* . Редактор отыскивает этот файл и переходит в режим редактирования. Например,

```
>EDI Z1.FTN <BK>
```

```
[0029 LINES READ IN]
```

```
[PAGE 1]
```

*

Здесь первая строка - командная, вторая и третья - сообщение редактора, звездочка - подсказка, после которой набирается команда редактора.

3. В режиме редактирования из входного (специфицированного в командной строке *EDI*) файла в рабочий буфер загружается до 80 строк. С помощью команд редактирования можно (и только построчно) оперировать с содержимым буфера, либо со всей оставшейся частью файла, начиная с текущей строки.

4. Если необходимо продолжить ввод или коррекцию данных по достижению конца буфера (редактор сообщит об этом строкой [*EOB*]), следует выдать команду редактора *REN* : "переписать содержимое рабочего буфера в выходной файл и загрузить рабочий буфер следующим блоком текста (до 80 строк) из входного файла". Выходным файлом является либо вновь создаваемый с помощью редактора файл, либо новая версия (см.п.3.1) корректируемого файла. При достижении в ходе редактирования конца входного файла редактор выдает сообщение [*EOF*] .

5. По команде редактора *EX* ввод или коррекция текста будет прекращена, содержимое рабочего буфера и оставшиеся строки входного файла переписаны в выходной файл, а управление передано программе *MCR* .

Опишем наиболее употребительные команды текстового редактора *EDI* .

Позиционирование и выдача на экран

TOF

- возврат к началу входного файла (все сделанные исправления в файле сохраняются и создается новая версия выходного файла).

REN

- вывод текущего содержимого буфера в выходной файл и загрузка буфера следующим блоком текста из входного файла.

B

- позиционирование на первую строку в буфере.

BO

- позиционирование на последнюю строку в буфере.

Nn

- позиционирование на n строк вперед ($n > 0$) или назад ($-n > 0$).

P

- выдача на экран текущей строки.

Pn

- выдача на экран n строк, начиная с текущей, и позиционирование на последнюю выданную строку (эта строка становится новой текущей строкой).

NPn

- позиционирование на n строк вперед ($n > 0$) или назад ($-n > 0$) и выдача последней строки на экран; если n отсутствует, то производится позиционирование на следующую строку и вывод ее на экран.

LI

- выдача на экран всех строк буфера, начиная с текущей.

LP

- выдача всех строк буфера, начиная с текущей, на псевдоустройство *CL* :

Коррекция и поиск

A \sqcup текст

- "текст" присоединяется к концу текущей строки.

AP \sqcup текст

- см. команду *A* ; дополнительно текущая строка выдается в скорректированном виде.

C /текст 1/текст 2/

- замена в текущей строке последовательности знаков "текст 1" на "текст 2".

*C //*текст 2/

- введение последовательности знаков "текст 2" в начало текущей строки.

*C /*текст 1//

- стирание в текущей строке последовательности знаков "текст 1".

*nLC/*текст 1/текст 2/

- в *n* строках, начиная с текущей, ищется последовательность знаков "текст 1", которая заменяется на "текст 2"; каждая скорректированная строка выводится на экран.

*PA /*текст 1/текст 2/

- см. команду *LC*, но замена производится во всех строках буфера, начиная с текущей.

nL текст

- в буфере, начиная с текущей строки, ищется *n*-я по счету последовательность знаков "текст".

nPL текст

- см. команду *L*, но поиск потенциально продолжается вплоть до конца входного файла.

nFL текст

- в буфере, начиная с текущей строки, ищется и затем выводится *n*-я по счету строка, начинающаяся (с 1-й позиции) с последовательности знаков "текст".

nPF текст

- см. команду *F*, но поиск потенциально продолжается вплоть до конца входного файла.

I текст

- вставка строки "текст" после текущей строки.

I < BK >

- переключение редактора в режим ввода; возврат в режим редактирования осуществляется после двукратного нажатия клавиши *< BK >*.

SA_n СПЕЦИФИКАЦИЯ _ ФАЙЛА

- *n* строк, начиная с текущей, пересылаются в специфицированный

файл, а если СПЕЦИФИКАЦИЯ _ ФАЙЛА отсутствует - в некоторую автоматически предоставляемую область памяти.

UNS СПЕЦИФИКАЦИЯ _ ФАЙЛА

- вслед за текущей строкой помещаются все строки специфицированного файла, а если СПЕЦИФИКАЦИЯ _ ФАЙЛА отсутствует - вслед за текущей строкой помещаются строки временно выведенные по команде SA_n .

FI СПЕЦИФИКАЦИЯ _ ФАЙЛА

- содержимое входного файла, начиная с текущей строки и до его конца, пересылается в специфицированный файл.

Стирание

Dn

- стирание текущей и $(n - 1)$ следующих строк при $n > 0$; стирание n строк перед текущей строкой при $-n > 0$ (по умолчанию $n = 1$).

DP_n

- эм. команду *D*; дополнительно на экран выводится новая текущая строка.

ERASE

- стирание всего содержимого буфера.

On

- стирание текущей и $(n - 1)$ следующих строк, а затем переключение редактора в режим ввода; возврат в режим редактирования осуществляется после двукратного нажатия клавиши $\langle BK \rangle$.

Завершение работы редактора

UC/Z

- передача содержимого буфера и оставшихся необработанными строк входного файла в выходной файл, закрытие обоих файлов и передача управления программе связи с оператором *MCR*.

EX

- эквивалентна *UC/Z*.

ED

- эквивалентна *UC/Z*, но входной файл стирается.

CL

- эквивалентна *UC/Z*, но редактор сохраняет управление (отвечая *EDIG >*) для создания нового или коррекции следующего файла.

CLD

- см. команду *CL* , но входной файл стирается.

Специальная команда управления

UC OFF

- выключение автоматического преобразования знаков верхнего регистра клавиатуры в соответствующие знаки нижнего регистра (т.е. разрешение пользоваться буквами русского алфавита).

В заключение отметим, что редактор *EDI* принимает последовательность команд, связанных знаком *&* . Например,

**LC AB & LC C/A/DE < BK >*

- в буфере, начиная с текущей строки, ищут последовательность знаков *AB*, в которой *A* заменяют на *DE* .

2.2. Компиляция программ .

Компилятор языка Фортран

Компиляция - это процесс перевода исходной программы, написанной в символьном коде, в программу на машинном языке, называемую объектной программой. При вызове компиляторов в ОСРВ пользователь формирует командную строку, где файл исходной программы указывается как входной, а в качестве выходного может указываться либо файл объектной программы (тип. *OBJ*), либо файл листинга (тип. *LST*), либо оба этих файла. Командная строка может снабжаться ключами, набор и действие которых, как правило, индивидуальны для каждого компилятора. Рассмотрим в качестве примера компилятор с языка Фортран.

Вызов компилятора: *> RUN x FOR < BK > или > FOR < BK >*

Ответ компилятора: *FOR >*

Командная строка:

FOR > ОБЪКТНЫЙ-ФАЙЛ, ФАЙЛ-ЛИСТИНГА=ИСХОДНЫЙ-ФАЙЛ [/ключи] < BK >

В спецификациях указанных файлов может быть опущено указание типа, так как компилятор *FOR* по умолчанию в качестве типа файла исходной программы берет *.FTN* , файла листинга - *.LST* , файла объектной программы - *.OBJ* . Если листинг не требуется, то спецификацию файла листинга следует опустить. В случаях, когда необходим лишь листинг программы, опускается спецификация файла объектной программы, но разделяющая запятая сохраняется.

примеры:

FOR > Z1. OBJ, Z1L. LZST=Z1. FTN < BK >

FOR > Z1 = Z1 < BK >

FOR >, TI: = Z1. FTN < BK >

Последняя командная строка используется для вывода листинга на терминал.

Выполнив командную строку, компилятор повторяет на экране свой ответ-подсказку. Если нет необходимости вводить командную строку для компиляции другой программы, то прекращение работы компилятора и удаление его из памяти производится вводом управляющего символа *УС/Э*.

Ключи командной строки *FOR* устанавливают следующие специальные запросы на компиляцию (примеры).

/DE

- транслировать строки исходного текста, отмеченные в *I*-й позиции буквой *D* (по умолчанию они расцениваются как комментарии).

/EX

- допускается длина входных строк исходной программы до 80 знаков включительно (по умолчанию длина входной строки равна 72 знакам).

/ID

- включить в листинг расширенную диагностику компилятора.

/I4

- резервировать для целых чисел по 4 байта памяти.

/VA

- разрешить применение "векторов" для оптимизации доступа к многомерным массивам.

Следующие ключи определяют вариант оформления листинга (т.е. протокола работы компилятора):

/LI:0 - только протокол ошибок,

/LI:1 - исходный текст программы и протокол ошибок,

/LI:2 - таблица символов и распределения памяти,

/LI:4 - листинг объектной программы,

/LI:7 - всеобъемлющий листинг.

По умолчанию берется комбинация действия ключей */LI:1* и */LI:2*.

Отметим действие следующих операторов Фортрана по управлению ходом программы:

STOP ['текст']

- "текст" выдается на экран терминала, а управление переходит к программе связи с оператором *MCR*;

PAUSE ["текст"]

- "текст" выдается на экран терминала, задача приостанавливается, и управление передается программе *MCR* ; выполнение программы может быть продолжено по команде *RES* программы *MCR* (см.п.1.5).

Стандартное для Фортрана закрепление логических номеров устройств приведено в табл.2.1.

Т а б л и ц а 2.1

Логический номер устройства	Имя устройства в системе	Имя и тип файла
1	<i>SY:</i>	<i>FOR001.DAT</i>
2	<i>SY:</i>	<i>FOR002.DAT</i>
3	<i>SY:</i>	<i>FOR003.DAT</i>
4	<i>SY:</i>	<i>FOR004.DAT</i>
5	<i>TI:(TT:)</i>	<i>FOR005.DAT</i>
6	<i>CL:(LP)</i>	<i>FOR006.DAT</i>

2.3. Компоновка объектных программ

Компоновку объектных программ в ОСРВ осуществляет служебная программа *TKB* - строитель задач. Связывая заданные объектные программы и разрешая ссылки к модулям из системной библиотеки, строитель задач создает файл образа задачи, содержащий готовую к выполнению программную единицу. Для запуска задачи пользователь выдает отдельную команду программы *MCR* (см.1.5).

Вызов программы: *>RUN k TKB <BK> ИЛИ >TKB <BK>*

Ответ программы: *TKB >*

Командная строка (основной формат):

TKB>TSK-ФАЙЛ,MAP-ФАЙЛ,STB-ФАЙЛ=ОБЪКТНЫЙ-ФАЙЛ1,... <BK>

TKB > ОБЪКТНЫЙ-ФАЙЛ-k,... <BK>

TKB >|| <BK>

Командная строка строителя задач может занимать несколько строк терминала. Пока пользователь будет заканчивать строку набором *<BK>* , строитель задач продолжает выдавать подсказку *TKB >* для продолжения командной строки. Окончание командной строки пользователь фиксирует вводом символа *//* на очередную подсказку *TKB*.

В спецификациях файлов командной строки ТКВ может быть опущено указание типа, так как построитель задач по умолчанию в качестве типа файла образа задачи (*ТСК-ФАЙЛ*) берет .*ТСК* , файла распределения памяти (*МАР-ФАЙЛ*)-*МАР* , файла определения символов (*СТВ-ФАЙЛ*)-*СТВ* , а все файлы объектных программ по умолчанию имеют тип .*OBJ* .

Если какой-либо из файлов, создаваемых построителем, не требуется, то его спецификацию в командной строке следует опустить. При этом разделяющие запятые, предшествующие сохраняемым спецификациям, должны быть оставлены. Например, команда

```
ТКВ > Z = ZA < BK >
```

освобождает построитель задач от создания файлов распределения памяти и определения символов, а команда

```
ТКВ > Z, , ZS = ZA < BK >
```

разрешает не создавать файл распределения памяти.

Файл распределения памяти содержит символьную информацию о размерах и расположении отдельных частей внутри задачи. Для получения листинга файла распределения памяти, например на устройстве широкой печати, достаточно вместо его спецификации указать имя этого устройства. Например,

```
ТКВ > Z1, LP: = Z1, Z2 < BK >
```

Заметим, что в примере файл определения символов не будет создан. Он представляется в двоичном, а не в символьном виде, и поэтому содержащаяся в нем информация бесполезна для большинства пользователей.

Построитель задач имеет множество ключей и параметров, дающих дополнительные возможности управлять созданием образа задачи; подобные сведения о них имеются в штатной документации ОСРВ.

Если по какой-либо причине создание образа задачи не может быть завершено, то построитель задач выводит на экран терминала соответствующее сообщение на английском языке. Полный перечень сообщений и их расшифровка приводятся в штатной документации системы.

3. РАБОТА С ФАЙЛАМИ

3.1. Полная спецификация файла

Стандартные средства для работы с файлами являются неотъемлемой частью ОСРВ. Формат полной спецификации файла имеет вид

ИМЯ - УСТРОЙСТВА: [ГР, ЧЛ] ИМЯ-ФАЙЛА.ТИП; ВЕРСИЯ

Рассмотрим компоненты этой спецификации.

1. ИМЯ - УСТРОЙСТВА определяет устройство, на котором располагается файл. Это имя состоит из двух алфавитных символов, номера устройства (одна или две восьмеричные цифры) и заканчивается двоеточием. Номер в имени устройства может отсутствовать, в этом случае система полагает его равным нулю. Например, *DK1* : - диск с подвижной головкой № 1; *MT* : - магнитная лента № 0.

Имя устройства может определять реальное физическое устройство или быть логическим именем устройства, соответствующего различным физическим устройствам (соответствие устанавливается пользователем). В табл.3.1 перечислены имена некоторых используемых устройств ОСРВ; буква *л* означает номер устройства.

Т а б л и ц а 3.1

Устройство	Имя устройства
Диск с подвижной головкой	<i>DKn</i> :
Диск гибкий	<i>DXn</i> :
Магнитная лента	<i>MTn</i> :
Устройство широкой печати	<i>LPn</i> :
Терминал	<i>TTn</i> :
Логическое устройство "вводной терминал"	<i>TI</i> :
Логическое устройство "системный диск"	<i>SY</i> :

Так как имена *TI* : , *TT* : и *LP* : идентифицируют вводные или выводные устройства без памяти, они обычно не появляются в полной спецификации файла: при ссылке на такое устройство в спецификации файла достаточно указывать только его имя. Например, следующая команда посылает копию файла *ABC.FTN* с гибкого диска № 1 на терминал: *PIP>TI:=DX1:ABC.FTN<BK>*

2. [ГР, ЧЛ] - код идентификации пользователя (*UID*) является числовым эквивалентом имени пользователя и состоит из двух восьмеричных чисел, взятых в квадратные скобки и разделенных запятой.

Регистрируясь в системе (см. п.1.3), пользователь прямо или косвенно (через ИМЯ __ ПОЛЬЗОВАТЕЛЯ) описывает идентифицирующий его

код. Этот код однозначно определяет каталог файлов пользователя. Каталог сам является файлом (*UFD*), хранящимся на выделенном пользователю устройстве прямого доступа, и содержит информацию о каждом файле пользователя на этом устройстве (имя, тип файла, местонахождение и т.п.). Файл *UFD* является единственным каталогом, к которому пользователь имеет доступ по умолчанию.

Система накладывает ограничения на доступ к каталогам и файлам других пользователей. Каждый файл включает в себя код защиты, определяющий, кто и какой доступ имеет к файлу; при попытке получить доступ к файлу система проверяет код идентификации пользователя и код защиты. Если они не обеспечивают прав доступа к файлу, система выдает на терминал сообщение об ошибке. Например, при попытке отредактировать файл, доступ к которому по данному *UIC* запрещен, программа *EDI* выдает следующее сообщение:
EDI -- PRIVILEGE VIOLATION

3. Компонента спецификации ИМЯ _ ФАЙЛА.ТИП рассматривалась в п.2.1.

4. Компонента ; ВЕРСИЯ представляет собой номер версии файла. Номер версии является восьмеричным числом и принимает значение от 1 до 77777. После создания файла система полагает номер версии равным 1, а затем всякий раз, создавая новую версию этого файла, система увеличивает номер версии на 1. Номер версии увеличивается, если в каталоге пользователя уже есть файл с таким же именем и типом.

Например, всякий раз после редактирования файла программа *EDI* создает выходной файл для размещения в нем отредактированного входного файла (см.п.2.1). Спецификация выходного файла идентична спецификации входного файла за исключением того, что номер версии после редактирования увеличился на 1. Аналогично повторная компиляция одного и того же файла исходной программы также приводит к файлам объектных программ с одинаковыми спецификациями, за исключением различий в номерах версий.

Для каждой компоненты спецификации файла, за исключением имени файла, допускается использование значения по умолчанию:

ИМЯ _ УСТРОЙСТВА :

- логическое устройство "системный диск" (*SU* :);

[*GR, 4A*]

- код идентификации пользователя, установленный в команде *HEL* (п.1.3) либо определенный командой программы *MCR*

SET /UIC = [ГР, ЧЛ]

ИМЯ — ФАЙЛА

- указывается всегда (нет умолчания);

.ТИП

- зависит от командной строки, в которой используется спецификация файла;

; ВЕРСИЯ

- для входных файлов - максимальная существующая версия, для выходных - максимальная существующая версия плюс 1.

Внимание! Некоторые команды требуют явного указания номера версии.

3.2. ПРОГРАММА РАБОТЫ С файлами

Служебная программа *PIP* используется для выполнения разнообразных манипуляций с файлами. Содержание операций, выполняемых программой, определяется значениями ключей, сопровождающих спецификации файлов в командной строке *PIP*.

Вызов программы: *>RUN X PIP <BK> ИЛИ >PIP <BK>*

Ответ программы: *PIP >*

Командная строка:

PIP>[ВЫХОДНОЙ-ФАЙЛ[/КЛЮЧИ]]=ВХОДНОЙ-ФАЙЛ1[ВХОДНОЙ-ФАЙЛ2,...][/КЛЮЧИ]

В командной строке компоненты ВЫХОДНОЙ — ФАЙЛ, ВХОДНОЙ — ФАЙЛ 1 и т.д. суть спецификации соответствующих файлов.

Для прекращения работы программы *PIP* необходимо в ответ на подсказку *PIP>* ввести управляющий символ *УС/З*.

Опишем ключи, наиболее часто используемые в командных строках *PIP*, и приведем краткие примеры их использования.

/LI

- вывести каталог файла пользователя, например:

PIP>DK1:/LI<BK> - по умолчанию на терминал (при выделенном для резиденции пользователя диске *DK1*);

PIP>LP:=DK1:/LI<BK>- на устройство широкой печати.

/FR

- вывести число свободных блоков на носителе, например на терминал:

PIP>DXφ:/FR<BK>

/DE

- стереть файл (необходимо обязательно указывать тип и версию файла!), например:

PIP > TEST2. FIN; 4/DE < BK >

/PU

- стереть все версии файла кроме последней, например:

PIP > AC. MAC/PU < BK >

/RE

- переименовать входной файл в соответствии со спецификацией выходного (указание версии обязательно !), например :

PIP > AB. FTN; 1=BA. FTN; 25/RE < BK >

/SP

- вывод файла на LP : с созданием промежуточного буфера.

При работе с программой *PIP* возможна замена какой-либо компоненты спецификации обрабатываемого файла (кроме имени устройства) на символ * , который имеет смысл "все". Например:

*.FTN;1 - все файлы типа . FTN версии 1;

AS.*;1 - все файлы с именем AS.FTN;

AS.FTN;* - все версии файла AS.FTN;

.FTN; - все файлы типа . FTN;

.;* - все файлы каталога.

Важнейшей операцией программы *PIP* является к о п и р о в а - н и е ф а й л о в. Командная строка, требующая выполнения операции копирования, не содержит ключей, входной файл подлежит копированию, а спецификация выходного файла указывает место назначения копии.

Например,

PIP > DK1: = ABC. MAP < BK >

Если в спецификации выходного файла, как в приведенном примере, опущены какие-либо компоненты (UIC , имя файла, тип или версия), то программа *PIP* принимает их по умолчанию равными соответствующим компонентам спецификации входного файла. Устройство, имя которого указывается в спецификации выходного файла, точнее, соответствующий том носителя должен быть установлен, монтирован, а также содержать требуемый каталог. Распределение устройств, их монтирование и создание каталогов выполняется специальными командами программы *MCR* (ALLOCATE, MOUNT и UFD).

Заметим, что вывод с помощью *PIP* файлов на терминал, устройство широкой печати является по-существу командами копирования:

PIP > TI: = AB. LST < BK >

PIP > LP: = AB. MAP < BK >

3.3. Программа преобразования файлов

Каждой операционной системе ЭВМ типа СМ-3,4 - ОСРВ, ДОС, РАФОС и др., свойствен собственный формат файлов. Транспортирование данных и программ между различными операционными системами возможно лишь при условии надлежащего преобразования форматов соответствующих файлов. Это преобразование, а также ввод/вывод файлов на некоторые типы внешних устройств (например устройство ввода/вывода перфоленты) могут быть выполнены в ОСРВ с помощью служебной программы *FLX*.

Вызов программы : *>RUN x FLX <BK> или >FLX <BK>*

Ответ программы : *FLX >*

Командная строка :

FLX > Выходной_Файл [Ключи] = ВХОДНОЙ_ФАЙЛ [КЛЮЧИ]

Здесь Выходной_ФАЙЛ, ВХОДНОЙ_ФАЙЛ суть спецификации соответствующих файлов.

Ключи управления форматом:

- /DO* - файл в формате ДОС,
- /RS* - " - ОСРВ,
- /RT* - " - РАФОС.

Ключи представления (*n* - длина блока, при отсутствии *n* файл переносится с переменной длиной блока):

- /FA:n* - символьное представление,
- /FB:n* - двоичное представление (представление файла объектной программы; максимальное *n* = 512),
- /IM:n* - дампинг-представление (представление файла образа задачи).

Ключи управления файлами:

- /BL:n* - количество связанных блоков,
- /CO* - файл, должен занимать непрерывный участок носителя,
- /DE* - стирание файлов в формате ДОС или РАФОС,
- /DI* - выдача каталога файлов носителя систем ДОС или РАФОС,
- /LI* - см. */DI*,
- /ZE* - стирание или новая инициализация носителя систем ДОС или РАФОС.

Примеры использования программы FLX

I. Копирование данных формата ДЭС на носитель ОСРВ;

```
FLX > DK1: /RS = DK3: TA.FTN /DO < BK >
```

II. Чтение перфоленты на носитель ОСРВ :

```
FLX > DK2: /RS = PR : СПЕЦИФИКАЦИЯ-ФАЙЛА /DO < BK >
```

III. Вывод на перфоленту с носителя ОСРВ :

```
FLX > PP : /DO = DK : TS2.FTN ; 5 /RS < BK >
```

IV. Копирование файла, представленного в символьном виде и формате РАФЭС, на носитель ОСРВ :

```
FLX > DK5: /RC /FA = DX1 : ABC /RT /FA < BK >
```

4. КОСВЕННЫЙ КОМАНДНЫЙ ПРОЦЕССОР

Наличие в ОСРВ косвенного командного процессора (ККП) позволяет программировать работу с программой MCR и служебными программами. Такая программа создается с помощью текстового редактора в виде косвенного командного файла (тип .CMD) и не подлежит компиляции. ККП обрабатывает наравне с собственными инструкциями все команды программы MRC, командные строки служебных программ EDI, FOR, ТКВ, PIP и FLX. Кроме того, ККП допускает интерпретацию вложенных косвенных командных файлов (глубина вложенности - до 4).

Вызов ККП: @СПЕЦИФИКАЦИЯ_ФАЙЛА [КЛЮЧИ]

Здесь СПЕЦИФИКАЦИЯ_ФАЙЛА указывает подлежащий интерпретации косвенный командный файл (имя - от 1 до 6 алфавитно-цифровых символов, тип .CMD).

Прокомментируем наиболее часто используемые инструкции собственно ККП.

.ASK СИМВОЛ ТЕКСТ-ВОПРОСА

- вывести на экран ТЕКСТ_ВОПРОСА с возможностью однозначного ответа [YES/NO]; при ответе пользователя "YES" СИМВОЛ получает значение 1, 0 - в противоположном случае.

.IFT СИМВОЛ.КОМАНДА

- выполнить инструкцию КОМАНДА, если СИМВОЛ равен 1; например,

.IFT A .GOTO 1000

.IFF СИМВОЛ.КОМАНДА

- *AND*

- логическая операция "И"; например,

. *IFT A. AND B*

. *OR*

- логическая операция "ИЛИ".

; КОММЕНТАРИЙ

- знак ";" открывает строку-комментарий.

Составитель - Сергей Викторович Смирнов

ИСПОЛЬЗОВАНИЕ ВАЖНЕЙШИХ КОМАНД И КЛЮЧЕЙ
СЛУЖЕБНЫХ ПРОГРАММ ОСРВ СМ ЭВМ

Редактор Е.Д.Антонова
Техн. редактор Н.М.Каленюк
Корректор Н.С.Куприянова

Подписано в печать 22.05.85. Формат 60x84 1/16.
Бумага оберточная белая. Печать оперативная.
Усл.п.л. 1,48. Уч.-изд.л. 1,4. Т.500 экз.
Заказ 4215 Бесплатно.

Куйбышевский ордена Трудового Красного Знамени авиационный
институт имени академика С.П.Королева, г.Куйбышев, ул.Исследо-
договардейская, 151.

Областная тип.им.В.П.Мяги, г.Куйбышев, ул.Венцека, 60.