



9.1. Структура ОС ДВК. 1. Монитор; 2. драйвер; 3. Компоновщик; 4. Редактор; 5. Интерфейс с файлами; 6. Утилиты; 7. Средства разработки; 8. Ядро; 9. Пользовательские программы.

## ВВЕДЕНИЕ В ОС ДВК

К числу распространенных ВС на базе МП К1801 относятся *диалоговые вычислительные комплексы* (ДВК). ДВК содержат дисплей, принтер, НГМД и одноплатную микро-ЭВМ, работающую под управлением операционной системы. Для подобных ВС наиболее хорошо зарекомендовали себя ОС **RT-11** (от англ. Real Time system — система реального времени) и ее разновидности, носящие названия **РАФОС** (Разделения Функций ОС), **ФОДОС** (Фоново-Оперативная Дисковая ОС), **ОС ДВК** и др. В состав этих ОС, обладающих модульным характером, может входить один из нескольких мониторов, среди которых есть и многозадачные (мультипрограммные). ОС ДВК обычно строится на базе *однозадачного монитора RT11SJ* (Single Job — одно задание).

Эта глава в общих чертах поясняет структуру ОС ДВК, правила использования несложных команд управления системой, работу типичного редактора текстов, этапы разработки программ для ВС.

### 9.1 Структура ОС ДВК

ОС ДВК состоит из головной управляющей программы (монитора), драйверов внешних устройств, утилитных программ и средств программирования на языках высокого уровня (рис. 9.1).

*Монитор* — ядро, координирующее работу всех остальных программ, составляющих ОС и осуществляющее связь с оператором (пользователем) через терминал. Богатый набор команд монитора, набираемых пользователем на клавиатуре терминала, по-

зволяет инициировать с их помощью все необходимые действия по управлению системой.

*Драйверы внешних устройств* — это специализированные программы, предназначенные для управления конкретными периферийными устройствами системы.

Монитор исполняет самостоятельно лишь небольшое количество простейших команд, для выполнения остальных он «призывает на помощь» отдельные системные программы, называемые *утилитными*. В их числе:

1) *редактор текста*, позволяющий ввести, модифицировать и записать на ВЗУ текстовый материал — это может быть программа или любой другой текстовый документ;

2) *утилиты файловой службы*, которые позволяют пересылать данные с одних ВУ на другие, копировать, стирать, обновлять наборы данных и т. д.;

3) *программа-компоновщик*, осуществляющая «сборку» программ из нескольких полученных в результате трансляции двоичных (объектных) модулей.

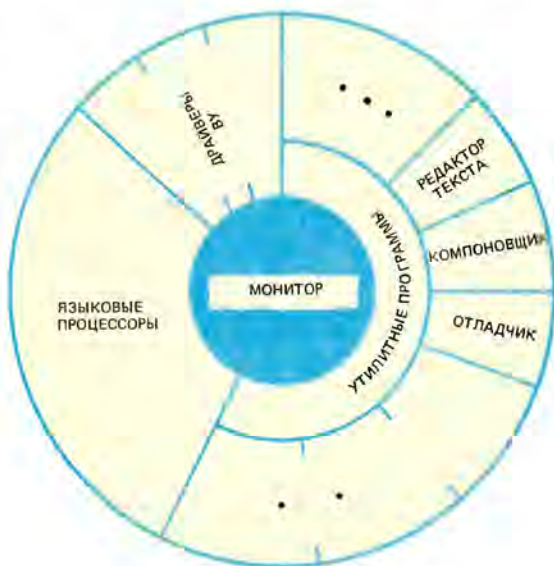
Специальные программы, называемые *языковыми процессорами*, позволяют разрабатывать и выполнять программы, написанные на различных языках программирования.

В системе ОС ДВК могут быть использованы различные пакеты *прикладных программ*: математические, статистические, экономические, системы автоматизации проектирования и т. д.

Все компоненты ОС ДВК хранятся на гибком магнитном диске и вызываются в ОЗУ ЭВМ по мере необходимости. Этим процессом управляет монитор, находящийся в ОЗУ постоянно с момента первоначального запуска системы, т. е. *начальной загрузки*. В ее

Рис. 9.1.

Структура ОС ДВК типична для многих операционных систем — это стройный набор «инструментов» программирования, помогающий создавать сложные и эффективные программы.



ходе нужно установить диск с операционной системой в накопитель и запустить программу начальной загрузки ОС, находящуюся в ПЗУ ЭВМ. По окончании загрузки монитора ОС ДВК на экране видеотерминала появляется сообщение вида:

```
RT11SJ V05.00
.SET TT SCOPE
```

Символ . (точка) в левой части экрана в конце сообщения означает, что загрузка окончена, т. е. монитор ОС ДВК находится в ОЗУ ЭВМ и ждет команд пользователя.

## 9.2. Управление системой. Командный язык .

*Команды ОС ДВК* строятся из английских ключевых слов и аргументов. Для удобства работы большинство ключевых слов можно сокращать до двух-трех первых букв. Каждую команду следует завершать нажатием клавиши ВК (возврат каретки), что инициирует выполнение требуемой функции. До нажатия ВК ошибки при наборе командной строки можно исправить с помощью клавиш ЗБ (забой — удаление последнего символа в строке) и СУ/У (удаление ошибочно набранной строки целиком). Обозначение типа СУ/У означает, что при нажатой клавише СУ нажимается клавиша У.

Если при вводе команды в ней пропущены какие-либо обязательные элементы или аргументы, то монитор в процессе обработки команды задает пользователю вопросы относительно их значений. Исполнив команду, монитор печатает точку в новой строке, что указывает на готовность принять очередную команду. В качестве примера рассмотрим простейшие команды TIME и DATE, служащие для установки или распечатки, соответственно, текущего времени и даты. Они подаются в следующем виде:

```
.TIME [<ЧЧ:ММ>]*
.DATE [<ЧЧ-МММ-ГГ>]
```

Если команды подаются без аргументов, то монитор сообщает текущее время (дату):

```
.TIME 15:00 <ВК>**
.DATE 3-OCT-86 <ВК>
.DATE <ВК>
3-OCT-86
.TIME <ВК>
15:00:09
```

---

\* Здесь и далее аргументы команд, требующие конкретного задания, и символ возврата каретки заключаются в угловые скобки для отличия от ключевых слов, и дополнительно в квадратные скобки, если аргументы можно опустить.

\*\* Символы, вводимые пользователем, выделены шрифтом.

## Файлы и операции с файлами

### Наименование файлов и устройств

Обычно данные хранятся на ВЗУ в форме файлов. *Файлом* называется произвольная группа данных, рассматриваемая как единое целое. Это может быть текст программы, программа в двоичном коде, письмо, сообщение, библиотека подпрограмм и т. п. Каждый файл имеет свое уникальное *наименование*, состоящее из системного имени устройства, на котором файл хранится, имени и типа файла, разделенных символом «точка». *Тип файла* может задаваться произвольно, но обычно он несет информацию о содержании файла, например:

SYS — монитор, драйвер, или служебный файл ОС ДВК;

BAS — текст программы на языке BASIC-11;

FOR — текст программы на языке FORTRAN-IV;

MAC — текст программы на языке MACRO-11;

OBJ — двоичный модуль — результат трансляции программы;

SAV — готовая к выполнению программа;

TXT — произвольный текстовый материал и т. п.

*Системное имя устройства* задается в виде «DDn:», где DD — двухбуквенный шифр типа устройства, а n — его порядковый номер (указывается лишь при наличии в системе нескольких одно-типных устройств), например:

DX0: — накопитель на гибких магнитных дисках номер 0;

MX1: — накопитель на гибких мини-дисках номер 1;

LP: — устройство печати;

TT: — терминал пользователя;

CT0: — накопитель на кассетной магнитной ленте номер 0;

PR: — считыватель с перфоленты;

PP: — ленточный перфоратор;

RK: — накопитель на жестких магнитных дисках емкостью 2,5 Мбайт и т. п.

Помимо этого, в ОС ДВК можно пользоваться *псевдонимами устройств*. Иными словами, с конкретным физическим устройством можно отождествить произвольное имя и затем обращаться к нему по этому имени. Указанная операция осуществляется командой ASSIGN:

.ASSIGN <имя устройства> <псевдоним>.

Специальный псевдоним DK: означает «рабочее устройство (диск) пользователя». В наименованиях хранящихся на нем файлов имя устройства может быть опущено (умолчание). В большинстве команд, оперирующих с файлами, допускается употребление вместо имени и/или типа файла символа «\*», означающего «любое имя/тип файла», например, запись MX0:\*. SYS подразумевает все файлы, хранящиеся на устройстве MX0: и имеющие тип SYS.

Список файлов (*каталог*), хранимых на ВЗУ, можно получить с помощью команды DIRECTORY. В каталог включается также информация о размере и дате создания каждого файла:

```

.DIR MX0: <BK>
SWAP .SYS 26 19-APR-85 TT .SYS 2 19-APR-85
DX .SYS 2 19-APR-85 MX .SYS 9 19-APR-85
LP .SYS 2 19-APR-85 DIR .SAV 17 19-APR-85
DUP .SAV 21 19-APR-85 PIP .SAV 16 19-APR-85
LINK .SAV 29 19-APR-85 MACRO .SAV 45 19-APR-85
LT .SYS 5 19-APR-85 FORMAT .SAV 13 19-APR-85
CREF .SAV 6 19-APR-85 DUMP .SAV 7 19-APR-85
ODT .OBJ 10 19-APR-85 SYSMAC .SML 54 19-APR-85
LDA .SAV 5 19-APR-85 VRF .SAV 3 19-APR-85
RT11SJ.SYS 70 19-APR-85 SP21 .SAV 21 19-APR-85
K52 .SAV 55 19-APR-85 CAMIN 7 19-APR-85
22 FILES, 425 BLOCKS
7 FREE BLOCKS

```

Размер файлов указывается в блоках по 512 байт. По желанию можно получить частичный каталог, включающий в себя один или несколько файлов, имеющих определенное имя и/или тип. Поясним сказанное на примерах:

```

.DIR MX0:* .SYS <BK>
SWAP .SYS 26 19-APR-85 TT .SYS 2 19-APR-85
DX .SYS 2 19-APR-85 MX .SYS 9 19-APR-85
LP .SYS 2 19-APR-85 LT .SYS 5 19-APR-85
RT11SJ.SYS 70 19-APR-85
7 FILES, 116 BLOCKS
7 FREE BLOCKS

```

```

.DIR RT11SJ.* <BK>
RT11SJ .SYS 70 19-APR-85
1 FILES, 70 BLOCKS
7 FREE BLOCKS

```

Команда, подаваемая в виде DIRECTORY/PRINTER, позволяет распечатать каталог на печатающем устройстве.

#### Копирование, переименование, удаление и вывод файлов на печать

Для копирования файлов с одного устройства на другое существует команда COPY, подаваемая в следующем виде:

**.COPY/SYS <наименования входных файлов> <наименования выходных файлов>**  
Например, команда

**.COPY MX0: PIP.SAV MX1:**  
создает копию файла PIP.SAV, хранящегося на устройстве MX0:, на устройстве MX1: под тем же именем. Ключ/SYS обязателен при копировании системных файлов, имеющих тип SYS. Например, для создания копии всего диска с операционной системой подается команда

**.COPY/SYS MX0: \*.\* MX1:**

Удаление (стирание) ненужных файлов производится командой `.DELETE/NOQ` [список стираемых файлов].

При удалении файла запись о нем вычеркивается из каталога и место, отводимое под него на диске, объявляется свободным.

Команда `RENAME` (переименовать) служит для изменения имени или типа файла:

`.RENAME` <старое имя файла> <новое имя файла>.

В заключение беглого обзора файловых операций приведем команды для вывода содержимого текстовых файлов на экран видеотерминала и на печать:

`.TYPE` <наименование файла> — вывод файла на экран (TT:);

`.PRINT` <наименование файла> — вывод файла на устройство печати (LP0:).

#### 9.4. Создание и редактирование текстовых файлов

В текстовых файлах может храниться самая разнообразная информация: программы на каком-либо языке программирования, таблицы, исходные данные для какой-либо программы, списки, письма, напоминание, статьи и т. п. Для создания и редактирования файлов используются редакторы текстов. Наиболее удобны т. н. «экранные» редакторы, ориентированные на работу с определенными типами алфавитно-цифровых видеотерминалов и широко использующие их собственные аппаратные возможности. К ним относится и редактор K52, входящий в состав ОС ДВК.

Редактор K52 создает для пользователя иллюзию, что экран видеотерминала — это «окно» размером 24 строки в редактируемом файле, причем положением «окна» можно управлять с помощью клавиш. Мы не будем рассматривать все функции и возможности K52, поскольку их очень много и они подробно описаны в соответствующей документации, а остановимся лишь на необходимом минимуме.





##### Запуск редактора текста

Редактирование или создание текстового файла начинается с подачи команды `EDIT`:




`.EDIT/K52/CREATE` <наименование файла>.

Ключевое слово «/CREATE» информирует K52 о том, что пользователь намерен создать новый файл с указанным в команде наименованием. При отсутствии ключа «/CREATE» подразумевается, что файл уже существует и в нем необходимо сделать изменения или добавления.

При создании файла редактор очищает экран терминала, устанавливает курсор в левый верхний угол экрана, после чего пользователь может начать ввод текстов в файл, завершая каждую строку, как и на обычной пишущей машинке, нажатием клавиши ВК (возврат каретки). В случае, если существующий файл редактируется или дополняется, на экране появляются его первые 24 строки. Все операции по внесению изменений в файл начинаются

с установки курсора в место предполагаемого изменения. Для этого служат клавиши с обозначениями , , , \*. Нажатие какой-либо из них перемещает курсор на одну позицию в указанном на клавише направлении.




#### Удаление ошибочно введенного символа

Удаление символов производится клавишами  и . При нажатии первой удаляется символ, указываемый курсором, а нажатие клавиши  удаляет символ слева от курсора (это удобно при вводе текста). При удалении символа правая от курсора часть строки сдвигается влево.



#### Вставка символов

Для вставки символов никаких специальных действий не требуется. Нажатие клавиши на алфавитно-цифровой клавиатуре вызывает появление символа в позиции курсора, причем текущая строка раздвигается автоматически.



#### Удаление строки



После установки курсора на строку, подлежащую удалению, требуемая операция производится последовательным нажатием клавиш , , . При этом текст под курсором сдвигается на одну строку вверх.



#### Вставка строки



Комбинация клавиш  и  вызывает появление новой (пустой) строки под текущей позицией курсора. Текст, расположенный ниже курсора, при этом сдвигается вниз.

#### Перемещение «окна»


В небольших пределах «окно» можно переместить при помощи клавиш управления курсором  и . Если курсор «упирается» в верхнюю или нижнюю границу экрана, то «окно» автоматически сдвигается. Для более быстрого перемещения «окна» по тексту имеются следующие команды:


,  — перемещение «окна» в конец файла;

,  — перемещение «окна» в начало файла;

,  — перемещение «окна» вперед-назад на 16 строк.

Направление перемещения в последней команде задается следующими командами:

 — установить режим сдвига «вперед»;

 — установить режим сдвига «назад».

---

\* Все клавиши, применяемые для операций редактирования текста и управления курсором, находятся на служебной клавиатуре видеотерминала справа от основной. Обозначение клавиш соответствует клавиатуре видеотерминала «Электроника 15ИЭ-00-013».

По окончании редакции/ввода текста необходимо выполнить следующую процедуру:

1) последовательно нажать клавиши  $\downarrow$  и  $\overline{7}$ , после чего в верхней строке экрана появится приглашение COMMAND;

2) подать команду EXIT, завершаемую нажатием клавиши  $\overline{\chi}$ . Это вызывает запись отредактированного текста в файл на диске, после чего на нижней строке экрана появляется символ «\*»;

3) при нажатой клавише  $\overline{CV}$  нажать клавишу  $\overline{C}$ . Работа редактора прекращается и управление передается монитору ОС ДВК.

### 9.5. Использование языков программирования

Существуют сотни языков программирования различных уровней, одни из которых являются проблемно-ориентированными, т. е. предназначенными для решения узкого круга задач в определенной области человеческой деятельности, другие более или менее универсальны. Из языков высокого уровня в ОС ДВК чаще всего используют BASIC и FORTRAN. Универсальный язык низкого уровня (Ассемблер) для рассматриваемых ЭВМ называется **MACRO-11**.

Подготовка программы на языках программирования с трансляторами компилирующего типа, в том числе на MACRO-11 и FORTRAN, включает в себя следующие этапы:

1) составление исходного текста программы на выбранном языке (кодирование программы);

2) ввод текста программы в ЭВМ с помощью редактора текста и запись его в файл на ВЗУ;

3) обработку текста соответствующим языковым процессором и получение в результате двоичного объектного модуля;

4) компоновку готовой к выполнению программы из одного или нескольких объектных модулей;

5) запуск программы.

При использовании компиляторов этапы 2)...4) выполняются посредством отдельных системных программ; если программа пишется на языке BASIC, соответствующие функции осуществляет сам интерпретатор.

#### Трансляция программы

Трансляция программы, написанной на языке MACRO-11, осуществляется с помощью команды MACRO:

.MACRO <наименование файла с текстом программы>.

Пусть программа пользователя хранится на устройстве ДК: в файле SUM.MAC. Тогда команда .MACRO SUM приведет к трансляции программы, а ее результат (двоичный *объектный модуль*) будет записан в файл «ДК:SUM.OBJ». При желании можно получить, помимо объектного модуля, и листинг, т. е. распечатку программы. В этом случае подаются команды:



- .MACRO SUM/LIST:TT: — для выдачи листинга на экран терминала;
- .MACRO SUM/LIST — для записи листинга в виде файла с именем SUM.LST на устройство ДК:.

Совершенно аналогично записываются команды для трансляции программы на языке FORTRAN-IV:

- .FORTRAN SUM
- .FORTRAN SUM/LIST: TT:
- .FORTRAN SUM/LIST

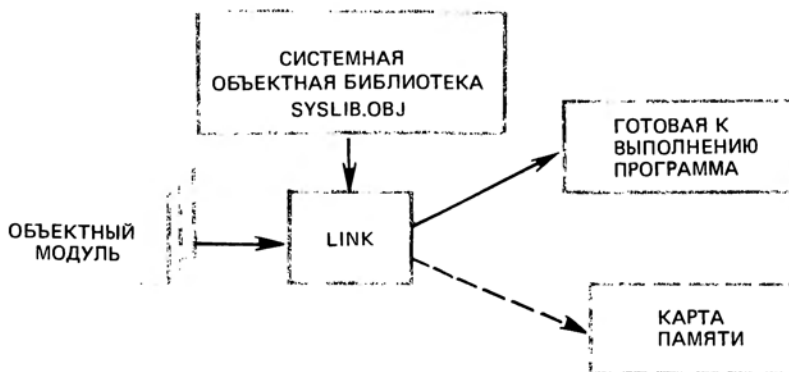
Файл SUM должен иметь тип FOR.

### Компоновка и запуск программы

Результат перевода исходного текста компилятором еще не представляет собой готовую к выполнению программу. Во-первых, сложную программу удобно собирать из нескольких модулей исходного текста, создаваемых и транслируемых отдельно. Во-вторых, программа может содержать обращения к хранящимся на ВЗУ библиотекам, содержащим стандартные и системные модули для процедур типа вычисления функций, ввода-вывода и т. д. В-третьих, во многих ОС заранее может быть неизвестно место в ОЗУ, где будет выполняться программа. Таким образом, для получения рабочей программы нужно объединить ее объектные модули, в том числе библиотечные, установив необходимые связи между ними, настроить адреса каждого модуля для выполнения в определенном месте ОЗУ, распределить память под рабочие области и массивы.

Эти и некоторые другие функции осуществляет *программа-компоновщик LINK* (рис. 9.2), команда запуска которого имеет вид:

В задачу утилиты LINK входит создание готовой к выполнению программы из одного или нескольких объектных модулей с автоматическим присоединением, если это необходимо, готовых модулей из системной библиотеки.



.LINK <список объектных модулей>/MAP.

Результатом работы компоновщика является файл, содержащий готовую к запуску и выполнению программу, а если указано ключевое слово /MAP, то и файл, описывающий распределение памяти между отдельными модулями, а также общий объем памяти, необходимый программе («карта памяти»). Выходным файлом присваивается имя первого из указанных в команде объектных модулей, причем файл с программой имеет тип «.SAV», а файл распределения памяти — «.MAP».

Запуск программы на выполнение производится командой .RUN <наименование файла>.

Интересная особенность программы LINK — возможность компоновки с перекрытиями, т. е. когда суммарный объем всех модулей написанной программы превышает допустимый верхний предел — около 28 К машинных слов. При этом в ОЗУ помещается не вся программа, а лишь ее основной модуль (корневой сегмент) и необходимые на данный момент подпрограммы, а остальные находятся на ВЗУ и загружаются в ОЗУ по мере необходимости, замещая при этом «отработавшие» модули.

Применение перекрытий в значительной мере снимает ограничение на максимальный размер программы. Недостатки этого механизма — некоторое замедление выполнения программы вследствие затрат времени на загрузку перекрытий с ВЗУ, а также необходимость соблюдения специфической организации программы.