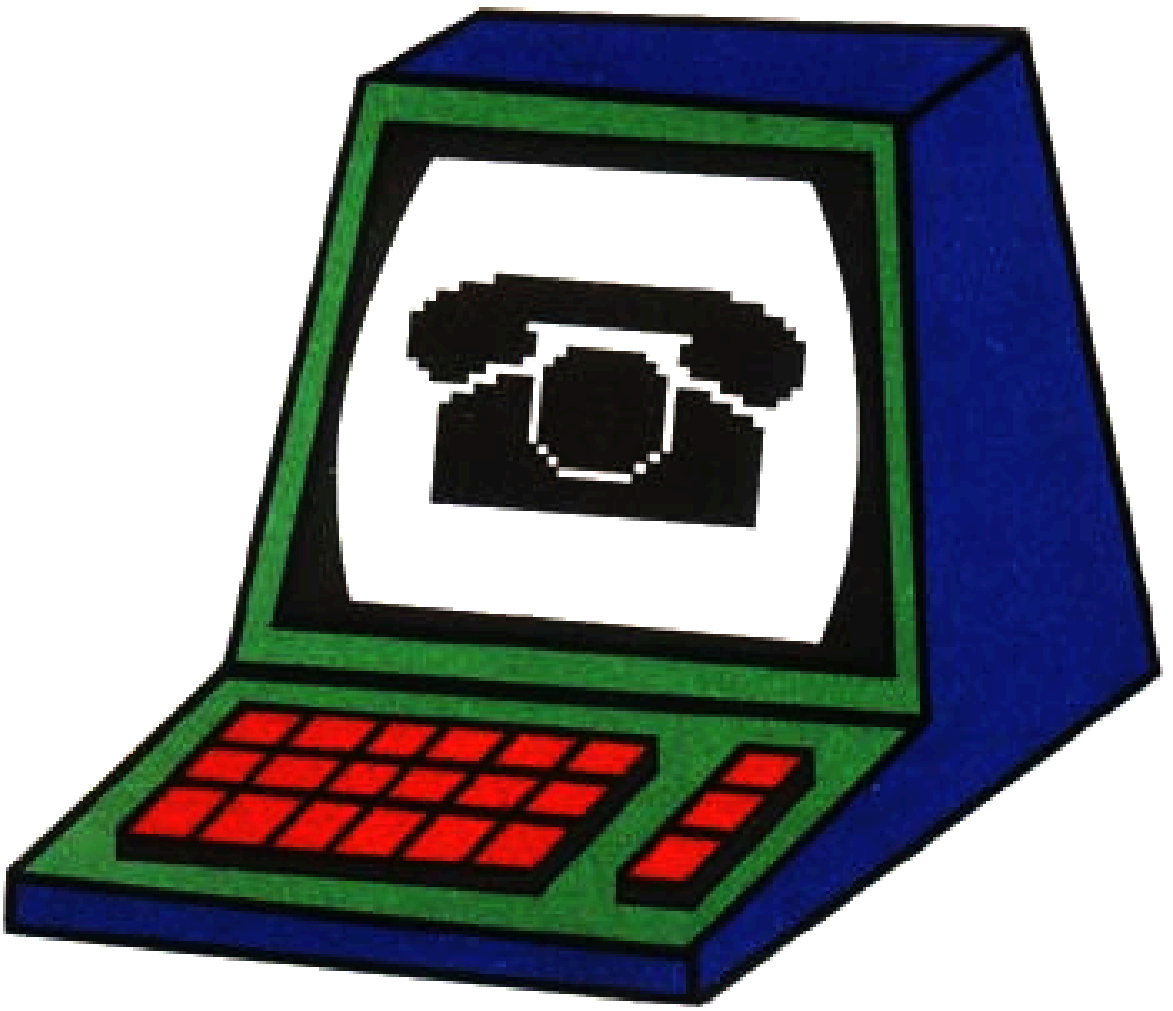
С.Н. ПОПОВ

ПЕРСОНАЛЬНЫЙ КОМПЬЮТЕР МИКРОША И ЕГО ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ



Попов Сергей Николаевич

ПЕРСОНАЛЬНЫЙ КОМПЬЮТЕР "МИКРОША" И ЕГО ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

НПК "ЭМИС"

Москва 1990

ВВЕДЕНИЕ

Персональный компьютер (ПК) МИКРОША выпускается с 1986 года и в настоящее время довольно широко распространен. Популярность МИКРОШИ во многом определяется его программной совместимостью с радиолюбительским персональным компьютером РАДИО-86РК. Его полное описание было опубликовано в журнале РАДИО, и при наличии печатной платы он мог быть повторен радиолюбителем со средней квалификацией. В настоящее время в СССР появляется все большее количество ПК, совместимых с ПК IBM PC. Конечно, характеристики ПК МИКРОША и ПК IBM PC несопостовимы, однако несопостовима и их стоимость. Кроме того, в стране выпускается несколько ПК, так или иначе программно совместимых с МИКРОШЕЙ (КРИСТА, АПОГЕЙ БК-01, ПАРТНЕР, ЭЛЕКТРОНИКА КР-01 и другие). Таким образом, материал данной книги может быть интересен владельцам подобных ПК.

Одной из особенностей ПК МИКРОША является широкое использование в нем больших интегральных схем (БИС). Эти БИС имеют множество режимов работы, задаваемых программно. Правильное использование этих особенностей позволяет получать новые качественные характеристики, например, при работе в графическом редакторе, описываемом во второй главе.

В ПК МИКРОША предусмотрена возможность расширения функциональных возможностей путем подключения дополнительных модулей как через системный разъем,

так и через разъем параллельного порта. В первой главе приводятся необходимые справочные сведения о составе и назначении сигналов на разъемах ПК.

Программное обеспечение ПК, как правило, разделяется на системное и прикладное. В данной книге предпочтение отдано системному обеспечению МИКРОШИ, так как умелое его использование позволит читателю более быстро и эффективно разрабатывать собственнные прикладные и системные программы.

В связи с небольшим предполагаемым объемом книги в ней во многих случаях приводится только краткая информация по техническим и программным средствам. Если необходимы более полные сведения, то читатель сможет их получить, обратившись к литературе, приведенной в конце книги.

1. ТЕХНИЧЕСКИЕ СРЕДСТВА ПК МИКРОША

1.1 Состав и назначение блоков ПК МИКРОША

Персональный компьютер МИКРОША выполнен на базе микропроцессорного комплекта БИС серии К580. Конструктивно все узлы компьютера располагаются на одной печатной плате. На этой же плате расположена клавиатура компьютера. ПК МИКРОША имеет следующие основные технические характеристики.

Тип микропроцессора	КР580ВМ80А
Разрядность шины данных	8
Разрядность шины адресов	16
Объем ПЗУ	2 Кбайт
Объем ОЗУ	32 Кбайт
Тактовая частота работы МП	1,77 МГц
Быстродействие	300000 оп./с
Алфавитно-цифровой режим	25 строк по 64 символа
Псевдографический режим	74x128 точек
Скорость записи/считывания	1200 бит/с

В состав ПК МИКРОША входит системный блок, содержащий печатную плату с основными элементами ПК, блок сетевого питания, устройство отображения на базе телевизора и устройство внешней памяти на базе кассетного магнитофона. К телевизору ПК может подключаться либо к антенному входу через высокочастотный модулятор, либо к входу "Видео". В последнем случае качество изображения значительно выше. Блок питания ПК формирует напряжения +5, +12 и -5 В, необходимые для питания всех узлов ПК.

На задней стенке системного блока расположен ряд разъемов, предназначенных для подключения питания, телевизора, магнитофона, принтера и блока расширения.

Рассмотрим кратко особенности построения основных блоков ПК, входящих в состав системного блока.

В состав блока микропроцессора входит тактовый генератор КР580ГФ24, микропроцессор КР580ВМ80А и ряд схем малой степени интеграции. Тактовый генератор стабилизирован кварцевым резонатором с частотой 16,3 МГц и формирует все необходимые для работы ПК частоты. Микропроцессор является основным управляющим компонентом ПК. Он формирует сигналы на шинах данных, адреса и управления. Особенностью ПК МИКРОША является использование упрощенной шины управления. В ПК обращение ко всем регистрам внешних устройств производится так же, как и к ячейкам памяти. Это обеспечивает большую гибкость при создании программ обслуживания внешних устройств. В таблице 1.1 показано распределение памяти в ПК МИКРОША.

ТАБЛИЦА 1.1

	Адрес
	0FFFFH
ПЗУ (Монитор), КПДП КР580ВТ57	0F800H
СВОБОДНО	0E000H
Таймер КР580ВИ53	0D800H
Контроллер дисплея КР580ВГ75	0D000H
ППА интерфейса ИРПР	0C800H
ППА (клавиатура, магнитофон)	0C000H
СВОБОДНО	08000H
ОЗУ (буфер экрана)	076C0H
ОЗУ (рабочие ячейки монитора)	07600H
ОЗУ (область программ пользователя)	00000H

Блок памяти ПК содержит ПЗУ и ОЗУ. В ПК используется одна микросхема ППЗУ с УФ-стиранием типа К573РФ2 объемом 2 Кбайт. В этом ПЗУ хранится основная управляющая программа ПК - программа монитор.

ОЗУ объемом 32 Кбайт выполнено на базе динамических БИС ОЗУ типа КР565РУ6Г с организацией 16Кх1. Часть оперативной памяти объемом 2,5 Кбайт используется как память отображения дисплея. Регенерация содержимого ОЗУ проводится в режиме прямого доступа к памяти, запросы на который формирует блок контроллера дисплея.

За счет использования внешних блоков расширения объем ОЗУ и ПЗУ ПК может быть увеличен соответственно до 8 и 48 Кбайт.

Блок контроллера дисплея содержит программируемый дисплейный контроллер КР580ВГ75, контроллер ПДП типа КР580ВТ57, ПЗУ знакогенератора К573РФ2 и ряд схем малой и средней степени интеграции. Дисплей ПК имеет два стандартных режима отображения, обслуживаемых программой монитор: алфавитно-цифровой и псевдографический. В алфавитно-цифровом режиме на экране отображается 25 строк по 64 символа в каждой. В псевдографическом режиме - 50 точек по вертикали и 128 точек по горизонтали. За счет перепрограммирования контроллера дисплея возможны и другие режимы отображения.

Знакогенератор дисплея содержит два набора символов. Первый набор содержит строчные латинские и русские буквы, а второй - строчные и прописный русские буквы. Одновременное отображение на экране символов из обоих наборов невозможно. Переключение наборов производится при помощи линии РВ7 ППА интерфейса ИРПР.

Кроме алфавитно-цифровых, знакогенератор содержит также ряд графических сиволов. Вывод символов на экран дисплея возможен как при помощи стандартной подпрограммы, расположенной в ПЗУ монитора, так и путем непосредственной записи кода символа в соответствующую ячейку буфера отображения.

В системном блоке ПК МИКРОША имеется громкоговоритель, подключенный к формирователю звуковых сигналов. В состав формирователя входит программируемый таймер КР580ВИ53 и усилитель. Выход канала 2 таймера подключен к схеме "И", на второй вход которой подается сигнал канала РС1 ППА клавиатуры. В выходу схемы "И" через усилитель подключен динамик. Вход разрешения счета канала 2 таймера подключен к выходу РС2 ППА клавиатуры. Таким образом, имеется возможность программного управления формирователем звуковых сигналов.

Клавиатура ПК МИКРОША выполнена в виде матрицы нормально разомкнутых контактов и содержит 68 клавиш. Клавиатура подключена к ПК при помощи ППА типа КР580ВВ55А. Все действия, связанные с опросом состояния клавиатуры, формированием кода нажатой клавиши и организацией режима автоповтора выполняются программными средствами ПК. Таким образом, имеется возможность гибкой программной перестройки режимов работы клавиатуры.

Клавиша "УС" предназначена для формирования кодов, помещенных в первых двух колонках стандартной таблицы кодов, а клавиша "РУС/ЛАТ" - для формирования кодов русских либо латинских букв.

В качестве устройства внешней памяти в ПК МИКРОША используется кассетный магнитофон. Запись цифровой информации производится с использованием двухфазного кодирования. Большинство действий по записи и чтению информации с магнитной ленты выполняется программными средствами ПК. Используемый способ записи позволяет записать на одну кассету типа МК60 до 300 Кбайт информации. Вывод информации на магнитную ленту производится через линию PC0, а ввод - через линию PC7 ППА клавиатуры.

1.2. Программно-доступные компоненты ПК

С точки зрения программиста весь ПК можно рассматривать как набор программно доступных регистров. Часть регистров являются внутренними регистрами микропроцессора. Обращение к этим регистрам возможно только из программ написанных в машинных кодах или на языке АССЕМБЛЕР. Другая группа регистров - это ячейки памяти ПК. Обращение к ним, в отличие от обращения к внутренним регистрам микропроцессора, возможно и из программ, написанных на языках высокого уровня, таких как Бейсик или язык С. Третья группа регистров - это регистры ввода-вывода информации и внутренние регистры периферийных БИС. Обращение к этим регистрам возможно как путем использования специальных машинных команд ввода-вывода в языке АССЕМБЛЕР или с помощью операторов вывода в языках высокого уровня.

В ПК МИКРОША обращение к регистрам ввода-вывода производится точно так же, как и к ячейкам памяти. Это означает, что при вводе и выводе можно использовать любые команды, предназначенные для чтения или записи информации в ячейки памяти. Такое решение позволило упростить схему ПК и предоставить программисту удобные средства управления вводом-выводом информации.

Дешифратор памяти в ПК построен по двухступенчатой схеме. Первая ступень дешифратора делит общее адресное пространство размером 64 Кбайт на 4 блока по 16 Кбайт каждый. Вторая ступень дешифратора делит старшие 16 Кбайт еще на 8 блоков по 2 Кбайта. Три вывода

дешифратора второй ступени остаются свободными и могут быть использованы для подключения дополнительных ПЗУ или периферийных БИС. В таблице 1.2 приведены справочные сведения об адресах и назначении регистров ввода-вывода ПК МИКРОША.

1.3 Подключение к ПК внешних устройств

Стандартными внешними устройствами для ПК МИКРОША являются: клавиатура, магнитофон и дисплей. Их подключение не вызывает особых трудностей и описано в руководстве по эксплуатации ПК. Для получения более высокого качества изображения на экране телевизора целесообразно отказаться от использования модулятора и подавать видеосигнал непосредственно на видеовход телевизора. Это, как правило, требует небольших доработок телевизора.

На разъеме подключения телевизора выведены следующие сигналы:

1 - напряжение питания модулятора, 2 - общий провод, 3 - видеосигнал.

На задней панели корпуса ПК имеется разъем параллельного интерфейса, который может быть использован для подключения различных внешних устройств и, прежде всего, принтера. На этот разъем выведены сигналы от ППА интерфейса. Следует иметь в виду, что в ПК не установлены буферные элементы на выходах ППА интерфейса, поэтому нагрузочная способность этих линий позволяет подключить к ним не более одного ТТЛ входа. Это также ограничивает длину соединительного кабеля (не более 1 метра).

После запуска ПК резидентный монитор настраивает ППА интерфейса в режим 0, причем каналы А и В - на вывод, а канал С - на ввод информации. Распределение контактов разъема параллельного интерфейса приведено в таблице 1.3.

Рассмотрим в качестве примера подключение принтера имеющего параллельный интерфейс типа CENTRONICS к ПК МИКРОША.

На разъем принтера, кроме входов данных, выведены сигналы управления обменом информацией и сигналы управления принтером. Не все из них абсолютно необходимы, поэтому ограничимся минимальным набором сигналов. В таблице 1.4 указана распайка кабеля для подключения принтера.

Для подключения принтера задействованы каналы А и С ППА интерфейса. Канал А служит для выдачи данных, С4 - для формирования сигнала STROBE, а С2 - для приема сигнала готовности принтера.

В таблице 1.5 приведена подпрограмма для выдачи символов на печать. Код печатаемого символа перед вызовом подпрограммы должен быть помещен в регистр С микропроцессора. В начале основной программы должна быть проведена инициализация ППА интерфейса при помощи следующих команд:

```
MVI     A,81H       порт А - вывод, В - вывод
                    С старш.-ввод, С младш.-вывод
STA     0C803H      запись в регистр управляющего
                    слова ППА интерфейса
MVI     A,9         команда установки бита С4
STA     0C803H      запись в регистр управляющего
                    слова ППА интерфейса
```


ТАБЛИЦА 1.2

Адрес	Тип БИС	Назначение регистра
C000H	КР580ВВ55А ППА клавиатуры	Чтение ответа клавиатуры
C001H		Запись кода сканирования контактов матрицы клавиатуры
C002H		4 младших разряда - запись. Разряд 0 - вывод на магнитофон; Р.1 - разрешение звука, Р.2 - управление счётчиком 2 таймера (разрешение/запрет счета), Р.3 - включение светодиода РУС/ЛАТ 4 старших разряда - чтение. Р.4 - чтение с магнитофона, Р.5 - клавиша РУС/ЛАТ Р.6 - клавиша УС, Р.7 - клавиша НР
C003H		Регистр управляющего слова ППА клавиатуры
C800H	КР580ВВ55А ППА интерфейса	Канал А ППА, настроен на вывод, заведен на разъем
C801H		Канал В ППА, настроен на вывод, младшие 5 разрядов заведены на разъем. Р5 и Р.6 задействованы. Р.7 - переключение знакогенератора (прописные/строчные буквы).
C802H		Канал С ППА, настроен на вывод, Р.0 - используется в интерфейсе локальной сети, Р.1 - подключен к выходу счетчика 1 таймера КР580ВИ53
C803H		Регистр управляющего слова ППА интерфейса
D000H	КР580ВГ75 Контроллер дисплея	Чтение/запись параметров видеоконтроллера
D001H		Запись команд видеоконтроллера и чтение регистра состояния
D800H	КР580ВИ53 Таймер	Загрузка/чтение счетчика 0
D801H		Загрузка/чтение счетчика 1
D802H		Загрузка/чтение счетчика 2
D803H		Запись команды режима для таймера
F800H	КР580ВТ57	Запись стартового адреса канала 0 КПДП
F801H		Запись количества пересылок и направления передачи канала 0 КПДП
F802H		Запись стартового адреса канала 1 КПДП
F803H		Запись количества пересылок и направления передачи канала 1 КПДП
F804H		Запись стартового адреса канала 2 КПДП
F805H		Запись количества пересылок и направления передачи канала 2 КПДП
F806H		Запись стартового адреса канала 3 КПДП
F807H		Запись количества пересылок и направления передачи канала 3 КПДП
F808H		Запись режима работы КПДП

ТАБЛИЦА 1.3

Вывод ППА	Контакт разъема	Вывод ППА	Контакт разъема
Порт A0	C3	Порт B0	B10
Порт A1	C2	Порт B1	A9
Порт A2	B2	Порт B2	A10
Порт A3	A1	Порт B3	C6
Порт A4	B1	Порт B4	B6
Порт A5	C1	Порт B5	C5
Порт A6	B3	Порт B6	B5
Порт A7	B4	Порт B7	C4
Порт C0	A8	Порт C4	C8
Порт C1	--	Порт C5	B8
Порт C2	C10	Порт C6	C7
Порт C3	B9	Порт C7	B7
Общий	A5	Общий	A4

ТАБЛИЦА 1.4

Принтер	Сигнал	МИКРОША	Порт	Назначение
1	STROBE\	C8	C4	Сигнал активен в 0. Сопровождает выдачу данных принтеру.
2	DATA1	C3	A0	Данные, разряд 0
3	DATA2	C2	A1	Данные, разряд 1
4	DATA3	B2	A2	Данные, разряд 2
5	DATA4	A1	A3	Данные, разряд 3
6	DATA5	B1	A4	Данные, разряд 4
7	DATA6	C1	A5	Данные, разряд 5
8	DATA7	B3	A6	Данные, разряд 6
9	DATA8	B4	A7	Данные, разряд 7
11	BUSY	C10	C2	Сигнал, информирующий компьютер о том, что принтер не готов принимать данные.
19-30	GND	A5		Общий провод

ТАБЛИЦА 1.5

МЕТКА	МНЕМОНИКА	ОПЕРАНДЫ	КОММЕНТАРИЙ
PORTA	EQU	0C800H	Порт A ППА интерфейса
PORTC	EQU	0C802H	Порт C ППА интерфейса
RUS	EQU	0C803H	Регистр управляющего слова ППА интерфейса
PRINTER:	PUSH	PSW	сохраним регистр A
WAIT:	LDA	PORTC	проверка сигнала BUSY
	ANI	4	если не готов, то переход на опрос
	JNZ	WAIT	
	MOV	A,C	A=коду символа
	STA	PORTA	вывод в порт A
	MVI	A,8	команда сброс бита C4
	STA	RUS	сигнал STROBE=0
	MVI	A,9	команда установки бита C4
	STA	RUS	сигнал STROBE=1
	POP	PSW	восстановим регистр A
	RET		возврат из подпрограммы

1.4 Системный разъем ПК и его использование

В ПК МИКРОША предусмотрена возможность подключения дополнительных модулей при помощи системного разъема. Эти модули могут выполнять разнообразные функции, например:

- увеличение объема оперативной памяти ПК (модуль ОЗУ);
- увеличение объема постоянной памяти ПК (модуль ПЗУ);
- подключение накопителя на гибких магнитных дисках (контроллер НГМД);
- программирование ППЗУ (программатор ППЗУ);
- создание диска на микросхемах ОЗУ и ПЗУ (электронный диск);
- обеспечение связи с удаленными ЭВМ (связной контроллер);
- прием и выдача аналоговых сигналов (модуль ЦАП-АЦП).

На системном разъеме имеется достаточный набор сигналов для подключения как перечисленных модулей, так и многих других. В таблице 1.6 приведено описание сигналов на этом разъеме. Если после имени сигнала стоит косая черта, то это означает, что для данного сигнала активный уровень - уровень логического нуля.

ТАБЛИЦА 1.6

Сигнал	Контакт	Назначение
A0	Б8	Шина адреса A0
A1	Б9	Шина адреса A1
A2	Б10	Шина адреса A2
A3	Б12	Шина адреса A3
A4	Б13	Шина адреса A4
A5	Б14	Шина адреса A5
A6	Б15	Шина адреса A6
A7	Б16	Шина адреса A7
A8	Б17	Шина адреса A8
A9	Б18	Шина адреса A9
A10	Б24	Шина адреса A10
A11	Б23	Шина адреса A11
A12	Б20	Шина адреса A12
A13	Б21	Шина адреса A13
A14	Б22	Шина адреса A14
A15	Б19	Шина адреса A15
D0	A15	Шина данных D0
D1	A16	Шина данных D1
D2	A17	Шина данных D2
D3	A18	Шина данных D3
D4	A22	Шина данных D4
D5	A2	Шина данных D5
D6	A20	Шина данных D6
D7	A19	Шина данных D7
RD\	A7	Сигнал чтения из памяти или устройств
WR\	Б6	Сигнал записи в память или устройства
RESET	A12	Сигнал сброса. Активен примерно 1 мс после нажатия кнопки "СБРОС"
INT	A3	Запрос прерывания. Для использования требуется установка перемычки на плате.
READY	Б28	Сигнал готовности. Для использования требуется установка перемычки на плате.
SYNC	A5	Сигнал начала машинного цикла.
OSC	Б27	Сигнал с выхода тактового генератора с частотой 16,3 МГц.
Ф2ТТЛ	A24	Сигнал с частотой 1,77 МГц.
32K\	Б4	Сигнал с выхода дешифратора. Активен при обращении по адресам 8000H-BFFFH.
CS1\	Б26	Сигнал с выхода дешифратора. Активен при обращении по адресам E000H-E7FFH.
CS2\	A25	Сигнал с выхода дешифратора. Активен при обращении по адресам E800H-EFFFH.
CS3\	Б25	Сигнал с выхода дешифратора. Активен при обращении по адресам F000H-F7FFH.
CS4\	A4	Сигнал с выхода дешифратора. Активен при обращении по адресам F800H-FFFFH.
OUT0	A8	Сигнал с выхода счетчика 0 таймера.
RAS\	A10	Сигнал регенерации содержимого динамического ОЗУ.
CAS2\	A9	Сигнал выборки динамического ОЗУ, расположенного по адресам 8000H-BFFFH.
+5В	Б29, Б30 Б1	Питание (ток < 300 мА).
-5В	A26	Питание (ток < 50 мА).
Общий	A1, A2 A29, A30	Общий провод питания

Нагрузочная способность на линиях адреса, данных и управления не превышает одного ТТЛ входа, поэтому при самостоятельном конструировании дополнительных модулей следует предусмотреть использование шинных формирователей. Кроме этого, следует учесть, что блок питания ПК практически не имеет запаса по мощности. Для энергоемких модулей требуется внешний источник питания.

2. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ПК МИКРОША

Опыт использования персональных компьютеров однозначно показывает, что, по-видимому, самой важной их характеристикой следует считать разнообразие и доступность программного обеспечения. Именно это обстоятельство обеспечило широчайшее использование во всем мире таких ПК, как APPLE, IBM PC, SPECTRUM.

Хотя у этих ПК имелись достойные конкуренты, обладающие более высокими техническими характеристиками, они, однако, проиграли соревнование из-за меньшго разнообразия их программного обеспечения.

ПК МИКРОША появился одним из первых в СССР. В настоящее время выпускается несколько других моделей, имеющих с ним программную совместимость. Эти причины обусловили достаточно широкий спектр программ для МИКРОШИ. Далее описаны программные средства, в основном разработанные автором.

2.1. Организация программного обеспечения

Программное обеспечение ПК МИКРОША строится по иерархическому принципу. На нижнем уровне иерархии находится программа "Монитор". При помощи этой программы производится загрузка с магнитной ленты в ОЗУ и запуск в работу всех других программ. Эти программы для ввода вывода информации используют ряд стандартных подпрограмм, входящих в состав "Монитора".

Отсутствие в составе ПК устройств внешней памяти с произвольным доступом (магнитные диски) накладывает

существенные ограничения на состав и возможности программного обеспечения ПК. Как правило, все используемые программы для ПК МИКРОША имеют объем меньше 29 Кбайт. Это обстоятельство затрудняет создание программных комплексов и программных средств работы с большими массивами данных.

Если ПК снабжен блоком расширения, содержащим ПЗУ объемом 16 Кбайт, то все рассматриваемые далее программы могут находиться в этом ПЗУ. В этом случае сразу после включения питания в распоряжении оператора имеются редактор текста, интерпретатор языка Бейсик и графический редактор.

Рассмотрим теперь ряд программ, входящих в состав ПО персонального компьютера МИКРОША.

2.2 Резидентный монитор

Монитор является основной управляющей программой ПК МИКРОША. Она расположена в ПЗУ и занимает старшие 2 Кбайт в адресном пространстве. Монитор предназначен для загрузки и запуска в работу программ с магнитной ленты и отладки программ, написанных в машинных кодах. Кроме этого, монитор содержит ряд стандартных подпрограмм ввода-вывода, к которым можно обращаться из прикладных программ. Таким образом, монитор позволяет освободить программиста от написания программ физического ввода-вывода.

После запуска в работу монитор производит настройку всех периферийных БИС в требуемый режим работы и выдает на экран приглашение к вводу директив в следующем виде:

```
-МИКРОША-
->
```

Все директивы состоят из одной прописной буквы латинского алфавита и содержат до трех параметров. Параметры отделяются друг от друга запятой. Параметры набираются непосредственно после имени директивы. Для ввода параметров используется шестнадцатиричная система счисления.

Все допустимые директивы можно разделить на три группы: директивы работы с памятью, директивы ввода-вывода и директивы запуска и отладки программ. Имеющиеся директивы представлены в таблице 2.1.

ТАБЛИЦА 2.1

	ДИРЕКТИВЫ МОНИТОРА
	ДИРЕКТИВЫ РАБОТЫ С ПАМЯТЬЮ
DA1, A2	РАСПЕЧАТКА СОДЕРЖИМОГО ОБЛАСТИ ПАМЯТИ В ВИДЕ ТАБЛИЦЫ ШЕСТНАДЦАТИРИЧНЫХ КОДОВ. A1-НАЧАЛЬНЫЙ АДРЕС, A2-КОНЕЧНЫЙ АДРЕС.
LA1, A2	РАСПЕЧАТКА СОДЕРЖИМОГО ПАМЯТИ В ВИДЕ ТАБЛИЦЫ СИМВОЛОВ. A1-НАЧАЛЬНЫЙ АДРЕС, A2-КОНЕЧНЫЙ АДРЕС
FA1, A2, D8	ЗАПОЛНЕНИЕ ОБЛАСТИ ПАМЯТИ ОТ АДРЕСА A1 ДО АДРЕСА A2 ВОСЬМИРАЗРЯДНЫМ КОДОМ D8
MA	ПРОСМОТР И МОДИФИКАЦИЯ СОДЕРЖИМОГО ПАМЯТИ, НАЧИНАЯ С АДРЕСА A
TA1, A2, A3	ПЕРЕСЫЛКА СОДЕРЖИМОГО ОБЛАСТИ ПАМЯТИ С АДРЕСАМИ ОТ A1 ДО A2 В ОБЛАСТЬ С НАЧАЛЬНЫМ АДРЕСОМ A3
CA1, A2, A3	СРАВНЕНИЕ СОДЕРЖИМОГО ОБЛАСТИ ПАМЯТИ С АДРЕСАМИ ОТ A1 ДО A2 С ОБЛАСТЬЮ ПАМЯТИ С НАЧАЛЬНЫМ АДРЕСОМ A3. ЕСЛИ ЕСТЬ РАЗЛИЧИЯ В СОДЕРЖИМОМ, ТО ОНИ ВЫВОДЯТСЯ НА ЭКРАН.
SA1, A2, D8	ПОИСК В ОБЛАСТИ ПАМЯТИ С АДРЕСАМИ ОТ A1 ДО A2 ВСЕХ АДРЕСОВ, ПО КОТОРЫМ ХРАНИТСЯ БАЙТ D8
	ДИРЕКТИВЫ ЗАПУСКА И ОТЛАДКИ ПРОГРАММ
GA1, A2	ЗАПУСК ПРОГРАММЫ С АДРЕСА A1. ЕСЛИ УКАЗАН АДРЕС A2, ТО ПРИ ЕГО ДОСТИЖЕНИИ УПРАВЛЕНИЕ ПЕРЕДАЕТСЯ ПРОГРАММЕ МОНИТОР.
X	ПРОСМОТР И МОДИФИКАЦИЯ СОДЕРЖИМОГО ВНУТРЕННИХ РЕГИСТРОВ МИКРОПРОЦЕССОРА.
	ДИРЕКТИВЫ ВВОДА-ВЫВОДА
I	ВВОД ИНФОРМАЦИИ С МАГНИТНОЙ ЛЕНТЫ В ОЗУ
OA1, A2	ВЫВОД СОДЕРЖИМОГО ОБЛАСТИ ПАМЯТИ С АДРЕСАМИ ОТ A1 ДО A2 НА МАГНИТОФОН

Если директива набрана с ошибкой, то монитор не выполняет ее и выдает на экран знак вопроса. При помощи клавиши "<-" можно редактировать набираемую директиву. Директива начинает выполняться только после нажатия клавиши "ВК".

В таблице 2.2 приведены стандартные подпрограммы монитора. Как правило, передача параметров к подпро-

граммам и возвращение результата из подпрограмм производится во внутренних регистрах микропроцессора.

ТАБЛИЦА 2.2

НАЗНАЧЕНИЕ	АДРЕС ВЫЗОВА	ПАРАМЕТРЫ
ВВОД СИМВОЛА С КЛАВИАТУРЫ	0F803H	ВХОДНЫЕ: - ВЫХОДНЫЕ: A - ВВЕДЕННЫЙ КОД
ВВОД БАЙТА С МАГНИТОФОНА	0F806H	ВХОДНЫЕ: A=FFH - С ПОИСКОМ СИНХРОБАЙТА A=8 - БЕЗ ПОИСКА СИНХРОБАЙТА ВЫХОДНЫЕ: A - ВВЕДЕННЫЙ КОД
ВЫВОД СИМВОЛА НА ЭКРАН	0F809H	ВХОДНЫЕ: C - ВЫВОДИМЫЙ КОД ВЫХОДНЫЕ: -
ЗАПИСЬ БАЙТА НА МАГНИТОФОН	0F80CH	ВХОДНЫЕ: C - ВЫВОДИМЫЙ БАЙТ ВЫХОДНЫЕ: -
ОПРОС СОСТОЯНИЯ КЛАВИАТУРЫ	0F812H	ВХОДНЫЕ: - ВЫХОДНЫЕ: A=00 - НЕ НАЖАТА A=FFH - НАЖАТА
РАСПЕЧАТКА БАЙТА НА ЭКРАНЕ В 16-РИЧНОМ ВИДЕ	0F815H	ВХОДНЫЕ: A - ВЫВОДИМЫЙ ВЫХОДНЫЕ: -
ВЫВОД СООБЩЕНИЙ НА ЭКРАН	0F818H	ВХОДНЫЕ: HL - АДРЕС НАЧАЛА ВЫХОДНЫЕ: -
ВВОД КОДА НАЖАТОЙ КЛАВИШИ БЕЗ ОЖИДАНИЯ	0FEEAH	ВХОДНЫЕ: - ВЫХОДНЫЕ: A=FFH - НЕ НАЖАТА ИНАЧЕ - КОД КЛАВИШИ

Стандартные подпрограммы не изменяют содержимого других внутренних регистров микропроцессора. Для обращения к стандартной подпрограмме из прикладной программы достаточно подготовить необходимые параметры во внутренних регистрах микропроцессора и поместить в программу команду вызова подпрограммы, расположенной по адресу, указанному в таблице 2.2.

Использование стандартных подпрограмм позволяет обеспечить переносимость ПО.

2.3 Монитор-загрузчик

ПК МИКРОША имеет очень много общего с компьтером РАДИО-86РК. Схемотехника основных узлов практически совпадает. Однако, имеется ряд отличий, которые не позволяют производить обмен программами между этими двумя компьютерами. Отличия эти заключаются в разных адресах устройств, в форматах записи на магнитную ленту и отсутствии в резидентном мониторе ПК МИКРОША ряда подпрограмм ввода-вывода, имеющихся в компьютере РАДИО-86РК.

Монитор-загрузчик предназначен для обеспечения обмена программами между МИКРОШЕЙ и РАДИО-86РК. В таблице 2.3 приведены его шестнадцатиричные коды. Монитор-загрузчик располагается в оперативной памяти по адресам 5800H-5FFFH. После загрузки в оперативную память и запуска в работу с адреса 5800H монитор выдает стандартное приглашение к вводу директив. Набор директив и их формат полностью соответствует директивам монитора РАДИО-86РК. С помощью монитора-загрузчика можно вводить в память программы с магнитной ленты, записанные в формате РАДИО-86РК, а затем, запустив резидентный монитор, выводить программы на ленту в формате МИКРОШИ. Естественно, возможен и обратный процесс.

Другое назначение монитора-загрузчика - предоставление в распоряжение программиста более широкого набора подпрограмм ввода-вывода. Обращение к ним должно происходить с учетом того, что начальный адрес таблицы переходов равен 5800H, а не F800H. Таким образом, при адаптации программ от РАДИО-86РК необходимо найти в них обращения к подпрограммам резидентного монитора и заменить их на обращения к мониторузагрузчику.

ТАБЛИЦА 2.3

```
5800: C3 36 58 C3 B5 5E C3 E8 5B C3 0A 5D C3 96 5C C3
5810: 0A 5D C3 53 5E C3 F5 5C C3 2E 59 C3 CF 5E C3 D0
5820: 5A C3 D4 5A C3 06 7B C3 99 5B C3 66 5B C3 1E 5B
5830: C3 A5 5F C3 A9 5F 3E 98 32 03 C0 3E 99 32 03 C8
5840: AF 32 01 C8 3E B6 32 03 D8 21 02 D8 36 05 36 05
5850: 31 CF 76 CD 1E 5B 21 00 76 11 5F 76 0E 00 CD F9
5860: 59 21 CF 76 22 1C 76 21 AD 5F CD 2E 59 CD 1E 5B
```

```
5870: 21 FF 75 22 31 76 21 22 1D 22 2F 76 3E C3 32 26
5880: 76 31 CF 76 21 BC 5F CD 2E 59 32 02 C0 CD FA 58
5890: 21 81 58 E5 21 33 76 7E FE 58 CA A5 5A F5 CD 38
58A0: 59 2A 2B 76 4D 44 2A 29 76 EB 2A 27 76 F1 FE 44
58B0: CA D1 59 FE 43 CA E3 59 FE 46 CA F9 59 FE 53 CA
58C0: 00 5A FE 54 CA 0B 5A FE 4D CA 32 5A FE 47 CA 4B
58D0: 5A FE 49 CA DB 5A FE 4F CA 7D 5B FE 4C CA 14 5A
58E0: 3E 3F CD 09 5D C3 81 58 3E 33 BD CA FD 58 E5 21
58F0: C9 5F CD 2E 59 E1 2B C3 FF 58 21 33 76 06 00 CD
5900: B5 5E FE 08 CA E8 58 FE 7F CA E8 58 C4 09 5D 77
5910: FE 0D CA 26 59 FE 2E CA 81 58 06 FF 3E 52 BD CA
5920: E0 58 23 C3 FF 58 78 17 11 33 76 06 00 C9 7E A7
5930: C8 CD 09 5D 23 C3 2E 59 21 27 76 11 2D 76 0E 00
5940: CD F9 59 11 34 76 CD 66 59 22 27 76 22 29 76 D8
5950: 3E FF 32 2D 76 CD 66 59 22 29 76 D8 CD 66 59 22
5960: 2B 76 D8 C3 E0 58 21 00 00 1A 13 FE 0D CA 9A 59
5970: FE 2C C8 FE 20 CA 69 59 D6 30 FA E0 58 FE 0A FA
5980: 8E 59 FE 11 FA E0 58 FE 17 F2 E0 58 D6 07 4F 29
5990: 29 29 29 DA E0 58 09 C3 69 59 37 C9 7C BA C0 7D
59A0: BB C9 CD B0 59 CD 9C 59 C2 AE 59 33 33 C9 23 C9
59B0: CD CF 5E FE 03 C0 CD 1E 5B C3 E0 58 E5 21 C2 5F
59C0: CD 2E 59 E1 C9 7E C5 CD F5 5C 3E 20 CD 09 5D C1
59D0: C9 CD C8 5B CD C5 59 CD A2 59 7D E6 0F CA E1 59
59E0: C3 D4 59 0A BE CA F2 59 CD C8 5B CD C5 59 0A CD
59F0: C6 59 03 CD A2 59 C3 E3 59 71 CD A5 59 C3 F9 59
5A00: 79 BE CC C8 5B CD A2 59 C3 00 5A 7E 02 03 CD A5
5A10: 59 C3 0B 5A CD C8 5B 7E B7 FA 21 5A FE 20 D2 23
5A20: 5A 3E 2E CD 09 5D CD A2 59 7D E6 0F CA 14 5A C3
5A30: 17 5A CD C8 5B CD C5 59 E5 CD FA 58 E1 D2 47 5A
5A40: E5 CD 66 59 7D E1 77 23 C3 32 5A CG 9C 59 CA 66
5A50: 5A EB 22 23 76 7E 32 25 76 36 F7 3E C3 32 30 00
5A60: 21 74 5A 22 31 00 31 18 76 C1 D1 E1 F1 F9 2A 16
5A70: 76 C3 26 76 22 16 76 F5 E1 22 1E 76 E1 2B 22 14
5A80: 76 21 00 00 39 31 1E 76 E5 D5 C5 2A 14 76 31 CF
5A90: 76 CD C8 5B EB 2A 23 76 CD 9C 59 C2 81 58 3A 25
5AA0: 76 77 C3 81 58 21 CD 5F CD 2E 59 21 14 76 06 06
5AB0: 5E 23 56 C5 E5 EB CD C8 5B CD FA 58 D2 C8 5A CD
5AC0: 66 59 D1 D5 EB 72 2B 73 E1 C1 05 23 C2 B0 5A C9
5AD0: 2A 02 76 C9 E5 2A 00 76 7E E1 C9 3A 2D 76 B7 CA
5AE0: E6 5A 7B 32 2F 76 CD 06 5B CD C8 5B EB CD C8 5B
5AF0: EB C5 CD 66 5B 60 69 CD C8 5B D1 CD 9C 59 C8 EB
5B00: CD C8 5B C3 E0 58 3E FF CD 4F 5B E5 09 EB CD 4D
5B10: 5B E1 09 EB E5 CD 5A 5B 3E FF CD 4F 5B E1 E5 21
5B20: 01 D0 36 00 2B 36 4D 36 1D 36 99 36 93 23 36 27
5B30: 7E 7E E6 20 CA 31 5B 21 08 F8 36 80 2E 04 36 D0
5B40: 36 76 2C 36 23 36 49 2E 08 36 A4 E1 C9 3E 08 CD
5B50: E8 5B 47 3E 08 CD E8 5B 4F C9 3E 08 CD E8 5B 77
5B60: CD A5 59 C3 5A 5B 01 00 00 7E 81 4F F5 CD 9C 59
5B70: CA AB 59 F1 78 8E 47 CD A5 59 C3 69 5B 79 B7 CA
5B80: 85 5B 32 30 76 E5 CD 66 5B E1 CD C8 5B EB CD C8
5B90: 5B EB E5 60 69 CD C8 5B E1 C5 01 00 00 CD 96 5C
5BA0: 05 E3 E3 C2 9D 5B 0E E6 CD 96 5C CD E0 5B EB CD
5BB0: E0 5B EB CD D6 5B 21 00 00 CD E0 5B 0E E6 CD 96
5BC0: 5C E1 CD E0 5B C3 1E 5B C5 CD BC 59 7C CD F5 5C
5BD0: 7D CD C6 59 C1 C9 4E CD 96 5C CD A5 59 C3 D6 5B
5BE0: 4C CD 96 5C 4D C3 96 5C E5 C5 D5 57 3E 80 32 08
5BF0: F8 21 00 00 39 31 00 00 22 0D 76 0E 00 3A 02 C0
5C00: 0F 0F 0F 0F E6 01 5F F1 79 E6 7F 07 4F 26 00 25
5C10: CA 84 5C F1 3A 02 C0 0F 0F 0F E6 01 BB CA 0F
5C20: 5C B1 4F 15 3A 2F 76 C2 2C 5C D6 12 47 F1 05 C2
5C30: 2D 5C 14 3A 02 C0 0F 0F 0F 0F E6 01 5F 7A B7 F2
5C40: 5B 5C 79 FE E6 C2 4F 5C AF 32 2E 76 C3 59 5C FE
5C50: 19 C2 07 5C 3E FF 32 2E 76 16 09 15 C2 07 5C 21
5C60: 04 F8 36 D0 36 76 23 36 23 36 49 3E 27 32 01 D0
5C70: 3E E0 32 01 D0 2E 08 36 A4 2A 0D 76 F9 3A 2E 76
5C80: A9 C3 F1 5C 2A 0D 76 F9 CD 1E 5B 7A B7 F2 E0 58
5C90: CD B0 59 C3 EC 5B E5 C5 D5 F5 3E 80 32 08 F8 21
5CA0: 00 00 39 31 00 00 16 08 F1 79 07 4F 3E 01 A9 32
5CB0: 02 C0 3A 30 76 47 F1 05 C2 B6 5C 3E 00 A9 32 02
5CC0: C0 15 3A 30 76 C2 CA 5C D6 0E 47 F1 05 C2 CB 5C
5CD0: 14 15 C2 A8 5C F9 21 04 F8 36 D0 36 76 23 36 23
5CE0: 36 49 3E 27 32 01 D0 3E E0 32 01 D0 2E 08 36 A4
```



```
5CF0: F1 D1 C1 E1 C9 F5 0F 0F 0F 0F CD FE 5C F1 E6 0F
5D00: FE 0A FA 07 5D C6 07 C6 30 4F F5 C5 D5 E5 21 D7
5D10: 5D E5 2A 02 76 EB 2A 00 76 3A 04 76 3D FA 3B 5D
5D20: CA B7 5D E2 C5 5D 79 D6 20 4F 0D FA 36 5D C5 CD
5D30: 0B 5E C1 C3 2A 5D AF 32 04 76 C9 79 E6 7F FE 1F
5D40: CA F5 5D FE 0C CA 04 5E FE 0D CA 45 5E FE 0A CA
5D50: 99 5D FE 08 CA 28 5E FE 18 CA 0B 5E FE 19 CA 34
5D60: 5E FE 1A CA 17 5E FE 1B CA F0 5D FE 07 C2 8A 5D
5D70: 01 F0 1F 3A 02 C0 F6 06 32 02 C0 0B 78 B1 C2 7B
5D80: 5D 3A 02 C0 E6 F9 32 02 C0 C9 77 CD 0B 5E 7A FE
5D90: 03 C0 7B FE 08 C0 CD 34 5E 7A FE 1B C2 17 5E E5
5DA0: D5 21 C2 77 11 10 78 01 9E 07 1A 77 23 13 0B 79
5DB0: B0 C2 AA 5D D1 E1 C9 79 FE 59 C2 36 5D CD 04 5E
5DC0: 3E 02 C3 37 5D 79 D6 20 4F 0D 3E 04 FA 37 5D C5
5DD0: CD 17 5E C1 C3 C9 5D 22 00 76 EB 22 02 76 3E 80
5DE0: 32 01 D0 7D 32 00 D0 7C 32 00 D0 E1 D1 C1 F1 C9
5DF0: 3E 01 C3 37 5D 21 F4 7F 11 25 09 AF 77 2B 1B 7B
5E00: B2 C2 FB 5D 11 08 03 21 C2 77 C9 7B 23 1C FE 47
5E10: C0 1E 08 01 C0 FF 09 7A FE 1B 01 4E 00 C2 25 5E
5E20: 16 02 01 B0 F8 14 09 C9 7B 2B 1D FE 08 C0 1E 47
5E30: 01 40 00 09 7A EE 03 01 B2 FF C2 42 5E 16 1C 01
5E40: 50 07 15 09 C9 7D 93 D2 4B 5E 25 6G 1E 08 01 08
5E50: 00 09 C9 3A 02 C0 E6 20 CA 60 5E 3A 05 76 B7 C0
5E60: E5 2A 09 76 CD CF 5E BD 6F CA 7C 5E 3E 01 32 0B
5E70: 76 26 15 AF 22 09 76 E1 32 05 76 C9 25 C2 73 5E
5E80: 3C CA 74 5E 3C CA A3 5E C5 01 03 00 CD 73 5D C1
5E90: 3A 0B 76 26 C0 3D 32 0B 76 CA 9E 5E 26 28 3E FF
5EA0: C3 74 5E 3A 02 C0 E6 20 CA A3 5E 3A 06 76 2F 32
5EB0: 06 76 C3 6C 5E CD 53 5E B7 CA B5 5E AF 32 05 76
5EC0: 3A 09 76 B7 C0 3A 01 C8 2F 32 01 C8 C3 B5 5E 3A
5ED0: 02 C0 E6 20 C2 DA 5E 3E FE C9 AF 32 01 C0 32 02
5EE0: C0 3A 06 76 E6 01 F6 06 32 03 C0 3A 00 C0 3C C2
5EF0: F4 5E 3D C9 E5 21 01 07 7D 0F 6F 2F 32 01 C0 3A
5F00: 00 C0 2F B7 C2 0F 5F 25 F2 F8 5E 3E FF E1 C9 2E
5F10: 20 3A 00 C0 2F B7 CA 0B 5F 2D C2 11 5F 2E 08 2D
5F20: 07 D2 1F 5F 7D FE 01 CA 4D 5F DA 46 5F 07 07 07
5F30: C6 20 B4 C3 59 5F 19 1A 0C 00 01 02 03 04 20 1B
5F40: 09 0A 0D 1F 08 18 7C 21 3E 5F C3 51 5F 7C 21 36
5F50: 5F 85 6F 7E FE 40 E1 D8 E5 6F 3A 02 C0 67 E6 40
5F60: C2 6D 5F 7D FE 40 FA 92 5F E6 1F E1 C9 3A 06 76
5F70: B7 CA 7D 5F 7D FE 40 FA 7D 5F F6 20 6F 7C E6 80
5F80: C2 92 5F 7D FE 40 FA 8E 5F 7D EE 20 E1 C9 7D E6
5F90: 2F 6F 7D FE 40 E1 F0 E5 6F E6 0F FE 0C 7D FA A3
5FA0: 5F EE 10 E1 C9 2A 31 76 C9 22 31 76 C9 1F 0D 0A
5FB0: 2D 20 6D 69 6B 72 6F 7B 61 20 2D 00 0D 0A 2D 2D
5FC0: 3E 00 0D 0A 18 18 18 18 00 08 20 08 00 0D 0A 20
5FD0: 50 43 2D 0D 0A 20 48 4C 2D 0D 0A 20 42 43 2D 0D
5FE0: 0A 20 44 45 2D 0D 0A 20 41 46 2D 0D 0A 20 53 50
5FF0: 2D 19 19 19 19 19 19 00 F5 C5 CD 68 60 CD 03 60
```

2.4. Интерпретатор языка Бейсик

Поставляемый вместе с ПК интерпретатор Бейсика имеет ряд ограничений, что существенно затрудняет разработку программ. В данной главе описывается интерпретатор Бейсика "Микрон", адаптированный для использования в МИКРОШе.

Интерпретатор хранится на магнитной ленте и загружается в ОЗУ при помощи директивы монитора "I". Интерпретатор занимает 8 Кбайт ОЗУ, начиная с адреса 0000H.

Для ввода-вывода информации интерпретатор использует подпрограммы монитора.

Запускают интерпретатор в работу по директиве монитора G0. На экране дисплея при этом появляется сообщение:

BASIC

NEW?

Если в ответ на этот вопрос нажать клавишу "Y", то интерпретатор сотрет имеющуюся в его буфере программу. В противном случае текст программы будет сохранен. После этого выдается сообщение:

ЖДУ:

Это означает, что интерпретатор готов к приему директив.

Интерпретатор имеет два основных режима работы: режим непосредственной интерпретации и режим интерпретации программы, хранящейся в памяти.

Программа на Бейсике состоит из последовательности пронумерованных строк. Строкам принято присваивать номера с интервалом, равным 10. Номера строк могут быть любыми - от 0 до 65529. Каждая строка программы может состоять из одного или нескольких операторов. Если операторов несколько, то они отделяются друг от друга символом ":" - двоеточие.

В Бейсике существует два типа констант: числовые и символьные. Числовые константы - это любые десятичные числа в диапазоне от -10 до +10 , символьные - последовательность любых отображаемых символов, заключенная в кавычки. Точность задания констант - 6 десятичных цифр. Если перед константой помещен знак "&", то это означает, что константа указана в шестнадцатиричной системе счисления.

Соответственно типам переменных в программах на Бейсике используются переменные также двух типов: числовые и символьные. Обращение к переменным производится по имени. Имя переменной состоит из одного или двух символов, первый из которых обязательно должен быть буквой латинского алфавита, а второй - буквой латинского алфавита или цифрой. Символьные переменные после имени содержат знак "$".

Группе переменных одного типа может быть присвоено общее имя. Такие переменные называют переменными с индексами или массивами переменных.

Для обращения к каждой отдельной переменной в массиве используют один или несколько индексов. Наименьшее значение индекса равно 0, а наибольшее определяется размером массива. Если индекс один, то говорят, что массив одномерный, два - двухмерный и т.д. Имена массивов подчиняются тем же правилам, что и имена переменных. Индексы необходимо указывать в круглых скобках после имени массива. Разрешено использование как массивов числовых переменных, так и массивов символьных переменных.

Переменные и константы служат операндами в выражениях языка Бейсик. Кроме них в выражения входят также знаки операций, скобки и имена функций. Обращение к функциям происходит по их имени, аргумент при этом указывают в круглых скобках после имени.

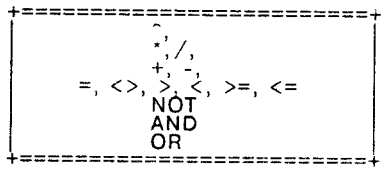
Все выражения можно разделить на 4 типа: арифметические, символьные, выражения отношения и логические. В таблице 2.4 приведены знаки операций, принятые в Бейсике. Числовые переменные и константы могут принимать участие в выражениях любого типа. Для символьных переменных и констант разрешены только операции отношения и конкатенации (слияния). Последняя операция обозначается знаком "+".

При вычислении результата выражения все операции производятся в определенном порядке. Таблица 2.5 показывает приоритет операций, определяющий порядок вычислений. Чем выше в таблице находится знак той или иной операции, тем выше ее приоритет.

ТАБЛИЦА 2.4

ЗНАК ОПЕРАЦИИ	ОПЕРАЦИЯ
	АРИФМЕТИЧЕСКИЕ ОПЕРАЦИИ
+	СЛОЖЕНИЕ
-	ВЫЧИТАНИЕ
*	УМНОЖЕНИЕ
/	ДЕЛЕНИЕ
^	ВОЗВЕДЕНИЕ В СТЕПЕНЬ
	ОПЕРАЦИИ ОТНОШЕНИЯ
>	БОЛЬШЕ
<	МЕНЬШЕ
=	РАВНО
<>	НЕ РАВНО
>=	БОЛЬШЕ ИЛИ РАВНО
<=	МЕНЬШЕ ИЛИ РАВНО
	ЛОГИЧЕСКИЕ ОПЕРАЦИИ
NOT	ОТРИЦАНИЕ ("НЕ")
AND	ЛОГИЧЕСКОЕ УМНОЖЕНИЕ ("И")
OR	ЛОГИЧЕСКОЕ СЛОЖЕНИЕ ("ИЛИ")

ТАБЛИЦА 2.5



Директивы Бейсика

Для ввода текста новой программы необходимо сначала задать ему соответствующий режим работы - директивой NEW, а затем с клавиатуры набрать программу. Вводимую строку можно сразу редактировать, вставляя пропущенный символ перед символом, под которым расположен курсор. Символы, расположенные справа от курсора, можно стирать при помощи клавиши F2. Ввод каждой строки заканчивают нажатием на клавишу "ВК", причем курсор может располагаться в любом месте строки. Если при вводе строки нажать клавишу "АР2" и затем какую-либо другую клавишу, то интерпретатор автоматически введёт в строку соответствующее служебное слово. В таблице 2.6 показано соответствие служебных слов и клавиш клавиатуры.

Просмотреть текст программы можно при помощи директивы **LIST N1,N2**. Параметры директивы задают диапазон строк, выводимых на экран. Если необходимо удалить какую-либо строку программы, достаточно просто набрать ее номер и нажать клавишу "ВК", а чтобы вставить строку в середину программы нужно набрать любой номер строки, попадающий в интервал между двумя соседними, затем саму строку и нажать клавишу "ВК". Если надо удалить несколько строк, то следует воспользоваться директивой **DELETE N1,N2**. Отредактировать строку можно при помощи оператора EDIT N, где N - номер строки.

При вводе текста программы удобно использовать директиву **AUTO N1,N2**. В этом случае интерпретатор автоматически нумерует строки программы, начиная с номера N1 с шагом N2.

ТАБЛИЦА 2.6

Латинский алфавит:		Русский алфавит:			
A	CLS	А	FN	Ш	CHR$
B	FOR	Б	THEN	Э	LEFT$
C	NEXT	Ц	NOT	Щ	RIGHT$
D	DATA	Д	STEP	Ч	MID$
E	INPUT	Е	AND		
F	DIM	Ф	OR	Спец. символы:	
G	READ	Г	SGN	[	CLOAD
H	CUR	Х	INT	\	CSAVE
I	GOTO	И	ABS	]	NEW
J	RUN	Й	USR		TAB
K	IF	К	FRE		TO
L	RESTORE	Л	INP	@	SPC
M	GOSUB	М	POS		SCREEN$
N	RETURN	Н	SQR	*	INKEY$
O	REM	О	RND	+	AT
P	STOP	П	LOG	-	BEEP
Q	OUT	Я	EXP	.	PAUSE
R	ON	Р	COS	/	VERIFY
S	PLOT	С	SIN	:	RENUM
T	LINE	Т	TAN	;	ACS
U	POKE	У	ATN	<	LG
V	PRINT	Ж	PEEK	=	LPRINT
W	DEF	В	LEN	>	LLIST
X	CONT	Ь	STR$		
Y	LIST	Ы	VAL		
Z	CLEAR	З	ASC		
			Цифры		
0	HOME	1	EDIT	2	DELETE
3	MERGE	4	AUTO	5	HIMEM
6	@	7	ASN	8	ADDR
9	PI				

Перенумерация строк программы проводится при помощи директивы **RENUM N1,N2**. Всем строкам программы будут присвоены новые номера, начиная с N1 с шагом N2. Если в этих директивах какой либо параметр не указан, то по умолчанию его значение равно 10.

Отладка программы на Бейсике производится также при помощи интерпретатора. Для этой цели использована возможность перевода интерпретатора из программного режима в непосредственный при выполнении оператора STOP или одновременном нажатии на клавиши "УС" и "Е".

Для выхода из режимов EDIT, AUTO, LIST, INPUT служит клавиша F4. В непосредственном режиме можно просмотреть и при необходимости модифицировать (изменить) значения переменных, снова просмотреть текст программы по директиве LIST и затем продолжить выполнение программы с того места, где она была прервана.

Рассмотрим теперь другие директивы интерпретатора.

Директива **RUN N** служит для запуска программы со строки с номером N. Если номер строки отсутствует, то работа программы начинается со строки с наименьшим номером. Всем переменным присваивается значение 0 или "пустая строка".

Директива **CONT** позволяет продолжить выполнение программы с того места, где она была прервана нажатием на клавиши "УС" и "Е" или при выполнении оператора STOP. В случае невозможности дальнейшего выполнения программы на экран дисплея выдается соответствующее сообщение об ошибке.

Директива **CSAVE** "ИМЯ" позволяет переписать программу на Бейсике из ОЗУ на магнитную ленту. Имя программы может отсутствовать, но кавычки обязательны.

Директива **CLOAD** "ИМЯ" предназначена для загрузки программ на Бейсике с магнитной ленты в ОЗУ. Если имя программы указано, то происходит поиск данной программы, если нет (указаны только одни кавычки), то в ОЗУ загружается первая же встретившаяся на магнитной ленте программа на Бейсике.

Директива **VERIFY** "ИМЯ" производит проверку правильности записи программы на магнитную ленту.

Директива **MERGE** "ИМЯ" поизводит присоединие указанной программы к уже имеющейся в ОЗУ. При этом номера строк второй программы должны быть больше максимального номера строки в первой программе.

Директива **HIMEM N** устанавливает верхнюю границу ОЗУ для программы на Бейсике.

Операторы Бейсика

В Бейсике имеется ряд операторов, которые можно разделить на две группы - выполняемые и невыполняемые. Невыполняемые операторы - это операторы описания.

Оператор **REM** служит для вставки в текст программы комментариев. Он не влияет на выполнение программы, так как все, что стоит в строке за этим оператором, интерпретатором игнорируется.

Оператор **DIM** предназначен для описания массивов, используемых в программе. Массив можно не описывать,

если его размерность не превышает 10. Одним оператором **DIM** можно описать сразу несколько массивов. Наименьшее значение индекса равно 0. Массив должен быть описан до его использования в программе.

Оператором **DATA** можно описывать данные непосредственно в программе. Значения данных присваиваются переменным программы при помощи оператора **READ.** Программа может содержать любое число операторов **DATA**, и располагаться они могут в любом месте программы, независимо от положения операторов **READ.** Оператором **DATA** могут быть описаны любые данные, как числовые, так и символьные.

Все данные, описанные при помощи оператора **DATA,** образуют так называемый "блок данных". Начало блока - данные, описанные самым первым в программе оператором **DATA,** конец - данные, описанные последним оператором **DATA.** Данные из блока можно считать, воспользовавшись оператором **READ.** Они будут считаны последовательно, начиная с первого. После каждого обращения к блоку данных происходит перемещение на одну позицию так называемого внутреннего (для интерпретатора) указателя данных.

Оператор **READ** предназначен для чтения данных из блока данных и присвоения значений переменным программы. Запись этого оператора в программе выглядит так: **READ X1, X2,..., XN,** где **XN** - имена числовых или текстовых переменных. При описании данных программист обязан строго соблюдать соответствие между типом данных и переменных. При каждом выполнении оператора **READ** указатель данных смещается на одну позицию.

Оператор **RESTORE N** служит для перемещения указателя данных на строку с номером **N.** Если **N** не указано, то указатель устанавливается на первый блок данных. Этот оператор обеспечивает возможность повторного считывания данных из блока.

Оператор **INPUT** позволяет вводить данные с клавиатуры дисплея непосредственно при выполнении программы на Бейсике. Значения введенных данных присваиваются переменным, имена которых указывают вслед за оператором **INPUT.** Это могут быть как числовые, так и символьные переменные.

При выполнении оператора **INPUT** на экране дисплея возникает символ "?". В ответ на этот вопрос с клавиатуры вводят данные, которые выводятся на экран в строку сразу после этого символа.

Если оператором **INPUT** необходимо ввести несколько переменных, то после ввода очередного значения необходимо нажать клавишу ",". При вводе можно вводить не только константы, но и выражения. Ввод данных заканчивается нажатием клавиши "ВК". Перед списком переменных в операторе **INPUT** может стоять строка символов, заключенная в кавычки. В этом случае при выполнении оператора на экран дисплея сначала будет выведено указанное сообщение, а затем знак "?".

Оператор **PRINT** предназначен для вывода на экран дисплея значений переменных, сообщений и результатов вычислений. Если оператор использован без операнда, то это приводит к печати одной пустой строки.

Операндов, стоящих вслед за оператором **PRINT,** может быть несколько, и тогда их отделяют друг от друга разделителями. В качестве разделителей используют символы: "," и ";", причем, при использовании первого символа под каждое выводимое значение отводится 14 позиций в строке, а второй служит для компактной печати результатов.

Для печати чисел ипользуется три формы записи: в виде целого числа, числа с десятичной точкой и в экспоненциальном виде. В любой форме "печатается" не более 6 цифр.

Если после последнего операнда в операторе **PRINT** стоит разделитель, то при выполнении следующего оператора **PRINT** печать продолжается в той же строке. Если же разделитель не стоит, то печать начинается с новой строки.

Если после оператора **PRINT** стоит **AT X,Y,** то вывод на экран будет произведен в соответствующем месте экрана.

Оператор **CUR X,Y** служит для перемещения курсора в позицию с координатой X по горизонтали и координатой Y по вертикали. Начало отсчета - левый нижний угол экрана. Диапазон изменения координат курсора по горизонтали - 0-63, по вертикали - 0-24.

Программа на языке Бейсик выполняется строка за строкой, в соответствии с их номерами. Однако имеется ряд операторов, меняющих естественный ход выполнения программы.

Оператор **GOTO N** - это оператор безусловной передачи управления на строку с номером N.

Операторы **GOSUB N** и **RETURN** служат для организации подпрограмм. Оператор **GOSUB** служит для вызова подпрограммы, начинающейся со строки с номером N. Заканчиваться подпрограмма должна обязательно оператором **RETURN.** После его выполнения происходит возврат в то место основной программы, откуда произошел вызов подпрограммы. Допускается многократная вложенность подпрограмм, степень которой ограничена только объемом свободной памяти.

Операторы **ON X GOTO N1, N2,** ... и **ON X GOSUB N1, N2,..** в зависимости от результата вычисления выражения X реализуют условную передачу управления на одну из строк программы, номер которой указан в списке, следующим за оператором.

При выполнении оператора сначала вычисляется значение выражения X, от результата берется целая часть, которая и служит указателем номера строки в списке. Например, если результат выражения равен 1, то управление будет передано на строку N1, двум - на строку N2 и т.д.

Оператор **STOP** предназначен для прекращения выполнения программы и перевода интерпретатора в непосредственный режим.

Оператор **FOR X TO Y STEP Z** - это оператор инициализации цикла, а оператор **NEXT X** - оператор конца цикла. Все, что находится между этими операторами, называют телом цикла.

В операторе инициализации цикла **X** - выражение, задающее имя переменной цикла и присваивающее ей начальное значение, **Y** - выражение, определяющее конечное значение переменной цикла. Выражение Z определяет значение, на которое должна измениться переменная цикла (шаг цикла) после каждого выполнения оператора **NEXT.** Это выражение приводится только в том случае, если шаг цикла отличен от плюс единицы. После оператора **NEXT** указывается имя переменной цикла. Шаг цикла может принимать и отрицательное значение.

Условный оператор **IF X THEN (СПИСОК)** - один из фундаментальных операторов, имеющихся почти в любом алгоритмическом языке программирования высокого уровня. Работа его заключается в следующем:

- проверяется выполнение условия X;
- если X - истина, то выполняются операторы, стоящие в строке после слова THEN;
- если X - ложь, то управление передается следующей строке программы.

Выражение X может включать проверку самых различных условий с использованием как операций отношения, так и логических операций.

Оператор **CLS** предназначен для стирания информации с экрана и всегда должен выполняться прежде, чем начнется формирование графических изображений.

Оператор **HOME** предназначен для очистки экрана и установки курсора в левый верхний угол экрана.

Оператор **PLOT X,Y,Z** позволяет погасить (или высветить) точку с координатами X по горизонтали и Y по вертикали. Если последний операнд равен 1, то соответствующая точка засвечивается, 0 - гаснет.

Оператор **LINE X,Y** позволяет вычерчивать или стирать отрезки прямых линий. Операнды X и Y задают координату конечной точки отрезка. Для задания координат начала отрезка, а также его вида (засветка или гашение) служит оператор **PLOT.**

Оператор **POKE X,Y** позволяет запчисать в ячейку памяти, адрес которой задан выражением X, величину, равную результату выражения Y (значение результата должно находится в диапазоне от 0 до 255 (0H - 0FFH).

Оператор **PAUSE** N позволяет приостановить выполнение программы на N секунд.

И, наконец, еще один оператор, имеющийся в языке Бейсик. Оператор **CLEAR** X предназначен для очистки памяти. Если параметр X не указан, то после выполнения оператора всем числовым переменным присваивается значение 0, а всем символьным - значение "пустая строка". Если же параметр X указывается, то в памяти выде-

ляется область размером X байт, предназначенная для хранения символьных переменных.

Функции Бейсика

В языке Бейсик имеется ряд встроенных функций, которые позволяют значительно упростить написание некоторых программ. Существуют функции, работающие с числовыми и символьными данными, функции преобразования типов данных, а также функции обращения к таким машинным ресурсам, как содержимое ячеек памяти. В этом порядке и будем их рассматривать.

В таблице 2.7 перечислены стандартные встроенные функции языка Бейсик. Две функции из таблицы требуют дополнительных пояснений.

Функция **INT(X)** возвращает в качестве результата наибольшее число, которое меньше или равно X.

Функция **RND(1)** занимает особое место среди других функций, так как широко используется для разработки игровых программ и программ моделирования. Отметим, что аргумент этой функции всегда должен быть равен 1. Результат работы функции - случайное число в диапазоне от 0 до 1.

Тригонометрические функции требуют, чтобы аргумент был указан в радианах.

Язык Бейсик имеет развитые встроенные средства для обработки текстов, что выгодно отличает его от многих других языков программирования высокого уровня. Рассмотрим встроенные функции для работы с символьной информацией.

Результатом работы функции **LEN(X$)** является число, равное количеству символов (длине) символьной переменной X$.

Обращение к функции **LEFT$(X$,N)** позволяет выделить строку символов длиной N, начиная с крайнего левого символа.

Функция **RIGHT$(X$,N)** делает то же самое, но начиная с крайнего правого символа.

ТАБЛИЦА 2.7

ФУНКЦИЯ	РЕЗУЛЬТАТ РАБОТЫ
SQR(X)	КОРЕНЬ КВАДРАТНЫЙ ИЗ X
EXP(X)	ЭКСПОНЕНЦИАЛЬНАЯ ФУНКЦИЯ
LOG(X)	НАТУРАЛЬНЫЙ ЛОГАРИФМ ОТ X
LG(X)	ДЕСЯТИЧНЫЙ ЛОГАРИФМ ОТ X
ABS(X)	АБСОЛЮТНАЯ ВЕЛИЧИНА X ABS(X) = X, ЕСЛИ X>0 0, ЕСЛИ X=0 -X, ЕСЛИ X<0
SGN(X)	ЗНАК X SGN(X) = 1, ЕСЛИ X>0 0, ЕСЛИ X=0 -1, ЕСЛИ X<0
SIN(X)	СИНУС ОТ X (X В РАДИАНАХ)
COS(X)	КОСИНУС ОТ X (X В РАДИАНАХ)
TAN(X)	ТАНГЕНС ОТ X (X В РАДИАНАХ)
ASN(X)	АРКСИНУС ОТ X
ACS(X)	АРККОСИНУС ОТ X
ATN(X)	АРКТАНГЕНС ОТ X
INT(X)	ЦЕЛАЯ ЧАСТЬ ОТ X
RND(1)	СЛУЧАЙНОЕ ЧИСЛО ОТ 0 ДО 1

Воспользовавшись функцией **MID$(X$,N1,N2)**, получим строку символов длиной N2, начиная с позиции N1. Отсчет позиций ведется слева направо.

В программах на Бейсике часто возникает необходимость перевода данных из числового вида в символьный и наоборот. Этой цели служат две специальные встроенные функции.

Функция **STR$(X).** Аргумент X - число или арифметическое выражение, а результат работы функции - строка, являющаяся символьным представлением данного числа.

Функция **VAL(X$)** предназначена для обратного преобразования данных из символьного вида в числовой, начиная с крайнего левого символа символьной переменной X$.

В Бейсике есть две встроенные функции для работы с кодами символов:

- функция **ASC(X)** переводит код первого символа X в десятичное число,
- функция **CHR$(X)** позволяет получить строку, состоящую из единственного символа, код которого равен X (аргумент функции не должен превышать 255).

Для управления форматом печати результатов на экране дисплея служат следующие три функции:

- функция **POS(1)**, результатом работы которой является целое число, равное номеру позиции последнего отпечатанного символа в текущей строке,
- функция **SPC(X)**, позволяющая вставлять в печатаемую строку X пробелов (аргумент X не должен превышать 255),
- функция **TAB(X)** - это функция горизонтальной табуляции, которая позволяет переместить курсор в заданную позицию в строке. Аргумент X и в этом случае не должен превышать 255. Он указывает, сколько позиций необходимо отступить от левого края строки.

При разработке больших программ, а также программ обработки текстов, довольно часто необходимо определить объем свободной памяти, не занятой текстом программы и переменными. Для этих целей в языке Бейсик предусмотрена функция **FRE(X).** Если аргумент этой функции число, то результатом будет количество свободных байтов в памяти. Если аргумент - символьное выражение, то результат - количество свободных байтов в буфере для символьных переменных.

Результатом работы функции **PEEK(X)** будет десятичное число, равное содержимому ячейки памяти, адрес которой определен аргументом X.

Функция **INKEY$** возвращает в качестве результата символ, если на клавиатуре была нажата какая либо клавиша, и пустую строку в противном случае.

Функция **USR(X)** предназначена для организации связи программ, написанных на Бейсике, с подпрограммами, написанными на ассемблере или в машинных кодах. Аргумент X - адрес ячейки памяти, начиная с которой записана программа в машинных кодах. Поэтому, если в выражении встретится обращение к функции **USR(X),** то управление будет передано подпрограмме, расположенной по адресу X. В конце подпрограммы обязательно должна стоять команда **RET,** после выполнения которой управление возвращается программе на Бейсике. Результат работы подпрограммы (в виде одного байта) перед возвратом из подпрограммы помещается в аккумулятор.

Функция **ADDR(X)** дает адрес первого байта значения переменной X.

Функция **SCREEN$(X,Y)** возвращает символ, расположенный на экране дисплея в позиции X,Y.

Кроме перечисленных встроенных функций, в Бейсике имеется возможность вводить в программе определение новых функций и в дальнейшем обращаться к ним в различных выражениях по имени. Определить функцию можно в любом месте программы при помощи оператора **DEF,** важно только, чтобы строка с этим определением была исполнена до обращения к функции. Имена всех функций должны начинаться с символов FN, за ними следуют один или два символа, ограничения на которые такие же, как и на имена переменных.

Оператор **DEF** можно использовать только в программном режиме и, кроме того, допускается определение функций только от одного числового аргумента. Параметр X, стоящий в операторе определения, является формальным, необходимым для обозначения функциональной зависимости. При обращении к функции вместо него указывается фактический аргумент, который заменяет формальный параметр, стоящий в правой части оператора присваивания.

Сообщения интерпретатора об ошибках

Интерпретатор языка Бейсик позволяет в процессе выполнения программы обнаруживать и анализировать ошибки. При обнаружении ошибки на экран выдается одно из следующих сообщений.

NEXT БЕЗ FOR

В программе встретился оператор NEXT, для которого не был выполнен соответствующий оператор FOR.

СИНТАКСИС

Неверен синтаксис строки.

RETURN БЕЗ GOSUB

В программе встретился оператор RETURN без предварительного выполнения оператора GOSUB.

МАЛО ДАННЫХ ПРИ DATA

При выполнении программы нехватает данных для оператора READ, т.е. данных, описанных операторами DATA, меньше, чем переменных в операторах READ.

НЕВЕРНЫЙ АРГУМЕНТ

Аргумент функции не соответствует области определения данной функции. Например, отрицательный или нулевой аргумент функции LOG(X), отрицательный аргумент у функции SQR(X) и т.д.

ПЕРЕПОЛНЕНИЕ

Переполнение при выполнении арифметических операций. Результат любой операции не может быть больше +1,7x10 в 38 степени или меньше -1,7x10 в 38 степени .

МАЛО ОЗУ

Недостаточен объем памяти. Возможные причины: велик текст программы, слишком длинны массивы данных, вложенность подпрограмм и циклов больше нормы, слишком много переменных.

НЕТ СТРОКИ

Нет строки с данным номером. Возникает при выполнении операторов перехода.

НЕВЕРНЫЙ ИНДЕКС

Индекс не соответствует размерности массива.

ПОВТОРНОЕ ОПИСАНИЕ

Повторное выполнение операторов DIM или DEF, описывающих массив или функцию, которые уже были описаны ранее.

ДЕЛЕНИЕ НА НОЛЬ

Попытка разделить на ноль.

ТОЛЬКО В ПРОГРАММЕ

Попытка выполнить операторы INPUT или DEF в непосредственном режиме.

НЕСООТВЕТСТВИЕ ДАННЫХ

Несоответствие типов данных. Возникает при попытке символьной переменной присвоить числовое значение и наоборот.

МАЛ БУФЕР

Переполнение буферной области памяти, отведенной для хранения символьных переменных. Для расширения объема буфера служит директива CLEAR.

X$>255

Длина символьной переменной превышает 255.

СЛОЖНО

Выражение, содержащее сиимвольные переменные, слишком сложно.

НЕЛЬЗЯ

Невозможность продолжения выполнения программы по директиве CONT.

НЕТ DEF

Попытка обратиться к функции, не описанной оператором DEF.

2.5. Программа МАРС

Разработка программ на языке Ассемблер требует, как правило, довольно значительных ресурсов внешней и оперативной памяти. Кроме того, этот трудоемкий процесс требует многократного повторения всех основных операций (редактирование текста, трансляция, отладка). Если в составе ПК имеется накопитель на гибких магнитных дисках, то эта проблема решается достаточно просто.

Рассматриваемая ниже программа МАРС для ПК МИКРОША включает в себя строчно-ориентированный редактор текста, ассемблер и символьный отладчик и позволяет разрабатывать небольшие программы на языке ассемблера микропроцессора КР580ИК80А в достаточно комфортном для программиста режиме.

Редактор MAPC

Программа MAPC имеет объем 5,5 Кбайт и размещается в памяти с адреса 0000H. Вся остальная память используется для хранения текста программы и ее кодов после трансляции. Редактор похож на редактор, имеющийся обычно в интерпретаторах языка Бейсик, и подразумевает нумерацию строк программы. Редактор MAPC имеет следующие директивы:

NEW - эта директива подготавливает редактор к вводу текста новой программы, имеющийся в буфере текст при этом стирается.

CSAVE - директива записи на магнитофон текста программы: После ее выдачи в ответ на запрос необходимо набрать имя программы. Имя имеет длину до 16 символов и используется для последующего поиска программы.

CLOAD - директива чтения текста с ленты. Если имя не указывается, то считывается первый же встретившийся на ленте текст.

AUTO N1, N2 - директива автоматической нумерации строк программы, начиная с номера N1 с интервалом N2. Если параметр N2 не указан, то строки нумеруются через 10.

LIST N1,N2 - вывод на экран текста программы, начиная со строки N1 до строки N2. Если эти параметры не указаны, то выводится весь текст программы. Вывод текста на экран может быть приостановлен нажатием клавиши "ПРОБЕЛ" и продолжен нажатием любой другой клавиши. Вывод текста может быть прерван нажатием клавиши "F4".

RENUM N1,N2 - перенумерация строк всей программы. Параметр N1 - показывает, с какой строки начинать перенумерацию, параметр N2 - через сколько нумеровать строки. Выдача этой директивы без параметров приводит к перенумерации всей программы через 10.

SIZE - выдача на экран количества свободных ячеек памяти для хранения текста программы.

Набор текста программы следует проводить с использованием клавиши "ТАБ" для отделения полей меток мнемоники и операндов друг от друга.

Отладчик MAPC

Отладчик, входящий в состав программы MAPC, позволяет отлаживать программу с использованием символических имен и меток, определенных в исходном тексте программы. Таким образом, нет необходимости запоминать абсолютные адреса и после внесения исправлений в программу "привыкать" к новым их значениям. Символические имена, определенные в программе, указываются в качестве параметров директив. Конечно, можно указывать и абсолютные значения.

Отладчик имеет следующие директивы (они во многом, естественно, повторяют директивы монитора):

DUMP A1,A2 - распечатка содержимого памяти в виде шестнадцатиричных кодов с адреса A1 до адреса A2;

SET A - просмотр и модификация содержимого ячеек памяти, начиная с адреса A. Ввод набранного значения осуществляется по нажатию клавиши "ПРОБЕЛ", окончание работы директивы - нажатием на клавишу "ВК";

REGISTER - распечатка текущего содержимого регистров отлаживаемой программы;

BREAK A - установка контрольной точки в отлаживаемой программе по адресу A. После достижения этой точки при выполнении отлаживаемой программы управление передается отлаживаемой программе;

GOTO A - запуск программы с адреса A;

FIND A1,A2,D8 - по этой директиве проводится поиск байта D8 в области памяти от адреса A1 до A2;

FILL A1,A2,D8 - заполнение области памяти константой;

MOVE A1,A2,A3 - пересылка содержимого области памяти A A2 в область памяти с начальным адресом A3;

ASCII A1,A2 - распечатка содержимого области памяти A1-A2 в виде таблицы символов;

CODE СИМВОЛ - распечатка численного значения кода указанного символа;

При указании абсолютных параметров в этих директивах необходимо помнить, что если после числа не стоит буква H, то оно считается десятичным, в противном случае - шестнадцатиричным.

Ассемблер MAPC

Ассемблер, входящий в состав программы MAPC, выполняет все операции непосредственно в ОЗУ ПК и имеет следующие директивы:

SET - директива присвоения значений символическим именам;

DS - резервирование ячеек памяти;

DB - резервирование байтов и помещение в них указанных констант;

DW - резервирование слов (2 байта) и помещение в них указанных констант;

ORG - указание начального адреса программы;

END - указание конца программы.

Ассемблер позволяет в поле операнда помещать выражения, содержащие операции сложения и вычитания (например 33H+55), а также символьные константы, заключенные в апострофах (например, 'АБВГД'). После шестнадцатиричных констант следует указывать букву "H", после восьмиричных - "Q". Если никакая буква не указана, то число считается десятичным. Символические имена, используемые в программе для меток и операндов должны иметь длину не более 5 символов.

При работе ассемблера коды оттранслированной программы должны размещаться в какой-то области памяти. Первоначально для этой цели выделен буфер размером 1Кбайт, начиная с адреса 7100H. Изменить начальный адрес этой области можно при помощи следующей директивы:

REGION A1 - начальный адрес области транслируемых программ становится равным A1.

Ассемблер вызывается следующей директивой:

ASSEMBLER 1,2,3,4,5,S

Числа задают режимы работы ассемблера:

1 - первый проход ассемблера (просмотр синтаксиса и формирование таблицы меток);

2 - выдача листинга программы на экран дисплея;

3 - вывод кодов оттранслированной программы на магнитную ленту;

4 - запись кодов оттранслированной программы в буферную область памяти, назначенную ранее директивой

REGION. Эти коды потом можно записать на магнитную ленту при помощи соответствующей директивы резидентного монитора;

5 - запись кодов программы, предназначенной для работы в тех адресах, где находится программа MAPC, в буферную область, назначенную оператором REGION.

После трансляции программу необходимо переместить директивой монитора в свои рабочие адреса. Таким образом, этот режим позволяет транслировать программы, которые должны работать в начальной области памяти;

S - выдача на экран таблицы имен, определенных в программе и их значений.

Если при работе ассемблера будут обнаружены какие-либо ошибки, то в самой левой позиции строки листинга оттранслированной программы выводится одна латинская буква - признак ошибки. Имеются следующие признаки ошибок:

P - неправильно записано символьное имя или числовая константа;

O - произошло переполнение таблицы символических имен (слишком большая программа);

D - повторное использование символического имени в качестве метки;

M - неверное использование символического имени в качестве метки;

C - использование недопустимых символов в численных константах;

T - неверное написание команды языка ассемблера.

Для ускорения ввода директив программы MAPC возможно использование следующих сокращений:

LIST = L.	RENUM = RE.
CSAVE = CS.	CLOAD = CL.
RENUME = R.	SIZE = S.
ASSEMBLER = ASS.	DUMP = D.
REGISTER = REGIS.	GOTO = G.
BREAK = B.	FIND = F.

2.6. Графический редактор

Графический редактор предназначен для подготовки графических изображений в ручном и автоматическом режимах.

При работе с графическим редактором экран дисплея содержит 128 точек по горизонтали и 74 точки по вертикали. При этом вертикальные линии, также как и горизонтальные, отображаются как непрерывные. Такой формат отображения достигается путем соответствующей настройки контроллера дисплея КР580ВГ75.

Графический редактор позволяет организовать в памяти ПЭВМ 8 экранных буферов для хранения графической информации и 1 буфер для текстовой информации с форматом отображения 25 строк по 64 символа в каждой. Все буферы имеют свой номер (от 0 до 8). Текстовый буфер имеет номер 0, а графические - от 1 до 8.

Работа с редактором заключается в следующем. После запуска на экране появляется меню директив. Вид этого меню показан в таблице 2.8. Выдав соответствующую директиву пользователь может создавать новые изображения в ручном или автоматическом режиме, сохранять их на магнитной ленте, загружать с магнитной ленты ранее подготовленные изображения и вносить в них изменения, в автоматическом режиме выводить изображения на экран в произвольном порядке и сопровождать их различными звуковыми эффектами.

ТАБЛИЦА 2.8

МИКРОША ГРАФИЧЕСКИЙ РЕДАКТОР ВЕРСИЯ 2.01989 Г.	
>> ОСНОВНЫЕ ДИРЕКТИВЫ <<	
Р - РУЧНОЙ РЕЖИМ	У - СТРОКА СОСТОЯНИЯ (Д)
З - ЗАПИСЬ ЭКРАНА НА ЛЕНТУ	Ф - ФОРМИРОВАНИЕ МАКРОКОМАНДЫ
Ч - ЧТЕНИЕ ЭКРАНА С ЛЕНТЫ	Т - ЗАПИСЬ МАКРОКОМАНДЫ НА ЛЕНТУ
К - КОПИРОВАНИЕ ЭКРАНОВ	Л - ЧТЕНИЕ МАКРОКОМАНДЫ С ЛЕНТЫ
С - СТИРАНИЕ ЭКРАНА	Г - ВЫПОЛНЕНИЕ МАКРОКОМАНДЫ
ВАША КОМАНДА?	
ПОЛЕ ДЛЯ МАКРОКОМАНДЫ	

Редактор позволяет, кроме подготовки графических изображений, выводить на экран и алфавитно-цифровую информацию в двух режимах. В первом режиме формат экрана - 37 строк по 64 символа. Во втором режиме символы формируются графически с использованием матрицы размером 6 на 8 точек. В этом случае на экран дисплея можно вывести до 9 строк алфавитно-цифровой информации по 21 символу в каждой. Размер символов примерно в 3 раза больше, чем при использовании стандартного режима отображения. Рассмотрим каждый из режимов работы редактора подробно.

Директивы ручного режима

После выдачи директивы "PN" (ручной режим) на экране дисплея отображается текущее содержимое буфера, номер которого определяет параметр N. При первоначальном запуске редактора проводится очистка всех буферов. При последующих запусках редактора содержимое буферов не стирается. При вводе директив и их параметров последний набранный символ может быть удален при помощи клавиши "<-" (стрелка влево). Ввод директивы заканчивается нажатием на клавишу "ВК". Директивы, набранные с какими либо ошибками, просто игнорируются, при этом выдается звуковой сигнал индикации ошибки (высокий тон, затем низкий).

В качестве графического курсора используется мигающий квадратик, равный по размерам одной отображаемой графической точке. При помощи команд, перечисленных в таблице 2.9, его можно перемещать в произвольное место экрана. Все команды выдаются путем нажатия и удержания клавиши "УС" и одной из указанных клавиш. Основные команды перемещения графического курсора выдаются простым нажатием на клавишу функциональной клавиатуры.

ТАБЛИЦА 2.9

КЛАВИША	КОМАНДА	ПРИМЕЧАНИЕ
"<-"	Курсор влево	"УС"+"H"
"->"	Курсор вправо	"УС"+"X"
"^"	Курсор вверх	"УС"+"Y"
" "	Курсор вниз	"УС"+"Z"
F2	Курсор вверх-вправо	"УС"+"A"
" "	Курсор вверх-влево	"УС"+"L"
F5	Курсор вниз-вправо	"УС"+"D"
F3	Курсор вниз-влево	"УС"+"B"
"УС"+"P"	"Прозрачный курсор"	
"УС"+"K"	Курсор - "карандаш"	
"УС"+"L"	Курсор - "ластик"	
"УС"+"F"	Пометить начало отрезка	
"УС"+"S"	Провести отрезок	
"УС"+"M"	Пометить начало блока	"ВК"
"УС"+"J"	Пометить конец блока	"ПС"
"УС"+"N"	Переслать блок в позицию курсора	
"УС"+"I"	Проинвертировать изображение в блоке	"ТАБ"
"УС"+"E"	Стереть блок	
"УС"+"U"	Снять определение блока	
"УС"+"]"	Сдвинуть экран вверх на 2 точки	
"УС"+"["	Сдвинуть экран вниз на 2 точки	
"УС"+"O"	Сдвинуть экран влево на 2 точки	
"УС"+"V"	Сдвинуть экран вправо на 2 точки	
"AP2"	Включить текстовый экран	"УС"+"["

При перемещении курсора по экрану возможны 3 варианта. В первом случае курсор перемещается по экрану, не изменяя его содержимого ("прозрачный курсор"), во втором - за курсором остается след в виде светлой линии ("курсор - карандаш") и в третьем - перемещение курсора приводит к стиранию точек на экране (курсор - "ластик"). Выбор одного из этих режимов осуществляется соответствующими командами, также приведенными в таблице 2.9.

Кроме получения изображений на экране с помощью "курсора-карандаша", оператор может формировать отрезки прямых линий следующим образом. Курсор помещается в то место экрана, где будет начало отображаемого отрезка, и оператор выдает команду "УС"+"F". Затем курсор помещается в то место, где будет конец отрезка (предварительно следует включить "прозрачный" курсор) и выдается команда "УС"+"S". На экране сразу появляется отрезок прямой линии, соединяющий эти две точки. Установленные с помощью команды "УС"+"F" координаты начала отрезка сохраняется до того момента, когда эта команда будет выдана вновь.

При формировании изображений часто возникает необходимость переноса части изображения из одного места экрана в другое или стирания части изображения. Для этого служат команды работы с блоком. Под блоком в данном случае понимается прямоугольная область на экране, заданная координатами левого нижнего и правого верхнего углов. В любой момент работы редактора может быть определен только один блок.

Для того, чтобы пометить блок, курсор необходимо поместить в его левый нижний угол и выдать команду "УС"+"M". Затем курсор помещают в правый верхний угол блока и выдаюет команду "УС"+"J". Для того, чтобы получить копию блока в другом месте экрана, оператор перемещает курсор туда, где будет находится левый нижний угол перемещенного блока и выдает команду "УС"+"C". В указанном месте экрана появится копия блока. Команда копирования сохраняет определение блока, так что его можно копировать многократно в разные места экрана. Если команда "УС"+"J" выдается до того, как была выдана команда "УС"+"M", или начало блока было помечено на другом экране, то она просто игнорируется.

Команда "УС"+"U" отменяет заданное ранее определение блока. Команда "УС"+"E" позволяет стереть все, что находится внутри помеченного блока. После ее выполнения определение блока снимается. Команда "УС"+"I" позволяет проинвертировать изображение, находящееся внутри блока. Повторная выдача этой команды восстанавливает первоначальный контраст.

Манипуляции с блоком можно проводить не только в пределах одного экрана. Блок, помеченный на одном из экранов, может быть скопирован на любой другой. Для этого после задания блока при помощи описанных выше директив необходимо сменить текущий экран на нужный (при помощи последовательности команд "AP2"+"PN") и выдать директиву копирования. Нажатие на клавишу

"АР2" при работе в ручном режиме всегда приводит к выдаче на экран меню директив, в соответствии с таблицей 2.8.

При помощи команд, приведенных в таблице 2.9, оператор может сдвинуть содержимое графического экрана вверх, вниз, влево или вправо на 2 точки. Следует учитывать, что часть сдвинутого изображения при этом будет потеряна.

Для того, чтобы оператор мог легко ориентироваться в установленных режимах работы, в самом низу экрана отображается так называемая строка состояния. Назначение отдельных полей строки следующее:

"К=Л" - вид курсора (К -"карандаш", Л -"ластик, "П" - "прозрачный");
"Э=1" - номер текущего графического экрана;
"X=103" - позиция курсора по горизонтали;
"Y=20" - позиция курсора по вертикали;
"XL=99 - координата начала отрезка по горизонтали;
"YL=12" - координата начала отрезка по вертикали;
"B=1" - номер графического экрана, на котором помечен блок. Если номер равен 0, то блок не помечен;
"X1=10" - координата левого нижнего угла блока по горизонтали;
"Y1=11" - координата левого нижнего угла блока по вертикали;
"X2=30" - координата правого верхнего угла блока по горизонтали;
"Y2=20" - координата правого верхнего угла блока по вертикали.

Если координаты начала отрезка или блока не определены, то в соответствующих позициях строки состояния помещаются пробелы. Информация в строке состояния модифицируется при каждом изменении координат курсора на экране.

Пользуясь этой справочной информацией можно легко определить, какие текущие режимы работы установлены и осуществить "привязку" местоположения курсора к изображению. Переключив курсор в "прозрачный" режим и перемещая его по изображению можно провести оцифровку изображения и использовать эти данные в прикладной программе.

Команда "У" из списка, приведенного в таблице 2.8, позволяет разрешить или запретить отображение строки состояния. Первоначально отображение строки разрешено, о чем свидетельствует буква "Д" в круглых скобках. При каждой выдаче этой команды это условие заменяется на противоположное.

Если при работе в ручном режиме оператор нажмет одну из алфавитно-цифровых клавиш, то на экран дисплея будет выведено изображение соответствующего символа, а графический курсор автоматически переместится вправо на 6 позиций. При необходимости оператор может подкорректировать положение курсора, которое всегда определяет левый нижний угол выводимого символа. Таким образом, возможно формирование надстрочных и подстрочных символов.

Выход из ручного режима осуществляется при нажатии на клавишу "АР2". На экране при этом отображается первоначальное меню директив.

Подготовленные в ручном режиме изображения могут быть сохранены на магнитной ленте и в дальнейшем вновь введены в память ПЭВМ. Для записи служит директива "З". Ее формат следующий: ЗN. На ленту будет записано содержимое экрана с указанным номером.

Для чтения с магнитофона предусмотрена директива "ЧN". По этой директиве информация вводится в тот графический буфер, номер которого указан в директиве.

Для копирования содержимого одного экрана на другой служит директива "К". Она имеет следующий формат: KN1,N2. Параметр N1 определяет экран-источник, а параметр N2 - экран-получатель информации. При копировании содержимое экрана-источника не изменяется.

Для стирания всего содержимого экрана служит директива "СN". Параметр N определяет номер экрана.

Директивы автоматического режима

Одной из особенностей редактора является возможность автоматического формирования изображений и управление последовательностью их отображения на экране дисплея. Для этого оператор должен подготовить специ-

альную макрокоманду, описывающую требуемые действия.

Макрокоманда может вводится как с клавиатуры, так и с магнитной ленты. Для макрокоманды отведены нижние 17 строк на текстовом экране (поле макрокоманды). Таким образом, максимальная длина макрокоманды - 1088 символов (17X64=1088).

Ввод макрокоманды с клавиатуры дисплея начинается с выдачи директивы "Ф". При этом текстовый курсор (мигающая черточка) помещается в левом верхнем углу поля макрокоманды. Оператор может поместить курсор в любое место поля макрокоманды при помощи команд, перечисленных в таблице 2.10. Клавиша "ЗБ" служит для стирания символа, стоящего слева от курсора. Стирание всей информации в поле макрокоманды осуществляется путем двухкратного нажатия на клавишу "СТР".

ТАБЛИЦА 2.10

КОМАНДА	НАЗНАЧЕНИЕ	ПРИМЕЧАНИЕ
"<-"	Курсор влево на одну позицию	"УС"+"H"
"->"	Курсор вправо на одну позицию	"УС"+"X"
" "	Курсор вверх на одну позицию	"УС"+"Y"
" "	Курсор вниз на одну позицию	"УС"+"Z"
" "	Курсор в левый верхний угол поля	"УС"+"L"
"ЗБ"	Стереть символ слева от курсора	
"СТР"	Стереть всю макрокоманду	
"АР2"	Закончить ввод (редактирование)	

При вводе макрокоманды клавишу "ВК" следует нажать только в самом конце. При этом на экране появится светлый прямоугольник - признак конца макрокоманды. Выполнение директивы "Ф" заканчивается при нажатии на клавишу "АР2". При последующем запуске макрокоманды на исполнение при помощи директивы "Г" следует предварительно убедиться, что признак конца макрокоманды установлен в соответствующем месте поля макрокоманды.

При помощи директивы "Т" макрокоманда может быть записана на магнитную ленту, а при помощи директивы "Л" - считана с ленты в ОЗУ ПК.

Макрокоманда состоит из последовательности операторов и операндов. Операнды разделяются при помощи

символов "," (запятая). Между отдельными операторами макрокоманды допускается помещать любое число пробелов. Между операторами и операндами не должно быть других символов.

В таблице 2.11 приведены операторы макрокоманды и указано их назначение. Как видно, большинство действий по манипуляции с изображениями в ручном режиме доступны и в автоматическом режиме. Выполнение оператора "D" приводит к выдаче на экран содержимого указанного буфера. При этом он автоматически становится буфером для записи и для считывания информации, т.е. становится текущим буфером. Для того, чтобы готовить изображение в "невидимом" в данный момент буфере, необходимо назначить его буфером для записи при помощи оператора "W". Если после этого требуется проводить вывод изображений в отображаемый в данный момент буфер, необходимо вновь применить оператор "D". В любом случае общее правило такое: модификация содержимого буфера возможна только, если он назначен как буфер для записи при помощи оператора "W".

Позиция графического курсора изменяется при помощи оператора "P". Третий операнд этого оператора задает контраст для операторов "L" и "T". Начальная точка для проведения отрезков прямых линий определяется текущей позицией графического курсора.

Наличие двух операторов цикла (S и F) позволяет организовать вложенные циклы. При этом допустимым являются вложения

вида: S -- F -- O -- Q и F -- S -- Q -- O. Вложения вида: S -- F -- Q -- O и F -- S -- O -- Q недопустимы.

После запуска макрокоманды при помощи директивы "Г" из основного меню, она начинает выполняться. Если будет обнаружена какая-либо ошибка, то выполнение макрокоманды заканчивается, и выдается специальный звуковой сигнал индикации ошибки (низкий тон, затем высокий тон). Если при выполнении макрокоманды будет нажата клавиша "F4" на функциональной клавиатуре, то выполнение макрокоманды заканчивается досрочно.

ТАБЛИЦА 2.11

ОПЕРАТОР	НАЗНАЧЕНИЕ	ОПЕРАНДЫ
DN	Назначить буфер для отображения	N-номер буфера
WN	Назначить буфер для записи	- : -
RN	Назначить буфер для чтения	- : -
Y	Очистить текущий буфер	нет
CN1, N2	Копирование содержимого буфера	N1-ист., 2-получ.
UN	Сдвинуть экран вверх на N*2 точ.	N - количество
NN	Сдвинуть экран вниз на N*2 точ.	N - количество
EN	Сдвинуть экран влево на N*2 точ.	N - количество
HN	Сдвинуть экран вправо на N*2 точ	N - количество
PX, Y, Z	Высветить точку	X, Y- координаты Z-контраст (0 или 1)
LX, Y-	Провести отрезок	X, Y- координаты конца отрезка
BX1, Y1 X2, Y2	Определить блок в текущем буфере	X1, Y1- лев. нижн. X2, Y2- - прав. верхн.
I	Инвертировать изображение блока	нет
J	Стереть блок	нет
M	Переслать блок в позицию курсора	нет
Z	Снять определение блока	нет
SN	Начало цикла 1	N-количество повторений
Q	Конец цикла 1	нет
FN	Начало цикла 2	N-количество повторений
O	Конец цикла 2	нет
T строка	Выдача текста строки на экран	строка в кавычках
K	Ожидание ввода любого символа с клавиатуры	нет
GN1, N2	Выдача звукового сигнала	N1-высота тона N2-длительность
XN	Временная задержка	N-количество десятых долей с.

Запуск макрокоманды может быть осуществлен и без выхода в основное меню. Для этого, находясь в режиме формирования макрокоманды, необходимо поместить

текстовый курсор под тот оператор, начиная с которого будет выполняться макрокоманда и нажать клавиши "УС"+"G". Таким образом, в поле макрокоманды может быть помещено несколько отдельных макрокоманд, запуск которых можно осуществлять при помощи описанных действий. Эту возможность можно использовать и для отладки макрокоманд.

2.7. Редактор текстов

Одним из основных компонентов программного обеспечения любой ЭВМ является программа - редактор текстов или просто редактор. Редактор позволяет набирать на клавиатуре новые тексты, вносить изменения (редактировать) ранее набранные тексты, сохранять набранные тексты на магнитной ленте или магнитных дисках.

Описываемый редактор тестов является редактором экранного типа. Для выдачи директив редактирования оператор использует как основную, так и дополнительную клавиатуры. Использование дополнительной клавиатуры значительно упрощает и ускоряет работу оператора. При отсутствии дополнительной клавиатуры директивы выдаются путем нажатия и удержания клавиши "УС" и одной из алфавитно-цифровых клавиш.

Программа-редактор занимает 4 Кбайт памяти. Остальная часть ОЗУ используется в качестве текстового буфера. Таким образом, в ОЗУ объемом 32 Кбайт можно хранить и редактировать текст объемом примерно 15 машинописных страниц.

При работе редактора экран дисплея разделяется на несколько частей (окон). Вид экрана показан в таблице 2.12. В первом окне выводится список директив, доступных оператору в данный момент. При изменении режима работы редактора информация в этом окне также меняется. Будем называть это окно "окно меню".

ТАБЛИЦА 2.12

ОКНО МЕНЮ	
ОКНО СООБЩЕНИЙ	СПРАВОЧНОЕ ОКНО
ТЕКСТОВОЕ ОКНО	

Второе окно - "окно сообщений" - предназначено для выдачи различных сообщений, в частности, сообщений об ошибках допущенных оператором во время работы, или ошибок ввод-вывода.

Третье окно предназначено для выдачи информации о текущем положении курсора и объеме свободной памяти в текстовом буфере. Будем называть это окно справочным. Информация в этом окне обновляется при выдаче некоторых директив или сообщений об ошибках. Вид справочного окна приведен в таблице 2.13.

ТАБЛИЦА 2.13

ОТ НАЧАЛА: 4333 ДО КОНЦА: 3351 СВОБОДНО: 16432

Наконец, четвертое окно ("текстовое") предназначено для отображения редактируемоего текста. Все перечисленные окна видны на экране дисплея одновременно.

В текстовом окне отображается 8 строк редактируемого текста по 60 символов в каждой. В этом окне используется свой курсор, реализованный программными средствами. На экране он отображается в виде светлого прямоугольника. При помощи клавиш управления (клавиши со стрелками на дополнительной клавиатуре) курсор можно перемещать в любое место редактируемого текста. Причем, при попытке переместить курсор за пределы текстового окна вся информация в окне сдвигается вверх или вниз, влево или вправо в зависимости от направления движения курсора. Таким образом, через текстовое окно

можно просмотреть текст, содержащий любое количество строк и состоящий из строк произвольной длины. Ограничения на количество строк и количество символов в строке связаны только с имеющимся объемом ОЗУ.

Основные директивы редактора текстов

В каждый данный момент работы оператору доступен один из двух наборов директив редактирования, информация о которых отображается в окне меню. В таблице 2.14 показан вид окна меню при первоначальном запуске редактора. Рассмотрим, для чего предназначены и каким образом выполняются эти директивы.

ТАБЛИЦА 2.14

МИКРОША РЕДАКТОР ТЕКСТОВ МОСКВА 1989 Г.	
F1 - РУС+ЛАТ <-> ЛАТ F2 - ПРЕДЫДУЩАЯ СТРАНИЦА F3 - СЛОВО СЛЕВА П F4 - СЛОВО СПРАВА F5 - СЛЕДУЩАЯ СТРАНИЦА	СТР - СТИРАНИЕ СТРОКИ ЗБ- СТИРАНИЕ СИМВОЛА СЛЕВА С - СТИРАНИЕ СИМВОЛА СПРАВА ГТ - ТАБУЛЯЦИЯ АР2 - ПРОЧИЕ КОМАНДЫ

Директива "курсор влево" (клавиша <-).

Курсор перемещается влево на один символ. Если курсор находится в начале строки, то он перемещается в конец предыдущей. Если курсор уже находится в начале текста, то никаких изменений на экране не происходит.

Директива "кусор вправо" (клавиша ->).

Курсор перемещается вправо на один символ. Если курсор находится в конце строки, то он перемещается в начало следующей. Если курсор уже находится в конце текста, то никаких изменений на экране не происходит.

Директива "курсор вверх" (клавиша Л).

Если курсор уже находится в начале строки, то он перемещается в начало предыдущей. В противном случае курсор перемещается в начало строки, в которой он находится. Если курсор находится в начале текста, то никаких изменений на экране не происходит.

Директива "курсор вниз" (клавиша V).

Курсор перемещается в начало следующей строки. Если курсор находится в конце текста, то изменений на экране не происходит.

Директива "следующая страница" (клавиша F5).

В текстовом окне появляются следующие 8 строк редактируемого текста, которые и называются в данном случае "страницей". Курсор перемещается в левый верхний угол окна. Если в окне уже отображаются последние строки текста, то курсор перемещается в конец текста. Если курсор уже находится в конце текста, то содержимое текстового окна не меняется.

Директива "предыдущая страница" (клавиша F2).

В текстовом окне отображаются предыдущие 8 строк редактируемого текста. Курсор перемещается в левый верхний угол окна. Если курсор уже находится в начале текста, то содержимое окна не изменяется.

Директива "слово справа" (клавиша F4).

Курсор устанавливается перед первым символом следующего слова. Разграничителем слов считается символ "пробел" и символ "ВК". Если курсор находится в конце текста, то текстовое окно остается неизменным.

Директива "слово слева" (клавиша F3).

Курсор устанавливается перед первым символом предыдущего слова. Если курсор уже находится в начале текста, то изменений не происходит.

Директива "стирание символа слева" (клавиша "ЗБ")

Символ, находящийся непосредственно слева от курсора стирается. Если слева больше нет символов, то изменений не происходит.

Директива "стирание символа справа" (клавиша "ПС").

Символ, находящийся непосредственно справа от курсора стирается. Если справа больше нет текста, то изменений не происходит.

Директива "стирание строки" (клавиша "СТР").

Если курсор находится в начале строки, то вся строка стирается, включая и символ "ВК" находящийся в ее конце. В противном случае стирается часть строки справа от курсора, символ "ВК" не стирается. Если курсор находится в конце текста или в конце строки, то изменений не происходит.

Директива "ввод символа"(любая алфавитно-цифровая клавиша). Символ, соответствующий нажатой клавише, вводится в текст слева от курсора. Если в памяти нет больше места, то изменений не происходит.

Директива "ввод новой строки" (клавиша "ВК").

Символ "ВК" вводится в текст. Если курсор находится посредине строки, то эта строка разделяется на две. Если в памяти нет больше места, то изменений не происходит.

Директива "переключение набора знаков" (клавиша F1)

Каждое очередное нажатие на эту клавишу меняет набор отображаемых алфавитно-цифровых символов. Имеется два таких набора:

- прописные русские и прописные латинские буквы;
- прописные и строчные русские буквы.

Директива "горизонтальная табуляция" (клавиша "ТАБ")

Каждое очередное нажатие на эту клавишу приводит к перемещению курсора вправо по строке на 5 позиций.

Дополнительные директивы редактора текстов

Если при редактировании будет нажата клавиша "АР2", то в окне меню появляется новая информация, показанная в таблице 2.15. Нажатие одной из клавиш, показанных в меню, приводит к выполнению соответствующей директивы. Директива воспринимется независимо от того, в каком состоянии находится клавиатура ПК (горит или нет светодиод "РУС").

Рассмотрим назначение и порядок выполнения этих директив.

ТАБЛИЦА 2.15

Н - В НАЧАЛО ТЕКСТА	Б - ПОМЕТИТЬ БЛОК	П - ПЕЧАТЬ ТЕКСТА
К - В КОНЕЦ ТЕКСТА	М - ЗАПИСЬ БЛОКА	Е - КОНЕЦ РАБОТЫ
З - ЗАПИСЬ ТЕКСТА	У - СТЕРЕТЬ БЛОК	
Ч - ЧТЕНИЕ ТЕКСТА	В - БЛОК В БУФЕР	
С - СТЕРЕТЬ ТЕКСТ	И - БЛОК ИЗ БУФЕРА	

Директива "курсор в начало текста" (клавиша "Н").

Курсор перемещается в начало текста. Если курсор уже находится в начале текста, то изменений не происходит.

Директива "курсор в конец текста" (клавиша "К")

Курсор перемещается в конец текста. Если курсор уже находится в конце текста, то изменений не происходит.

Директива "запись текста" (клавиша "З").

При работе редактора используется компактное представление текста в текстовом буфере. Смежные символы "пробел" хранятся в текстовом буфере в виде одного байта. Это позволяет экономно использовать память. Для того, чтобы можно было легко различать по звуку тексты, подготовленные данным редактором от программ в начале вместо ровного тона записывается переменный звуковой тон.

Директива "чтение текста" (клавиша "Ч").

Если перед выдачей этой директивы в текстовом буфере уже имеется какой либо текст, то новая информация вводится в буфер сразу же за текущей позицией курсора. Это позволяет объединить в ОЗУ несколько частей текста в одну. Если при вводе будет обнаружено наличие ошибок, то на экран выводится соответствующее сообщение.

Директива "пометить начало блока" (клавиша "Б").

При редактировании текстов часто возникает необходимость вычеркнуть одно или несколько предложений или поменять их местами. Когда текст находится на бумаге, эта процедура выполняется обычно с помощью ножниц и

клея. В описываемом редакторе для этой цели имеется ряд директив. Все эти директивы работают с частью текста, называемой блоком. Первая из этих директив и позволяет пометить начало блока. Для этого надо поместить курсор в нужное место текста и нажать последовательно клавиши "АР2" и "Б". Для пометки конца блока не существует специальной директивы. Концом блока считается то место в тексте, где в данный момент расположен курсор.

Директива "удалить блок" (клавиша "У").

При выполнении этой директивы из текста удаляется помеченный ранее блок. При этом текст, находящийся за курсором смыкается с текстом, стоящим перед началом блока. Если курсор находится в тексте ранее метки начала блока, то никаких изменений в тексте не происходит.

Директива "записать блок на ленту" (клавиша "М").

При выполнении этой директивы помеченный блок записывается на ленту. Если курсор находится выше метки "начало блока", то выдается сообщение об ошибке. Формат записи блока полностью аналогичен формату записи всего текста.

Таким образом, описанные выше директивы позволяют выполнять с текстами процедуру "ножницы - клей". Если блок имеет небольшие размеры (менее 15 строк по 60 символов в каждой), то для выполнения этой процедуры можно воспользоваться описанными ниже директивами.

В ОЗУ ПК выделен небольшой временный буфер объемом 1 Кбайт, в который можно поместить блок или из которого можно данный блок переместить в текстовый буфер.

Директива "записать блок в буфер" (клавиша "В").

По этой директиве блок копируется во временный буфер. Имеющаяся до этого в буфере информация стирается. Если размер блока превышает размер буфера, то в буфер переписывается только часть блока.

Директива "прочитать блок из буфера" (клавиша "И").

Блок из временного буфера копируется в текстовый буфер сразу же за позицией курсора. После выполнения этой директивы содержимое буфера не меняется. Это позволяет очень просто вводить в текст многократно повторяющиеся фрагменты.

Директива "печать текста" (клавиша "П").

В редакторе предусмотрены средства для печати набираемых текстов на печатающем устройстве любого типа.

После выдачи директивы "П", в ответ на появляющиеся в окне сообщений вопросы, необходимо ввести ряд параметров, определяющих формат печатаемого документа:

"ЛЕВАЯ ГРАНИЦА ТЕКСТА?"

Оператор задает количество символов, определяющих отступ от левого края листа. Если будет просто нажата клавиша "ВК", то по умолчанию отступ равен 8 символам.

"КОЛИЧЕСТВО СТРОК?"

Оператор задает количество строк на странице. После печати заданного числа строк процесс печати приостанавливается, пока не будет нажата клавиша "ПРОБЕЛ" на клавиатуре компьютера. По умолчанию число строк равно 30.

"ПРОПУСК СТРОК?"

Оператор задает количество пустых строк, вставляемых после печати каждой строки текста. По умолчанию этот параметр равен 1.

"ШРИФТ?"

Этот параметр определяет тип шрифта.

После ввода всех указанных параметров начинается печать текста.

Директива "конец работы" (клавиша "Е").

После выдачи этой директивы управление передается программе МОНИТОР.

Сообщения редактора текстов

При работе редактора в окне сообщений могут появиться следующие сообщения.

НЕВЕРНЫЕ ДЕЙСТВИЯ

Это сообщение указывает на ошибку, допущенную оператором. Оно появляется, например, при попытке выдать несуществующую директиву или записи на ленту блока в тот момент, когда курсор находится выше метки начала блока и т.д.

ОШИБКА ВВОДА/ВЫВОДА

Это сообщение указывает на наличие ошибок при вводе или выводе данных с или на магнитную ленту.

НЕДОСТАТОЧНО ПАМЯТИ

Это сообщение указывает на то, что в текстовом буфере нет больше места для ввода новых символов.

При появлении сообщения об ошибке необходимо нажать любую клавишу на клавиатуре, после чего редактор вновь переходит в режим приема директив оператора.

ЛИТЕРАТУРА

1. Зеленко Г.В., Панов В.В., Попов С.Н. Домашний компьютер. М. Радио и связь. 1989 год.

2. Попов С.Н. Графический редактор для ПЭВМ МИКРОША. Микропроцессорные средства и системы N5, 1987 год.

3. Попов С.Н. Программа-имитатор ДОС для 8-разрядных ПЭВМ. Микропроцессорные средства и системы N6, 1987 год.

4. Горшков Д.В., Зеленко Г.В., Озеров Ю.В., Попов С.Н. Персональный радиолюбительский компьютер РАДИО-86РК. Радио N 4,5,6,7,8,9, 1986 год.

5. Попов С.Н. ПЗУ для Бейсика. Радио, N2, 1987 год.

6. Барчуков.В., Фадеев Е. Бейсик "Микрон", Радио, N8, 1988 год.

Содержание

П О П О В АЛЕКСАНДР СЕРГЕЕВИЧ

Персональный компьютер МИКРОША и его программное обеспечение

Подписано в печать 14.08.90 с готового оригинал-макета. Объем 4 п.л. Тираж 10 000 экз. Цена 3 руб. Заказ N1738
Печать офсетная. Отпечатано в ЦИТП Госстроя СССР.