

А. А. Ларченко

**Теневой сервис-монитор
для компьютера
SCORPION ZS 256**

Санкт-Петербург 1994

Ларченко А. А. Теневой сервис-монитор для компьютера Scorpion ZS 256. _ СПб.: МОА, 1993. —48с.

Приводится описание работы с программой теневого сервис-монитора (V2.2) - программного обеспечения встроенного в компьютер Scorpion ZS 256.

Все замечания и пожелания по улучшению книги, а также информация об обнаруженных ошибках в работе компьютера и сервис-монитора будут приняты авторами с благодарностью.

Оригинал-макет подготовлен автором.

© 1993 Ларченко А. А.

© 1992, 1993 Зонов С. Ю.

© 1993 МОА

® Scorpion ZS 256

Введение

Компьютер Scorpion ZS 256 в своей постоянной памяти имеет три достаточно независимые друг от друга программы. Во-первых, это интерпретатор языка Бейсик, который по совместительству выполняет еще функции операционной системы. После включения питания именно он первым получает управление и отвечает как за себя (выполнение собственно программы на Бейсике), так и за чисто системные функции (ввод и редактирование текста программы, загрузка из внешней памяти – с магнитофона, вывод на принтер, вызов «программы в кодах» и так далее).

Второй программой следует назвать дисковую операционную систему TR-DOS. Она вызывается из Бейсика и служит как бы его дополнением – через TR-DOS Бейсик получает доступ к другому устройству внешней памяти – дисководу. Необходимо отметить, что в данный конкретный момент времени на компьютере может выполняться одна программа. Это может быть либо сам Бейсик, распечатывающий экран на принтере; либо TR-DOS, выводящий на экран каталог дискеты, либо загруженная и запущенная через Бейсик игрушка. Объединить во времени эти программы обычными средствами невозможно.

Сервис-монитор – третья программа, находящаяся в памяти компьютера – разработана специально для Scorpion'a. Она дополняет Бейсик и TR-DOS, и делает возможным одновременное выполнение нескольких программ*. Монитор позволяет остановить программу, выполнить некоторые действия, а затем продолжить работу прерванной программы. Термином «некоторые действия» обозначены команды теневого сервис-монитора, подробное описание которого Вы найдете в этой книге.

Пользователям

Для чего нужно прерывать программу? Некоторые действия мы уже упоминали, например, распечатка экрана на принтере или вывод оглавления дискеты. Кроме этого, возможны проверка и форматирование дискет, сохранение состояния программы на дискете, с последующим запуском ее с прерванного места (хотя такая функция выполняется и стандартным TR-DOS'ом).

Однако прерывание программ не единственная задача, решаемая монитором. Другая, и не менее важная, функция монитора – управление аппаратными возможностями компьютера. Программа, называемая SetUp, позволяет изменять характеристики интерфейса принтера, скорость работы клавиатуры, выбирать типы дискет и другие параметры. Отметим, что в Scorpion'e вывод на принтер из Бейсика TR-DOS'a организован через монитор. Таким образом установив в мониторе характеристики своего принтера можно печатать из всех программ через стандартные средства операционной системы Бейсик.

* Замечание для специалистов в области программирования. Приведенное утверждение не означает, что из Sinclair'a сделали VAX или IBM PC.

Подробное описание системы меню сервис-монитора дано в первой части книги.

Хакерам

Иногда возникает желание «скрутить» неутомимый счетчик жизней в игре, или поискать новые возможности в плохо документированных программах. Сервис-монитор и тут выручит Вас. Прерванную программу можно не только посмотреть «изнутри», но и модифицировать по своему вкусу.

Команды монитора позволяют изменять содержимое любых компонентов компьютера – от регистров и состояний микропроцессора, ячеек памяти, и до регистров внешних устройств. Прерванную программу можно выполнять по шагам (трассировать) или, воспользовавшись специальными ловушками, следить за ходом ее выполнения.

Описанию работы команд и режимов монитора-отладчика посвящена вторая часть книги.

Программистам

Люди, разрабатывающие программы для компьютера, наверняка согласятся с тем, что средств предоставляемых Бейсиком и TR-DOSом далеко не всегда хватает даже для, казалось бы, простых вещей. Монитор, наряду с великолепными средствами отладки программ, дает в распоряжение программиста некоторые, специально для этого предназначенные, подпрограммы.

Система экранов пользователя предоставит многооконный интерфейс, а встроенный интерпритатор дисковых функций позволит обмениваться информацией с такими компьютерами, как Robotron или IBM. В третьей части книги Вы найдете подробное описание встроенных подпрограмм монитора.

Предусмотрены нами и интересы программистов пишущих специально для Scorpiona. Каждый компьютер имеет свой индивидуальный номер. Таким образом, можно писать программы, устанавливающиеся на определенный экземпляр машины, и работающие только на нем (подробнее см. приложения Б и В).

Профессионалам

Объем памяти, установленной в компьютере, не беспределен, поэтому часть функций монитора пришлось вынести из ПЗУ. В настоящее время ведется работа над профессиональным расширением. Эта программа должна работать совместно с монитором в оперативной памяти и будет загружаться туда с дискеты.

Наиболее интересной частью профессионального расширения является программа-анализатор для команд трассировки. Анализатор по сути является ядром языка Форт. Словарь языка расширен специфическими словами, позволяющими управлять ходом трассировки. Можно определять свои собственные слова, анализирующие в конце каждой выполненной команды те или иные параметры, а также писать свои программы.

Часть 1. Сервис-Монитор

1.1. Вход в сервис-монитор

Для вызова теневого сервис-монитора необходимо просто нажать на кнопку **MAGIC**. При этом на экране появится список команд (меню), каждая команда занимает одну строку (*пункт*). Для выбора нужной команды можно воспользоваться двумя способами. Во-первых, нажатие клавиши **ENTER** вызовет выполнение текущего пункта меню – строки на которую указывает курсор (яркая желтая полоска). Передвигать курсор можно с помощью клавиш «↑» (**UP**) и «↓» (**DOWN**). Во-вторых, можно просто нажать на клавишу, которая написана в начале строки меню. Последний способ предпочтительнее, так как занимает гораздо меньше времени, но пользоваться им рекомендуется после ознакомления со всеми функциями монитора.

Нажимать на кнопку **MAGIC** необходимо очень резко, в противном случае может возникнуть «двойной» вход в монитор – ситуация при которой дальнейшее выполнение программы становится невозможным.

Некоторые команды выполняются сразу, некоторые – вызывают выполнение новых меню, так называемых *подменю*. В любой момент можно перейти в главное меню (**Main menu**), нажав на клавишу **EDIT**.

Не все команды могут быть выполнены. Некоторые из-за установленной конфигурации и режимов, некоторые просто не реализованы в данной версии монитора. Пункты таких команд написаны в меню блеклыми белыми буквами.

Задествовав кнопку **MAGIC** для входа в монитор, мы «украли» ее у TR-DOS. Однако выполнить функцию, которую она обеспечивала (сброс на дискету текущего состояния программы) можно, выбрав пункт **Magic button** (см. стр. 8).

Функционально в меню сервис-монитора можно выделить команды четырех типов: входа в отладчик, работы с принтером, работы с диском, установки режимов. Однако сгруппированы команды не по типам, а по значимости – чем реже используется та или иная команда, тем «глубже» меню, в котором она находится. При описании команд сервис монитора мы будем указывать «путь» к команде – разделенные косой чертой названия тех меню, которые необходимо выбрать для выполнения команды. Например, для распечатки копии экрана на принтере нужно выбрать **Print screen/Printer**.

Ниже дан примерный порядок работы с командами сервис-монитора. Более полная информация содержится в разделе «1.6. Меню сервис-монитора» на стр. 8. Для быстрого нахождения страницы описания того или иного пункта удобно использовать приложение Ж.

1.2. Работа с принтером

Прежде чем что-либо распечатывать нужно убедиться в правильной установке характеристик принтерного интерфейса и самого принтера. От этого зависит правильная печать, как из монитора, так и из Бейсика. По умолчанию выбран принтер типа *Epson* с параллельным интерфейсом. Все параметры печати сведены в одно меню *Set Up/Printer*, см. стр. 11 (ниже мы опустим этот префикс для краткости).

Обратите внимание на тип используемого интерфейса (*Interface*). По умолчанию используется наиболее широко распространенный параллельный интерфейс типа *Centronics*. Если Ваш принтер использует последовательный интерфейс *RS-232*, то необходимо правильно выбрать его характеристики: скорость обмена (*Speed*), число битов данных (*Data bits*) и число битов остановки (*Stop bits*). Эти параметры должны соответствовать аналогичным в самом принтере.

Если Ваш принтер снабжен параллельным интерфейсом *ИР-ПР*, то, возможно придется проинвертировать выдаваемые данные (*Data*).

Остальные параметры меню *Set Up/Printer* обеспечивают «логическую» организацию печати. Пункт *String width* задает ширину печатной строки в символах. После вывода данного количества в символах автоматическим будет создана новая строка, то есть выданы два управляющих символа – «возврат каретки» (*Carriage return*) и «перевод строки» (*Line feed*).

Пункт *Page length* задает размер страницы в строках. После вывода заданного числа строк на принтер посылается управляющий символ «перевод формата» (*From feed*).

Для установки счетчиков символов и строк в исходное положение служит пункт *Reset*.

Автоматическое разбиение на строки и страницы, а также вывод управляющих символов можно выключить, задав нулевые значения в соответствующих пунктах меню.

Коды управляющих символов должны соответствовать таким же в принтере. При печати из Бейсика код 13 (перевод строки) будет заменен соответствующими управляющими символами. Обратите внимание еще на два момента, связанные с печатью из Бейсика.

Во-первых, интерпритатор Бейсика сам может подсчитывать длину распечатываемой строки. За это отвечают две системные переменные *COL* (23395) – позиция печати и *WIDTH* (23396) – ширина строки. По умолчанию ширина строки здесь задана равной 80 символам. При печати будет действовать тот ограничитель (из сервис-монитора или из Бейсика), размер которого меньше.

Во-вторых, при использовании команды *COPY* необходимо выключить автоматическое разбиение на строки и страницы, иначе на принтер будут выдаваться лишние коды. Печать копии экрана из монитора этого не требует.

Распечатывание копии экрана на принтере из монитора (*Print screen/Printer*, см. стр. 9) может быть произведена как в прямом (*Positive*), так и в инвертном (*Negative*) виде. Управляет этим пункт *Set Up/Print screen* (см. стр. 11).

1.3. Работа с диском

Как и TR-DOS сервис-монитор позволяет использовать практически любые типы дисководов и дискет. По умолчанию монитор настроен на самый распространенный тип дисковода: двухсторонний 80-ти дорожечный, но можно выбрать и другие типы. Все управление сведено в одно меню **Set Up/Disk utility** (см. стр. 13). Сразу отметим, что эти параметры используются только при дисковых операциях сервис-монитора и ни как не связаны с TR-DOSom.

Прежде всего необходимо задать номер устройства, для которого будут изменяться режимы. Сделать это можно, выбрав пункт **Drive**. Параметры дисковода задаются в трех смежных строках (см. стр. 13): **Track** - число дорожек, **Side** - число сторон и **Speed** - скорость перемещения головок.

Параметры дискет похожи на параметры дисковода и в общем случае должны им соответствовать. Задаются в трех смежных строках (см. стр. 14): **Track** - число дорожек, **Side** - число строк и **Sector** - размер сектора в байтах. Возможность отдельно задавать параметры для дисководов и дискет дает возможность поддерживать (например, при форматировании) старые типы.

Надо отметить, что монитор не проверяет достоверность выбранных параметров. Так установка 40-дорожечного дисковода и 80-дорожечной дискеты приведет к непредсказуемому эффекту. Однако обратное сочетание может иметь место. Таким образом, параметры дисковода всегда должны отвечать действительности, а параметрами дискеты можно варьировать.

Конкретные дисковые операции сведены в меню **Disk utility** (см. стр. 9). Текущий дисковод указан в пункте **Drive**. Его можно сменить и в **Set Up/Disk utility/Drive**.

Команда **Catalogue disk** (см. стр. 9) выдает каталог дискеты вставленной в соответствующий дисковод. Верхние три строки содержат информацию о дискете в целом. Каждая последующая строка содержит информацию об одном файле из каталога. За один раз выдается информация о 16 файлах, затем вывод приостанавливается. Для продолжения вывода нужно нажать на клавишу **ENTER**. Конец каталога отмечается словом **END**.

Команды **Format disk** и **Test disk** (см. стр. 10) используются соответственно для форматирования и проверки дискет. Типы дискет для этих команд должны быть заданы в меню **Set Up/Disk utility**. Параметр **Interleave** (см. стр. 13) задает величину чередования секторов при форматировании, тем самым определяя «скорость» работы дисков. Количество попыток считывания сектора при проверке диска выбирается в **Retry constant** (см. стр. 14).

1.4. Установка режимов

Кроме рассмотренных ранее режимов работы печати и дисковых операций, в меню **Set Up** предусмотрен выбор и изменение некоторых других режимов.

Два подменю **Keyboard monitor** и **Keyboard computer** (см. стр. 12) предназначены для изменения временных параметров клавиатуры соответственно для сервис-монитора и для интерпретатора Бейсика.

Подменю **Monitor** (см. стр. 13) устанавливает режимы работы монитора-отладчика. Для этих же целей используется **Set Up/Reset Monitor** (см. стр. 14).

1.5. Вход в отладчик

Для входа в монитор-отладчик служит пункт **Monitor** (см. стр. 8) главного меню. Описанию команд данного режима посвящена вся вторая часть этой книги.

1.6. Меню сервис-монитора

Строки меню приведены в том порядке, в котором они заданы в программе монитора. Содержание подменю приводится непосредственно за пунктом, его вызвавшим. Для удобства справа указано, из какого меню взят описываемый пункт.

O. Continue program

Main menu

Выполнение этого пункта меню приведет к продолжению работы прерванной программы. Следует отметить, что не все программы могут быть продолжены, см. приложение Б. «Технические подробности» на стр. 40.

G. Magic button

Main menu

Эта функция делает то же, что раньше делала кнопка **MAGIC BUTTON** — сохраняет прерванную программу на дискете. В последствии такую программу можно запустить оператором **GO TO** дисковой операционной системы.

По некоторым причинам функция **Magic button** иногда оказывается заштрихованной (отмеченной белым цветом). В этом случае нужно попытаться остановить программу в другом месте, предварительно продолжив ее выполнение функцией **Continue program**.

Сохранение программы «Волшебной» кнопкой возможно только тогда, когда подключен 1 банк ПЗУ (**ZX Spectrum 48**). Когда работают прикладные программы (игры в том числе) это условие в подавляющем большинстве случаев выполняется. Однако может случиться так, что монитор вызван как раз в момент установки 0 банка ПЗУ (**ZX Spectrum 128**)^{*}, в этом случае сохранить программу «Волшебной» кнопкой не удастся.

M. Monitor

Main menu

Выбор этой функции установит режим монитора, в котором предоставляется возможность полного контроля над прерванной программой: можно посмотреть и изменить содержимое памяти и регистров, проанализировать, где произошел останов программы, реассемблировать часть программы для изучения ее устройства.

Режим предназначен в основном для отладки своих программ, однако может быть использован и для «взлома» чужих, с целью анализа и изучения программистских приемов. Команды и методы работы в режиме монитора описаны во второй части книги.

Функция может быть отключена при использовании программ, написанных специально для компьютера **Scorpion ZS 256**

^{*} Это происходит, например, при редактировании программы в режиме **128 Бейсика**

P. Print screen**Main menu**

Эта функция предназначена для получения так называемой «твердой копии» экрана. В появившемся меню необходимо выбрать устройство вывода, это может быть принтер (P. Printer) или диск * (D. Disk). Функция запрещена, если в режиме монитора-отладчика отключено сохранение экрана командой Screen (см. стр. 24).

D. Disk utility**Main menu**

Этой функцией вызывается подменю дисковых операций, в которые входит вывод содержимого каталога, форматирование и проверка дискет. См. раздел 1.3. «Работа с диском» на стр. 7.

Любая дисковая операция может быть прервана клавишей BREAK. Во время выполнения дисковых операций могут возникать ошибочные ситуации (например, попытка вывода каталога неформатированного диска). В этом случае операция прерывается, а на экран выдается пояснительное сообщение с номером ошибки. Номера (коды) ошибок приводятся в описании драйвера дисковых операций (см. стр.32)

O. Previous menu**Disk utility**

Возврат к предыдущему меню.

D. Drive**Disk utility**

Выбор дисководов для изменения параметров настройки. Установленный дисковод (A или B) становится текущим рабочим дисководом для сервис-монитора.

C. Catalogue disk**Disk utility**

Вывод на экран каталога дискеты из текущего дисковода. Верхние три строки содержат информацию о дискете в целом. Первая строка выдает имя диска:

Disk «Name»	диск с именем Name.
Следующая строка	информирует о типе дискеты.
40Track	40 дорожек;
80 Track	80 дорожек;
One Side	односторонняя;
Double Side	двухсторонняя.

Здесь следует отметить, что все дискеты системы TR-DOS отформатированные стандартным образом будут показываться как односторонние. Это происходит из-за того, что TR-DOS не маркирует стороны дискеты, а использует для указания типа дискеты специальные байты, расположенные в области каталога. Значение этих байтов показывается за типом дискеты в скобках. Первый байт всегда равен 10 (если только дискета не испорчена). Второй байт указывает тип дискеты:

16	80 дорожечная, двухсторонняя;
17	40 дорожечная, двухсторонняя;
18	80 дорожечная, односторонняя;
19	40 дорожечная, односторонняя.

* Диск реализован, начиная с версии 2.0 prof. В версии 2.0 и более ранних этой функции нет.

Третья строка информирует о файловой структуре. Количество файлов на диске указывается за словом File. Первое число – общее количество файлов, второе – число удаленных файлов. За словом free указывается неиспользованный объем дискетки в секторах. Два шестнадцатичных числа в скобках показывают местоположение первого неиспользованного сектора (номер строки/номер дорожки).

Каждая последующая строка содержит информацию об одном файле из каталога. Сначала указывается имя файла, затем в треугольных скобках – его тип. Имена удаленных файлов помечаются символом треугольничка. Последующие три числа информируют об адресе загрузки файла, его длине* и размере в секторах. Последние два числа в строке указывают местоположение первого сектора файла на диске (номер сектора, номер дорожки).

За один раз на экран выдается информация о 16 файлах, затем вывод приостанавливается. Для продолжения вывода нужно нажать на клавишу ENTER. Конец каталога отмечается словом END.

F. Format disk

Disk utility

Форматирование диска на выбранном дисковом. На появившийся запрос Insert disk, Press Y key необходимо ответить «Y» для начала форматирования, или любую другую клавишу для отмены операции. При форматировании дискетки на экран выдается статистическая информация о ходе выполнения операции. Слева показывается номера дорожки и стороны дискетки, разделенные символом «/». Треугольнички справа обозначают сектора. Цветом треугольничков показывается выполняемая операция и ее результаты. Зеленые треугольнички на голубом фоне отображают процесс записи секторов на дорожку. Затем следует проверка записи и фон изменяется на желтый. После проверки зеленый цвет треугольничков указывает на хорошие сектора, а красный – на плохие. В нижней строке указывается суммарное количество отформатированных секторов (Total) и количество хороших секторов (Gold).

В отличие от TR-DOS сервис-монитор показывает реальное количество секторов на диске. Из-за этого, например, вместо привычного числа 2544 будет высвечено 2560. Напомним, что TR-DOS занимает 0 дорожку (16 секторов) под системные нужды (каталог, magic и др.).

Если дискета форматируется в стандарте TR-DOS (256 байтные сектора), то по завершению форматирования в нижнем служебном экране необходимо ввести имя дискетки. Напомним, что имя должно содержать не более 8-ми символов.

Параметры форматирования должны быть предварительно установлены в меню SetUp подменю Disk utility (см. стр. 13).

T. Test disk

Disk utility

В этом режиме происходит проверка дискетки, путем последовательного считывания всех имеющихся на ней секторов. Процесс проверки полностью аналогичен процессу форматирования дискетки, за исключением того, что производится только считывание информации.

* Эти параметры зависят от типа файла

- S. Set Up** **Main menu**
Это меню служит для установки и изменения различных режимов работы теневого сервис-монитора.
- O. Previous menu** **Set Up**
Возврат к предыдущему меню.
- S. Print screen** **Set Up**
Управляет выходом изображения экрана на принтер. Для изменения режима нужно нажать на клавишу **ENTER**. **Positive** заставляет печатать копию экрана в обычном виде, а **Negative** – в инверсном.
- P. Printer** **Set Up**
Управляет режимами работ. принтера.
- O. Previous menu** **Set up/Printer**
Возврат к предыдущему меню.
- I. Interface** **Set Up/Printer**
Устанавливает тип аппаратного интерфейса принтера: параллельный – Centronics или последовательный – RS232.
- S. Speed** **Set Up/Printer**
Задаёт скорость по последовательному интерфейсу. В мониторе предусмотрены два стандартных значения скорости: 1200 и 9600 бод. Профессиональное расширение монитора позволяет устанавливать произвольную скорость обмена.
- A. Data bits** **Set Up/Printer**
Устанавливает количество информационных бит в посылке данных по последовательному интерфейсу. Выбирается из двух стандартных значений 7 или 8 бит.
- P. Stop bits** **Set Up/Printer**
Устанавливает количество стоповых бит для последовательного интерфейса. Выбирается из двух стандартных значений: 1 или 2 бита.
Параметры Speed, Data bits и Stop bits выбираются соответственно установленным в принтере.
- D. Data** **Set Up/Printer**
Устанавливает режим передачи данных в принтер: positive – данные передаются в прямом виде, negative – в инверсном.
- C. Carriage return** **Set Up/Printer**
Задаёт код, который будет передаваться на принтер в конце строки. Заданное по умолчанию значение 13 является стандартным, принятым в большинстве принтеров, управляющим символом «возврата каретки». Нулевое значение выключает передачу хода.

F. Line feed**Set Up/Printer**

Задается код, который будет передан на принтер вслед за кодом Carriage return. Значение 10, заданное по умолчанию, является символом «перевода строки». Нулевое значение выключает передачу хода.

V. From feed**Set Up/Printer**

Задаёт код, который будет передан на принтер в конце страницы – когда количество выведенных строк превысит параметр Page length (см. ниже). В большинстве принтеров код управляющего символа «перевод формата» равен 12. Нулевое значение выключает передачу хода.

L. String width**Set Up/Printer**

Задаёт количество символов в строке, после достижения которого на принтер будут последовательно выведены коды Carriage return и Line feed. Нулевое значение выключает проверку на длину строки.

W. Page length**Set Up/Printer**

Задаёт количество строк на странице, после достижения которого на принтер будет передан код Form feed. Нулевое значение выключает проверку на длину страницы.

R. Reset**Set Up/Printer**

Инициализирует внутренние переменные монитора, связанные с размером страницы. Рекомендуется выполнять после аппаратного сброса принтера.

K. Keyboard monitor**Set Up**

Устанавливает временные параметры клавиатуры сервис-монитора.

O. Previous menu**Set Up/Keyboard monitor**

Возврат к предыдущему меню.

B. Constant debouncing**Keyboard monitor**

Антидребезговая задержка. Можно изменить в диапазоне от 1 (минимальная) до 10 (максимальная).

D. Delay before repeat**Keyboard monitor**

Задержка перед автоповтором, изменяется от 25 (минимальная) до 254 (максимальная).

R. Delay repeat**Set Up/Keyboard monitor**

Задержка между повторами клавиш, изменяется от 1 (минимальная) до 150 (максимальная).

Следует отметить, что значения этих параметров сильно взаимосвязаны. Например, при одинаковых значениях задержек автоповтора реальное время отклика будет различно при изменении параметра антидребезговой задержки.

C. Keyboard computer**Set Up**

Устанавливает временные параметры клавиатуры компьютера, параметры аналогичны меню Keyboard monitor. Изменение режимов в этом меню изменяет системные переменные компьютера.

M. Monitor**Set Up**

Устанавливает режимы работы сервис-монитора.

O. Previous menu**Set Up/Monitor**

Возврат к предыдущему меню.

R. Reset RAM**Set Up/Monitor**

Включает (on) или выключает (off) режим очистки оперативной памяти при сбросе компьютера кнопкой **RESET**.

Отметим, что хотя в состоянии off физической очистки памяти не происходит, все системные переменные Бейсик-интерпретатора инициализируются. В том числе и отвечающие за электронный диск.

S. Sound**Set Up/Monitor**

Включает (on) или выключает (off) режим сохранения состояния музыкального процессора при входе в монитор. Когда режим включен, при входе в монитор музыкальный процессор будет выключен, а по возвращению в выполняемую программу вновь включен.

D. Disk utility**Set Up**

Устанавливает режим работы с диском. Здесь можно установить характеристики подключенных дисководов и используемых дисков для операций форматирования и проверки дисков (см. также раздел 1.3. «Работа с диском» на стр. 7). Сервис-монитор использует практически все возможности встроенного в Scorpion ZS 256 контроллера дисков 1818BG93.

O. Previous menu**Disk utility**

Возврат к предыдущему меню.

D. Drive**Set Up/Disk utility**

Выбор дисковода для изменения параметров настройки. Установленный дисковод становится текущим рабочим дисководом для сервис-монитора

T. Track**Set Up/Disk utility**

Выбор количества дорожек дисковода. Поддерживаются 40- и 80-ти дорожечные дисководы.

S. Side**Set Up/Disk utility**

Выбор рабочих сторон дисковода. Поддерживаются одно- и двух-дорожечные дисководы.

P. Speed**Set Up/Disk utility**

Выбор скорости перемещения магнитной головки. Скорость может быть выбрана из четырех значений от минимальной (LOW) до максимальной (HIGH).

L. Interleave**Set Up/Disk utility**

Задание параметров чередования секторов при форматировании дисков. Изменив параметры чередования секторов можно физически изменить местоположение секторов на диске, а, следовательно, и время доступа к сектору. Иными словами, этот параметр задает скорость работы с дискетой.

Между считыванием двух последовательно расположенных секторов программа должна произвести обработку полученной информации. Если время обработки велико, то дискета успеет повернуться в дисководе настолько, что следующий сектор окажется пропущенным, и сможет быть обработан только на очередном обороте диска. Сократить время ожидания можно, изменив физическое местоположение секторов на диске, так чтобы следующий сектор располагался не сразу за предшествующим, а через некоторое число секторов. Управляет чередованием секторов параметр **Interleave**. Если он равен 1 секторы располагаются последовательно, если 2 – через один (1,9,2,8...), если 3 – через два и так далее. Заметим, что на скорость работы влияет как чередование секторов, так и время обработки информации в компьютере. Стандартно в TR-DOS секторы расположены через один (**Interleave** = 2), однако, как показывает практика, процессор успевает обрабатывать информацию за один межсекторный интервал, поэтому, часто более удобным оказывается значение **Interleave** равное 1.

R. Track

Set Up/Disk utility

Выбор количества дорожек на диске. Обычно выбирается такое же значение, как и для дисковода.

I. Side

Set Up/Disk utility

Выбор рабочих сторон диска. Обычно выбирается такое же значение, как и для дисковода.

E. Sector

Set Up/Disk utility

Выбор размера сектора на диске. Стандартный размер сектора для дискеты TR-DOS 256 байт. Возможность менять размер сектора (128/256/512/1024 байта) может оказаться полезной для других операционных систем, таких как CP/M или **Micro-DOS**.

Y. Retry constant

Set Up/Disk utility

Установка количества попыток считывания при сбоях. Используется при проверке правильности формата.

R. Reset monitor

Set Up

Устанавливает исходное состояние сервис-монитора. Этой функцией можно «сбросить» монитор, не выключая питания. Отметим, что исходное состояние устанавливается не непосредственно, а только после аппаратного сброса компьютера.

A. About monitor

Main menu

Команда выводит на экран сообщение об авторских правах и номер версии теневого сервис-монитора.

Часть 2. Монитор-Отладчик

2.1. Строеение экрана

Экран теневого монитора разбит на пять функционально законченных подэкранов, каждый из которых предназначен для индикации своих, строго определенных функций. Слева сверху отображается содержимое регистров микропроцессора, «застынутое» в момент входа в теневого монитор. Следом за ними располагаются 4 байта «дампа по регистрам» – содержимое ячеек памяти, адресуемой соответствующим регистром. Для регистра F приводится «раскладка» по флагам. Ниже располагается служебная информация: состояние прерываний (INT – включены ON или выключены OFF), конфигурация памяти компьютера – номера банков ПЗУ (ROM) и ОЗУ (RAM), номер текущего экрана (SCR).

В верхнем окне справа реассемблер текущей (указываемой регистром PC и выделенной желтой полоской) и четырех следующих команд.

Непосредственно под реассемблером команд расположено окно, предназначенное для отображения служебной информации. Сразу после включения в нем отображается состояние монитора (см. команду SHOW на стр. 24).

В нижней части экрана приведены 64 ячейки дампа памяти. Каждая строка дампа начинается с адреса первой в строке ячейки памяти, далее следуют восемь значений в шестнадцатиричном виде, а затем те же восемь значений в виде ASCII символов. В поле шестнадцатиричных символов выводится курсор, состоящий из угловых скобок (> <), показывающий так называемую текущую ячейку памяти. Эта ячейка (mesh) служит исходной точкой для большинства команд монитора. Над дампом памяти справа выдается реассемблер команды из текущей ячейки.

Нижние две строки предназначены для сообщений монитора (см. приложение А «Сообщения монитора» на стр. 37) и ввода команд. Сообщения выводятся на верхней голубой строке, а команды монитора набираются в строчном редакторе в самой нижней строке экрана.

2.2. «Горячие» клавиши

Для увеличения производительности некоторые часто выполняемые команды не вводятся в редакторе, а выполняются простым нажатием на какую-нибудь клавишу. Такие клавиши называются «горячими». Ниже приводится их полный список.

Следующие шесть команд дают возможность перемещаться по дампу памяти:

SS/Q	на один байт влево;
SS/E	на один байт вправо;
SS/W	на одну строку вверх;
SS/S	на одну строку вниз;
SS/A	на следующую команду (если команда не выполняется);

SS/G переход на адрес, содержащийся в команде, расположенный в текущей ячейке памяти. Переход происходит для команд адресующих непосредственные ячейки памяти, например, JP addr, JR addr, CALL addr, LD A (addr).

SS/D Быстрый вход в команду **DASM** (см. стр. 20).

SS/F Продолжение поиска в памяти, см. команды **FIND** и **SEARCH** (на стр. 19).

CS/1 (EDIT) Прерывание выполнения команд монитора. Этой клавишей можно прервать действие таких команд, как **HELP**, **DISS**, **DUMP**.

Следующие программы служат для выполнения команд отлаживаемой программы (подробнее см. раздел 2.4.8. «Трассировка программы» на стр. 23):

SS/3 выполнение программы, указанной регистром PC;

SS/4 выполнение команды с быстрым выполнением подпрограмм;

SS/9 пропуск программы, содержание регистра PC увеличивается на длину текущей команды (сама команда не выполняется).

2.3. Редактор команд

Команды монитора вводятся и редактируются в строчном редакторе внизу экрана. Редактор поддерживает два режима работы – вставки и замены. В режиме вставки набираемые символы вставляются в текст перед курсором – мигающим прямоугольником. В режиме замены символы вводятся на место курсора, который принимает вид мигающего квадрата. Переключается режим вставки/замены клавишей **SS/1**.

Строчные символы вводятся простым нажатием на клавиши, прописные – совместно с клавишей **CS**, либо после фиксации клавишей **CS/2**.

Для перемещения по строке используются клавиши «←», **SS/5** (на символ влево) и «→», **SS/8** (на символ вправо). Редактировать строку можно с помощью клавиши **CS/0 (DELETE)**, которая удалит символ перед курсором.

Ввод команды заканчивается клавишей **ENTER**, при этом копия строки запоминается в специальном буфере – списке команд. Если ошибок нет, то команда выполняется, в противном случае в строке сообщений указывается возможная причина ошибки.

В любой момент можно вызвать на редактирование выполненные ранее команды из списка команд. Перемещение по списку производится клавишами «!», **CS/6** и «!», **CS/7**. Содержимое редактируемой строки (если не была нажата клавиша **ENTER**) при этом теряется.

2.4. Команды монитора

Команды монитора состоят из ключевого слова, за которым могут следовать параметры. Параметры разделяются знаками пробела или символами запятой (,) или точкой с запятой (;). От ключевых слов параметры могут не отделяться. Сами ключевые слова могут вводиться не полностью – при описании команд обязательная часть написана строчными буквами, а

факультативная – прописными. В качестве параметров могут выступать числа, выражения, мнемоники регистров, значения ячеек памяти, строки.

Первым символом числа должна быть цифра или префикс, задающий систему счисления. Если префикс отсутствует, число считается в установленной системе счисления (см. команду **BASE** на стр. 24). В табл. 1 перечислены возможные префиксы.

Префикс	Система счисления
%	Двоичная
@	Восьмеричная
	Десятичная
#	Шестнадцатеричная

Все числа считаются целыми без знака, диапазон представления 0...65535.

Числа в параметрах команд можно складывать и вычитать. Так, например, параметр со значением 8 можно представить простым числом 8, так и выражением 2+6. В мониторе не предусмотрены операции унарного минуса, однако возможна запись числа со знаком минуса. Такая запись интерпретируется как 0 - число.

Наряду с числами можно использовать мнемонические обозначения регистров. Например, параметр **HL** будет интерпретирован, как текущее значение регистра **HL**. В качестве мнемоник могут выступать:

байтовые регистры **A**, **B**, **C**, **D**, **E**, **H**, **L**, **I**, **R**. Альтернативные регистры **A'**, **B'**, **C'**, **D'**, **E'**, **H'**, **L'**. Флаговые регистры **F** и **F'**. Половинки индексных регистров **IXh**, **IYh** – старшие байты регистров **IX** и **IY**, **IXl** и **IYl** – младшие байты; двухбайтовые регистры **AF**, **BC**, **DE**, **HL**, **IX**, **IY**, **SP**, **PC**. Альтернативные регистры **AF'**, **BC'**, **DE'**, **HL'**.

Мнемоники можно вводить как прописными, так и строчными буквами.

Числа и выражения, заключенные в скобки, интерпретируются как значения ячеек памяти, адрес которых задает соответствующее выражение. При этом, если используются круглые скобки, то из памяти выбирается один байт, а если квадратные – то два смежных байта. Например, если значение регистра **IX** равно 30000, в ячейке памяти по адресу 30010 хранится 2, и по адресу 30011 – 256, то **(IX + 10)** будет интерпретировано как 2, а **[ix + 10]** – как 258.

Символы, заключенные в кавычки, интерпретируются как отдельные параметры со значением соответствующего символа **ASCII**. Так, например, «a» интерпретируется как 97. Некоторые команды, например, поиска, могут использовать строки – несколько символов, заключенных в кавычки.

Кроме вышеприведенных параметров допускается использовать еще три. Мнемоника **MEM** интерпретируется, как адрес текущей ячейки, **ON** – как 1 и **OFF** – как 0.

2.4.1. Изменение памяти регистров

В качестве ключевого слова в командах изменения регистров и памяти используется знак равенства (=), слева от которого ставится мнемоника соответствующего регистра, а справа – параметр. Например, для того, чтобы изменить значение регистра **PC** на 1234 необходимо выполнить команду **PC = 1234**. Также изменяется значение в памяти – слева от знака равенства дол-

жен стоять заключенный в скобки адрес. Например, $(\#c120) = 79$. Как и для параметров, различаются круглые и квадратные скобки.

Через знак равенства можно также изменить адрес текущей ячейки памяти, например, `mem = 65000` (см. также команду **MEMORY** на стр. 19).

Кроме вышеприведенного способа изменения ячеек памяти в мониторе предусмотрены следующие команды.

. data

запись в память списка **data** (чисел или стрингов), начиная с текущей ячейки памяти. После выполнения команды указатель `mem` устанавливается за последней измененной ячейкой.

.PC data

запись в память списка **data** (чисел или стрингов), начиная с адреса содержащегося в регистре **PC**. Содержимое самого регистра **PC** не изменяется.

DATA addr, data

запись списка **data**, начиная с адреса **addr**.

FILL .addr_st, addr_end, byte [,bank]

заполнение области памяти с адресами **addr_st** по адрес **addr_end** включительно байтом **byte**. Не обязательный параметр **bank** может указывать, какой банк ОЗУ включить на время выполнения команды.

PUSH data

занесение в стек программы значения **data**.

POP

извлечение из стека значения, практически эта команда аналогична команде $SP = SP + 2$, с той лишь разницей, что извлекаемое из стека значение индицируется в служебном окне.

2.4.2. Управление памятью

Следующие две команды предназначены для изменения конфигурации компьютера.

RAM bank

изменение текущего банка ОЗУ. Параметр **bank** может принимать значения от 0 до 15, а конфигурации 48 Бейсик только 0 или 8. Монитором не поддерживается установка 2-го банка памяти (см. приложение Б).

ROM bank

изменение текущего банка ПЗУ. Параметр **bank** может принимать значения 0 (128 Spectrum) или 1 (48 Spectrum). В конфигурации Бейсик команда запрещена.

MOVE addr_st1, addr_fin1, addr_st2 [,bank]

копирование первой области памяти, расположенной с адреса **addr-st1** по адрес **addr-fin1** (включительно) во вторую область памяти, расположенную с адреса **addr-st2**. Первая область располагается в текущей конфигу-

2.4.3. Поиск в памяти

FIND *addr_st, addr_end, data*

поиск списка байтовых значений **data** в памяти. Поиск ведется с адреса **addr_st** по адрес **addr_end** включительно. Количество элементов в списке **data** не должно превышать 20. Если поиск закончился успешно, в служебной строке указывается адрес первого найденного байта (**Found at addr**), а указатель текущей ячейки перемещается на этот адрес. При неуспешном поиске выдается соответствующее сообщение.

SEarch *addr_st, addr_end, bank, data*

поиск списка байтовых значений **data** в памяти. На время выполнения команды устанавливается банк ОЗУ **bank**. Поиск ведется с адреса **addr_st** по адрес **addr_end** включительно. Количество элементов в списке **data** не должно превышать 20. Если поиск закончился успешно, в служебной строке указывается адрес первого найденного байта (**Found at addr**). При неуспешном поиске выдается соответствующее сообщение.

Для продолжения поиска в указанных командами **FIND** и **SEARCH** границах можно воспользоваться клавишей **SS/F**.

CHeck *addr_st1, addr_fin1, addr_st2 [,bank]*

сравнение двух областей памяти. Область, заданная параметрами **addr_st1** — адрес начала и **addr_fin1** — адрес конца (включительно) сравнивается с областью памяти, начинающейся с адреса **addr_st2**. Параметр **bank** задает номер страницы ОЗУ, из которой выбираются байты второй области, если этот параметр не указан, сравнение происходит в текущей конфигурации памяти. Если области памяти одинаковы (сравнение происходит «байт в байт»), то появляется сообщение **Blocks identical**. В противном случае на экран выдаются адреса и содержимое различных ячеек памяти.

2.4.4. Реассемблирование и индикация памяти

MEMory *addr*

установка адреса текущей ячейки памяти **mem**.

DUmP [*addr_st[,addr_end]*]

выдача дампа памяти с адреса **addr-st** по **addr-end** включительно. Если параметры не указаны, то предполагается, что начальный адрес равен указателю текущей ячейки памяти (**mem**), а конечный — 65535.

LDUmP [*addr_st[,addr_end]*]

выдача дампа памяти на принтер. Параметры команды такие же, как у **DUMP**.

DISS [*addr_st[,addr_end]*]

реассемблирование блока памяти с адреса **addr-st** по **addr-end** включительно. Если параметры не указаны, то предполагается, что начальный адрес равен указателю текущей ячейки памяти (**MEM**), а конечный — 65535.

LDISs [addr_st[,addr_end]]

реассемблирование блока памяти на принтер. Параметры команды такие же, как у DISS.

Выводимую командами DUMP и DISS информацию можно остановить, нажав на клавишу пробела, продолжить вывод – нажав на любую клавишу.

DASm [addr]

реассемблирование памяти в режиме диалога. Параметр addr задает начальный адрес реассемблера, если он опущен, предполагается адрес текущей ячейки памяти. В команду DASM можно быстро войти по клавише SS/D.

На экран выдается реассемблированный текст команд, при этом можно пользоваться следующими клавишами (подкомандами):

SS/D и **EDIT** выход из команды DASM.

SS/5 и **SS/8** изменение адреса первой реассемблируемой ячейки на единицу. Подкоманда **SS/5** переводит указатель на предыдущую, а **SS/8** – на последующую ячейку памяти.

SS/6 и **SS/7** переход к следующей (**SS/7**) или предыдущей (**SS/6**) реассемблируемой инструкции.

ENTER ввод нового адреса реассемблера. Адрес задается в текущей системе счисления (см. команду BASE на стр.24). На режим ввода адреса влияет 4 бит команды **OPTION** (см. стр.24).

SS/G переход на адрес, содержащийся в первой (верхней) реассемблированной команде. Переход происходит для команд, адресуемых непосредственные ячейки памяти, например, JP addr, CALL addr, LD A,(addr).

0...9 запись (**0...4**)/восстановление (**5...9**) начального адреса реассемблера.

b установка ловушки № 8 на первую (верхнюю) команду.

p снятие ловушки № 8.

Несколько замечаний. Устанавливаемый в команде DASM адрес действителен только в течении очередного сеанса работы и не влияет ни на какие адреса монитора. Подкоманда **SS/6** (переход к предыдущей инструкции) не всегда правильно синхронизируется, из-за этого может возникнуть ситуация циклического изменения этой подкомандой одного и того же адреса. Для синхронизации в этом случае удобно использовать подкоманды **SS/5** и **SS/8**, которые изменяют адрес реассемблера на один байт.

Команда DASM имеет 5 ячеек памяти для реассемблера. При нажатии клавиш **0...4** в этих ячейках запоминается адрес первой (верхней) реассемблированной инструкции. В последствии можно восстановить запомненный адрес, нажав на одну из клавиш **5...9**. Соответствия клавиш приведены в табл. 2.

Кроме этого, подкоманда **SS/G** (переход на адрес) запоминает текущий адрес реассемблера в 0 ячейки памяти (для восстановления используется клавиша S).

Табл.2. Соответствие клавиш команды DASM

Для запоминания	0	1	2	3	4
Для восстановления	5	6	7	8	9

В реассемблере команды DASM указываются установленные ловушки маркируются красным цветом.

Одна из ловушек (с номером 8) специально предназначена для использования из команды **DASM**. В зависимости от текущего адреса клавишей **b** устанавливается программная, либо аппаратная ловушка однократного действия.

2.4.5. Работа с внешними устройствами

Для отладки внешних устройств предназначены две команды:

IN addr

чтение байта из порта ввода/вывода. Принятый байт отображается в служебном окне. Полный 16-ти разрядный адрес задается параметром **addr**.

OUT addr, data

Запись байта данных **data** в порт ввода/вывода. Полный 16-ти разрядный адрес задается параметром **addr**.

2.4.6. Ловушки (точки останова)

При выполнении или трассировке программы монитор предоставляет возможность использовать так называемые ловушки (точки останова). Ловушка устанавливается на конкретный адрес в памяти компьютера. В случае передачи управления на такой адрес ход выполнения программы прерывается и управление получает монитор. Продолжить выполнение прерванной программы можно обычными способами (командой **JUMP**, см. стр. 22, трассировкой и другими).

BREak number, type, addr, int_count[,bank]

установка ловушки на адрес **addr**.

Монитор позволяет установить до восьми ловушек (1...8), номер ловушки задается параметром **number**. Если ловушка с таким номером уже установлена, то информация о ней будет просто переписана.

Параметр **int_count** задает количество проходов по ловушке до ее активизации. Например, значение 1 (или 0) приведет к немедленному прерыванию программы, а значение 2 – к прерыванию программы в тот момент, когда ловушке управления будет передано второй раз.

Существует два ловушек: *физические* и *логические*. Физические ловушки могут быть установлены только в оперативной памяти, при этом они изменяют содержимое той ячейки памяти, на которую они установлены. Вместо записанного в ячейку кода монитор устанавливает байт **0cfh** – код команды **RST 8**. При этой команде монитор получит управление и прервет выполняемую программу. Изменение памяти происходит при выходе из монитора, затем, при возврате в монитор, содержимое ячейки будет восстановлено. Логические ловушки не модифицируют ячеек памяти и поэтому могут быть установлены как в оперативной, так и в постоянной памяти. Эти ловушки действуют только при трассировке программы средствами монитора (см. раздел 2.4.8. «Трассировка программы на стр. 23»).

Монитор поддерживает однократный и многократный режим работы ловушек. При однократном режиме ловушка активна до момента ее срабатывания и затем выключается. В многократном режиме ловушка остается активной и после выполнения.

Определяет тип и режим ловушек параметр **type** (см. табл. 3).

Параметр **bank** задает номер банка оперативной памяти, в котором будет установлена ловушка, если этот параметр опущен, то установка происходит в текущем банке.

BRK number, [on|off]

включение (on)/выключение (off) ловушки с номером **number**. Предварительно ловушка должна быть определена командой **BREAK**.

Выключенная физическая ловушка не устанавливается в память.

DELBreak number

удаление информации о ловушке с номером **number**. Эта команда аналогична команде **BRK number, OFF**, с той лишь разницей, что информация о ловушке полностью теряется. Пользоваться ей имеет смысл только в профессиональной версии монитора.

LISTBreak | LBreak

вывод на экран информации о ловушках. Информация выдается в виде таблицы (см. табл. 4)

Табл.3. Типы ловушек

Тип	Определение
0	Физическая, однократного действия
1	Физическая, активная после выполнения
2	Логическая, однократного действия
3	Логическая, активная после выполнения

Табл.4 Информация о ловушке

Имя	Значение
№	Номер ловушки
addr	Адрес, на который установлена ловушка
bank	Номер банка, для которого ловушка установлена
type	Тип ловушки
init	Начальное значение счетчика проходов по ловушке
count	Текущее значение счетчика проходов
word	Указывает на наличие (+) или отсутствие (-) программы для анализатора выполняемая только в профессиональном расширении

2.4.7. Выполнение программы

Jump [[addr],bank]

передача управления программе, параметр **bank** определяет номер банка ОЗУ при выполнении команды, если он пропущен, то остается текущий банк. Параметр **addr** задает адрес, куда будет передано управление, если он опущен, берется значение **PC**.

CALL [[addr],bank]

аналогично команде **JUMP** с той разницей, что в стек исполняемой программы записывается адрес программы возврата в монитор. Таким образом, исполняемая программа выполняется, как подпрограмма относительно монитора.

2.4.8. Трассировка программы

Трассировка программы заключается в пошаговом выполнении команд отлаживаемой программы полностью под управлением монитора. Таким образом, монитор в режиме трассировки не выполняет команды программы в прямом смысле слова, а эмулирует их выполнение. Все команды трассировки выполняют инструкции указываемые регистром РС.

В режиме ручной трассировки используются следующие «быстрые клавиши»:

SS/3 выполнение одной команды отлаживаемой программы;
 SS/4 выполнение одной команды отлаживаемой программы с быстрым выполнением подпрограмм. Подпрограммы (команды ассемблера CALL и RST) не транслируются этой командой, а непосредственно выполняются;
 SS/9 пропуск одной команды.

SLOW type

автоматическая трассировка. Режимы трассировки, задаваемые параметром type, определяются его отдельными битами (см. табл.5). Варьируя значениями битов можно добиться оптимального соотношения время/информация – пожертвовав не используемыми значениями, увеличив скорость трассировки*.

Табл .5. Режимы трассировки

Бит	Значение	
7	0	Трассировать все команды
	1	Не трассировать подпрограммы (автоматическое выполнение подпрограмм)
6	1	Не выводить реассемблер памяти
5	1	Не обновлять реассемблер памяти по содержимому чейки test
4	1	Не обновлять дампы памяти
3	1	Не обновлять дампы по содержимому регистров
2	0	Обновлять содержимое экрана монитора согласно битам 0,1,3...6
	1	Не обновлять экран монитора
1	1	Не обновлять содержимое регистров
0	0	Очищать экран монитора при обновлении информации
	1	Новую информацию записывать поверх старой

Необходимо отметить, что на скорость трассировки влияют еще несколько факторов. При выполнении каждой трассируемой команды область экрана (4000h...5affh) два раза переписывается в специальный буфер монитора. На это тратится довольно значительное время, если содержимое экранной области не играет роли для отлаживаемой программы, то имеет смысл отключить его сохранение (см. команду SCREEN на стр.24). Другим спосо

*) Максимальная скорость достигается в режиме 84h

SCREEN
 SLOW (7) } хорошее соотношение

бом увеличения скорости трассировки является прямое выполнение подпрограмм, расположенной в постоянной памяти (см. команду **OPTION** на стр.24). Эти подпрограммы практически не содержат ошибок и «поведение» их заранее известно.

2.4.9. Работа с экраном

SCReen [on|off|5|7]

управление экраном. Команда **Screen** без параметров покажет сохраненный экран, нажатие на любую клавишу вернет управление в режим монитора. Команда **Screen** с параметром включает (**on**) или выключает (**off**) режим сохранения экрана в специальном буфере монитора, точнее, сохраняются первые 6912 байт из 5 банка ОЗУ. Параметры команды 5 и 7 устанавливают соответственно в 5 или 7 банках ОЗУ.

К сожалению в компьютере **Scorpion ZS 256** (как, впрочем, и **ZX Spectrum 128**) не предусмотрено аппаратной проверки на номер используемого экрана. Поэтому монитор устанавливает тот или иной экран, исходя из содержимого ячейки **BANKM**, расположенной по адресу 23388 (копия системного регистра, поддерживаемая интерпритатором Бейсик-128).

2.4.10. Выход из монитора

Выход из монитора возможен по командам **JUMP** и **CALL** (см. раздел 2.4.7. «Выполнение программы» на стр.22).

EXit

возврат управления в систему меню теневого сервис-монитора.

2.4.11. Режимы монитора

Часть режимов работы монитора устанавливается через меню (см. меню «**SetUp/Monitor**» на стр.13). Кроме этого, используются следующие команды.

OPTion bit, on|off

задание некоторых режимов монитора. Включая (**on**) или отключая (**off**) отдельные биты этой команды (**bit**), можно изменить те или иные режимы работы монитора (см. табл.6).

WORKspace addr

задание адреса начала свободного буфера для команд трассировки. Объем буфера 8 байт. Располагаться буфер может в любом месте оперативной памяти в диапазоне адресов 23296...49143.

BASE 0|2|8|10|16

установка основания системы счисления, используемой при вводе чисел без префикса. По умолчанию (параметр 0) используется десятичная система.

SHow

вывод информации о состоянии монитора (см. табл.7).

2.4.12. Прочие команды

INTerrupt onoff

управление прерываниями. Эта команда разрешает (on) или запрещает (off) прерывание в программе, действуя аналогично командам микропроцессора EI и DI.

EX AF

смена индикации регистров AF на альтернативные.

Команды EX AF и EXX (см. ниже) влияют только на индикацию значений регистров в мониторе, реального переключения наборов не происходит.

EXX

смена индикации регистров BC, DE и HL на альтернативные.

Табл. 6. Состояние битов команды **OPTION**

Бит	Состояние ON (1)	Состояние OFF (0)
6	Начальный адрес команд SS/D берется из регистра PC	Начальный адрес для команд SS/D берется из указателя текущей ячейки (mem)
5	Запретить автоматическое выполнение команды SHOW	Разрешить автоматическое выполнение команды SHOW
4	Вводить адрес в команде DASM числом	Вводить адрес в команде DASM выражением
3	Не трассировать (выполнять сразу) подпрограммы, расположенные в ПЗУ	Трассировать ПЗУ
2	Не выводить расширенный реассемблер (реассемблирована будет только текущая команда)	Кроме реассемблера текущей команды будет выведена информация о четырех последующих
1	Расширить вывод реассемблера в командах DISS , LDISS и DASM , кроме адреса и мнемоники команды, будут представлены значения байт и их эквиваленты в ASCII	Выводятся только адреса и мнемоники команд
0	Выводить адреса переходов в реассемблере в десятичном виде	Выводить адреса переходов в реассемблере в шестнадцатичном виде

Табл. 7. Поля команды **SHOW**

Ключ	Значение
Analyser	Состояние режима анализатора (только для профессионального расширения)
Option	Текущее состояние битов команды OPTION
Workspace	Адрес начала буфера трассировки
Base	Основание системы счисления
Int mode	Установленный режим обработки маскируемых прерываний

? data

вывод параметра **data** в различных командах счисления. Перед числами ставятся соответствующие системам префиксы, см. табл.1. Этой командой удобно пользоваться, как калькулятором.

CATalogue [num_drive]

вывод на экран информации из каталога дискеты, вставленной в дискеточный привод по умолчанию (подробнее см. стр.9). Можно указать номер дисковода: 0 для A, 1 для B.

LCATalogue [num_drive]

вывод информации из каталога дискеты на принтер, параметры аналогичны команде **CATALOGUE**.

Help

показ в служебном окне краткой сводки команд монитора. Число после команды означает количество возможных параметров, знак - что параметром может быть меньше. Слово **complex** указывает на сложный формат параметров.

Часть 3. Расширение ОС Бейсик

Расположенная в ПЗУ программа сервис-монитора позволяет не только отлаживать свои собственные программы, но и предоставляет программисту целый ряд функционально законченных подпрограмм. Вызов подпрограмм из монитора осуществляется через команду RST 8, за которой должно следовать число (байт), указывающее код подпрограммы (см. табл.8).

Табл.8. Коды подпрограмм монитора

Код	Подпрограмма
80h	Вывод символа из аккумулятора на принтер
81h	Вызов функций встроенной дисковой BIOS
82h	Установка некоторых переменных монитора
83h	Очистка экрана пользователя
84h	Вывод символа на экран пользователя
85h	Отключение монитора-отладчика

3.1. Вывод на принтер

Код символа, выводимого на принтер, помещается в регистр аккумулятора. Если в процессе выполнения операции произошла ошибка (нажата клавиша BREAK), то при возврате управления вызывающей программе будет установлен флаг CY, в противном случае флаг сбрасывается.

LD a,13
RST 8
DEFB #80
JR C, ERROR

3.2. Вызов дисковой BIOS (81h)

Применяя средства интерпретатора системных функций операционной системы TR-DOS, не всегда удастся добиться желаемых результатов. Жестко заданный формат физического диска не дает возможности переносить данные между ZX Spectrum и другими компьютерами, такими как Robotron или IBM.

Встроенный в компьютер Scorpion ZS 256 драйвер дисковых операций (дисковый BIOS) позволяет в какой-то мере преодолеть недостатки ОС TR-DOS. Пользуясь драйвером, необходимо отдавать себе отчет в том, что программы не будут работать на других компьютерах.

Драйвер не использует системные переменные TR-DOS. Работа драйвера происходит автономно, он применяет свой собственный стек. В драйвере используются или применяются значения всех регистров, поэтому их значения необходимо предварительно сохранить. Режим прерываний не влияет на работу драйвера.

Неприятной особенностью является то обстоятельство, что все пересылки между диском и памятью ограничены 2 и 3 банками памяти процессора (ад-

реса 16384...49151 включительно). Несоблюдение этого условия приведет к разрушению информации в памяти сервис-монитора.

Как и интерпритатор системных функций, драйвер разбит на несколько функционально законченных подпрограмм, доступ к которым осуществляется через единую точку входа. Номер подпрограммы (код функции) передается в регистре С, параметры передаются через другие регистры, либо через память. Подпрограмма 0...6 написаны Н. Ю. Родионовым. Ниже приведено полное описание функций драйвера дисковых операций. В заголовке каждой подпрограммы указаны входные и выходные параметры.

0 (RESETC)

Сброс контроллера (и установка магнитных головок выбранного дисководов в начальное положение). Работу с дисководом рекомендуется начинать с этой функции.

1 (CHDRV)

вх: A[0...3] – номер дисковода.

Выбор текущего дисковода, все последующие операции будут происходить с выбранным дисководом. Физически дисковод выбирается при выполнении функций 2, 5...8, 10...14. При первом включении компьютера оба дисковода получают фиксированную спецификацию (которую можно посмотреть/изменить в меню сервис-монитора Setup/Disk utility, см. стр.13). Если коду спецификации диска присвоить значение 0fh (см. раздел 3.3.1. «Системные переменные монитора», стр.33), то функции 2, 5...8, кроме того, могут определять спецификацию дисковода при первом к нему обращении. Скорость перемещения головки дисковода не определяется и устанавливается максимальной (03h).

2 (MPOSIT)

вх: D[0...x] – номер логического трека.

Позиционирование на заданный логический трек. При выполнении этой функции читается и проверяется номер дорожки, на которой находится магнитная головка.

3 (IGNORE)

Игнорирование ошибки и продолжение выполнения функции 5 (или 6).

4 (RETRY)

Повторить в случае ошибки функцию 5 (или 6).

5 (READGR)

вх: D[0...x] – номер логического трека (первого обрабатываемого сектора);

E[0...x] – номер первого обрабатываемого сектора;

HL – адрес буфера;

B – количество секторов в операции.

Чтение группы секторов, расположенных подряд, начиная с указанного регистра D и E, в буфер, заданный регистром HL. Количество секторов задается в регистре B. Эта функция аналогична такой же в интерпритаторе системных функций. В случае возникновения ошибки чтения, можно воспользоваться функциями 3 или 4. Функция 4 повторит считывание сбойного

сектора, а выполнение функции игнорирует ошибку, при этом в буфере будет пропущено столько байт, сколько занимает сбойный сектор.

6 (WRITGR)

вх: как у функции 5.

Запись группы секторов, расположенных подряд, начиная с указанного регистра D и E, из буфера, заданного регистром HL. Количество секторов задается в регистре B.

7 (IDENT)

Идентификация диска. По этой команде производится комплексный тест дискеты, в процессе которого проверяется тип записи (частотная модуляция/модифицированная частотная модуляция), количество используемых сторон дискеты (только при неформатированной верхней стороне одностороннего диска), количество рабочих дорожек (40/80) (только, если нет старой разметки на 80), размер сектора (128/256/512/1024 байт). Эти параметры используются в дальнейшем всеми функциями, оперирующими с логическими величинами (2,3...6). Посмотреть и/или изменить код спецификации диска можно также с помощью функции 15.

8 (TR_DOS)

Идентификация диска для ОС TR-DOS, аналогична функции 18h интерпритатора системных функций. Перед вызовом необходимо установить рабочий буфер, размером не менее 256 байт, используя функцию 9.

9 (SETWRK)

вх: HL - адрес рабочего буфера.

Установка рабочего буфера для функции 8.

10. (SCANSEC)

вх: HL - адрес таблицы;

вых: таблица заголовков секторов, находящихся на дорожке

сканирование
дорожки

Форматирование дорожки. Операция выполняется для той дорожки, на которую установлена головка дисковода (либо после функции 2, либо после функции 14). В результате выполнения в памяти строится таблица расположения секторов на дорожке, начиная от индексного маркера. Эта таблица (или аналогичная ей) является входной для последующих функций.

На каждый сектор в таблице отводится 5 байт (см. табл.9).

Байты +1...+4 извлекаются непосредственно из адресных маркеров секторов.

11 (FORMAT)

вх: HL - адрес таблицы секторов;

DE - адрес указателя;

B - должен содержать 20h.

Форматирование дорожки, на которой находится магнитная головка дисковода. Форматирование производится согласно таблице секторов (см. табл.9). Данные из таблицы напрямую переносятся на дискету, никаких проверок на их корректность не производится, заканчивает таблицу идентификационный байт, больший 128. Размер сектора выбирается согласно байту + 4, заполняется сектор константой 0.

После записи очередного сектора в байт по адресу, содержащемуся в регистре DE, записывается число 2ch, а значение DE увеличивается на единицу.

12 (F_READ)

вх: HL – адрес таблицы секторов;

DE – адрес буфера дорожки.

Чтение дорожки. Порядок считывания определяется содержимым таблицы расположения секторов (см. табл.9). Считанные сектора помещаются в буфер дорожки, никаких разделителей между ними не ставится. Сектора со значением идентификационного байта 0FFh или 0FCh не считываются, но в буфере для них пропускается соответствующее длине (байт + 4) место. Если сектор не может быть считан, то соответствующий идентификатор в таблице заменяется на FDh. Идентификатор успешно считанного сектора заменяется на FCh, а по окончании операции все идентификаторы FCh автоматически переводятся в 00h.

Табл.9. Формат таблицы расположения секторов

Номер байта	Значение	
+0	Идентификационный байт	
	00h— информация о секторе достоверна	
	FFh— информация о секторе не достоверна (сбойный сектор)	
	FCh— конец таблицы секторов	
	FDh и FCh— см. функцию 12	
+1	Номер дорожки (физический) [0...x]	
+2	Номер стороны [0/1]	
+3	Номер сектора [1...x]	
+4	Код длины сектора	
	00—128	01—256
	02—512	03—1024

13 (F_WRITE)

вх: HL – адрес таблицы секторов;

DE – адрес буфера дорожки.

Последовательная запись. Порядок считывания определяется содержимым таблицы расположения секторов (см. табл.9). Сектора, для которых значение идентификационного байта 0FFh или 0FCh не записываются, а в буфере пропускается соответствующее длине такого сектора (байт + 4) место. Если сектор не может быть записан, то соответствующий идентификатор в таблице заменяется на 0FDh. По окончании записи все идентификаторы 0FCh автоматически переводятся в 00h.

14 (M_POSF)

вх: D – номер физической дорожки [0...x]

A – номер стороны: [0 – нижняя, 1 – верхняя].

Физическое позиционирование на трек, номер которого задан в регистре D. Не проверяется, на какой дорожке действительно находится магнитная головка дисковода.

15 (ERRES)

вх: HL – адрес таблицы секторов;
 E – заменяемый идентификатор;
 D – устанавливаемый идентификатор.

Замена идентификатора в таблице расположения секторов (см. табл.9).
 Все идентификационные байты, совпадающие с заданным в регистре E заменяются на указанный в регистре D.

16 (DISEXT)

вх: A – спецификация диска (см. табл.10);
 B – режим: 0FFh – установка, другое – чтение.

Чтение/установка спецификации дискеты (для каждого дисководов предусмотрена своя спецификация). Спецификация устанавливается также функцией 7 (и 8).

17 (DRVEXT)

вых: A – код спецификации дисководов (см. табл.11).

Чтение кода спецификации дисководов. Спецификация дисководов определяется и заносится в специальную ячейку внутри драйвера при выполнении теста, который проводится при первом обращении к дисководу функциями 2, 5...8 (либо при первом включении компьютера). Функции 10...14 тест не выполняют.

Табл.10 Байт спецификации диска

Бит	Назначение
0...1	Размер сектора (соответствует байту +4 в таблице размещения секторов)
2	Служебный бит (диск 40 работает на дисковом 80, не сохраняется, но используется—не менять)
3...4	Не используется
5	Количество дорожек (0—40, 1—80)
6	Тип записи (0—FM, 1—MFM)

Табл.11. Байт спецификации дисководов

Бит	Назначение
0...1	Скорость перемещения магнитной головки дисководов
2...5	Не используется
6	Количество дорожек (0—40)
7	Количество магнитных головок (0—одна)

19 (TIMSUB)

вх: B – байт установки time-out;
 вых: HL – адрес процедуры time-out.

Установка режима time-out (тайм аут) для счетчика включения двигателя дисководов.

Формат байта time-out представлен в табл.12.

Функции 2...8, 10...14 в начале работы включают двигатель дисководов, при этом обрабатывается задержка на время раскрутки двигателя. Если для какого-либо дисководов включен режим time-out, то драйвер для последу-

Табл. 12. Формат-байта time-out

Бит	Назначение
0...3	Биты определяют номер дисководов (0 для А, 1 для В)
4...7	Состояние time-out для дисководов. Сброшенный бит соответствует сброшенной функции, а установленный—включенной. 4-й бит соответствует дисководу А, 5-й—дисководу В.

ющих операций в состоянии определить время, в течение которого задержку можно не делать, это существенно сокращает время работы. Для обеспечения режима time-out необходимо каждые 1/50 сек, то есть раз в прерывание вызывать функцию 19 с обнуленным регистром В.

При работе драйвера могут возникать ошибки, связанные с нечитаемостью дискет, несоответствием типов, а также некоторые другие. В этом случае после возврата в программу пользователя будет установлен флаг переноса СУ. Идентифицировать ошибку можно, получив ее код из специальной переменной сервис-монитора (см. раздел 3.3.1. «Системные переменные монитора» на стр.33). Ниже даны коды ошибок драйвера дисковых операций, для специфических ошибок в скобках приведены номера функций, в которых они могут случиться.

- 1 нет готовности (диск не крутится или его нет);
- 2 попытка записи на защищенную дискету (функции 6, 11 и 13);
- 3 во время выполнения функции была нажата клавиша BREAK;
- 4 ошибка при чтении или записи сектора функциями 5 или 6, можно воспользоваться функциями 3 или 4;
- 5 ошибка позиционирования (не соответствует номер дорожки);
- 6 ошибка позиционирования (не считался адресный маркер дорожки);
- 8 ошибка позиционирования, несоответствие типов (попытка использовать 80 дорожечный диск на 40 дорожечном дисководе);
- 9 неформатированная дискета (функции 7, 8);
- 16 дискета не соответствует формату TR-DOS по физическим параметрам: размер сектора, плотность записи, либо нет кода идентификатора – 10h (функция 8);
- 17 несовпадение кода типа дискеты TR-DOS реальному формату;
- 18 не открыт рабочий буфер (функция 8);
- 20 были ошибки при чтении/записи дорожки (функции 12,13).

3.3 Установка переменных монитора (82h)

Через эту подпрограмму программист получает доступ к различным переменным монитора. Это позволяет изменять некоторые режимы работы (это свойство используется в профессиональном расширении), изменять цвета меню и экранов монитора, а также предоставляет возможность доступа к переменным дисковой BIOS.

Перед вызовом этой подпрограммы в регистр DE микропроцессора должен быть помещен код типа переменных (см. ниже), а в регистр BC – смещение к переменной. Бит СУ задает режим доступа, если он установлен – запись переменной, если сброшен – чтение. Значение переменной передается через аккумулятор.

Если необходимо изменить отдельный бит, то сначала нужно прочитать значение переменной, изменить бит, а затем записать измененное значение.

Ниже приведены значения некоторых переменных, представляющих интерес для программистов. После указания кода смещения приводится тип переменной (db – байтовая; dw – двухбайтовая; ds – многобайтовая, последующее число определяет количество байт).

3.3.1. Системные переменные монитора (DE=00)

BC = +12(db)

отдельные биты этого байта позволяют управлять выводом значений регистров на экран монитора (см. табл.13). Запись 1 запрещает вывод.

BC = +21(db)

отдельные биты этого байта (см. табл.14) позволяют управлять выводом на принтер.

Табл.13. Байт управления индикацией регистров

Бит	Регистр
7	PC
6	SP
5	IX
4	IY
3	HL (HL')
2	DE (DE')
1	BC (BC')
0	AF (AF')

Табл.14. Байт управления принтером

Бит	Значение, когда сброшен	Значение, когда установлен
7	Данные выводятся в прямом виде	Данные выводятся в инверсном виде
3	Два стоповых бита	Один стоповый бит
2	8 бит в байте данных	7 бит в байте данных
1	Скорость передачи 9600 бод	Скорость передачи 1200 бод
0	Интерфейс Centronics	Интерфейс RS232

3.3.2. Переменные режимов, цвета (DE=02)

BC = +0(db)

константа антидребезговой задержки монитора.

BC = +1(db)

константа задержки перед автоповоротом.

BC = +2(db)

константа задержки между автоповоротами.

BC = +5(db)

цвет бордюра, устанавливаемый монитором, при подаче звукового сигнала ошибки.

BC = +6(db)

код символа «возврата каретки» для принтера.

BC = +7(db)

код символа «перевода строки» для принтера.

BC = +8(db)

код символа «перевода формата» для принтера.

BC = +9(dw)

код, определяющий скорость передачи для последовательного интерфейса (для 9600 бод – 0bh).

BC = +0bh(dw)

код, определяющий скорость передачи для последовательного интерфейса (для 1200 бод – 6ah).

Bc = +0dh(ds 11)

коды, служащие для перевода принтера в графический режим (непосредственно за ними будет выдано 768 байт графической информации при печати экрана из монитора). Байт +0dh хранит количество управляющих кодов.

BC = +15h(ds 4)

коды, служащие для перевода принтера в текстовый режим, завершающие печать экрана. Байт +015h хранит количество управляющих кодов.

BC = +1dh(db)

число строк на странице при печати на принтере.

BC = +1eh(db)

число символов в строке при печати на принтере.

3.3.3. Переменные дисковой BIOS (DE=04)

BC = +4(ds 4)

байты хранят максимальное число секторов на дорожке при размере сектора 128, 256, 512, и 1024 байта соответственно.

BC = +0ah(db)

номер текущего дисковода: 0 – дисковода А, 1 – В.

BC = +0bh(db)

число попыток считывания при ошибке, начальное значение 3.

BC = +0ch(ds 2)

коды, определяющие тип дисководов (спецификации) А и В соответственно.

BC = +10h(ds 2)

коды, определяющие тип дискеты в дисководах А и В соответственно.

BC = +18h(ds 2)

коды, time-out для дисководов А и В соответственно.

BC = +1dh (db)

код ошибки дискового интерпритатора.

3.4. Очистка окна пользователя (83h)

Для внутреннего использования в мониторе была написана программа оконного интерфейса. Несмотря на более чем скромные возможности, она может оказаться полезной при написании своих собственных программ. Каждое окно описывается своим определенным дескриптором – блоком, состоящим из 13 байт. Местоположение окна на экране может выбираться произвольно, можно также определить несколько окон одновременно, адрес дескриптора текущего окна должен быть помещен в регистр IX микропроцессора. Перекрывающиеся окна не накладываются, а как бы «просвечиваются» сквозь друг друга.

Прежде всего окно необходимо описать, задав в дескрипторе начальные координаты окна и его размер. Все координаты отсчитываются от левого верхнего угла окна, либо экрана. Затем окно необходимо очистить функцией 83h, которая инициализирует некоторые переменные в дескрипторе. Назначение байт в дескрипторе окна приведено ниже

- +0, +1 Координаты курсора в окне (X,Y) – первоначально устанавливаются в ноль программой очистки окна, затем могут модифицироваться пользователем.
- +2, +3 Координаты начала окна (X,Y). Эти координаты измеряются в стандартных знаках размером 8*8 точек.
- +4, +5 Размер окна в стандартных символах (8*8 точек), первое число – размер по вертикали, второе – по горизонтали.
- +6 Байт хранит цифровые атрибуты окна.
- +7 Системные атрибуты окна, задаются отдельными битами (см. табл.15).

Табл.15 Формат байта системных атрибутов окна

Бит	Назначение
5	Запрет вывода, все выводимые символы игнорируются
4	Произвольный формат, по достижению правого края окна будут автоматически выполнены команды перевод строки и возврат каретки (см. функцию 84h, вывод символа в экран пользователя)
3	Выполнять автоматический возврат каретки при приходе символа перевода строки
2	Выполнять автоматический переход строки при приходе символа возврата каретки
1	Изменять атрибуты при выводе каждого символа (в противном случае атрибуты устанавливаются только во время очистки окна)
0	Не выводить символ курсора

- +8, +9 Служебные байты.

- +10 Размер символов в точках. Монитором поддерживается два размера символов в 6 и в 8 точек.
- +11 Размер окна в символах. Для символов в 8 точек этот размер совпадает с шириной окна. Если размер символа 6 точек, то перед инициализацией (очисткой) необходимо рассчитать размер окна и поместить в этот байт.
- +12 Код символа курсора. Код с этим символом будет выводиться вместе, отмечая положение курсора, если сброшен 0 бит в системных атрибутах окна.

3.5. Вывод символа в окно пользователя (84h)

Код выводимого символа должен находиться в аккумуляторе, а дескриптор используемого окна в регистре IX. Символы из набора ASCII (коды в диапазоне 32...127) непосредственно печатаются в определенном окне экрана. Возможно использование определенных символов.

1 Установка атрибутов. Непосредственно за управляющим кодом должен следовать вывод в окно байта атрибутов, текущие атрибуты окна запоминаются программой вывода в специальной ячейке памяти.

2 Восстановление запомненных атрибутов.

7 Звонок. Программа обрабатывает короткий звуковой сигнал. Бернор получает при этом цвет, установленный в специальной ячейке монитора (см. раздел 3.3.2. «Переменные режимов, цвета» на стр.33).

10 Первый строки. Если курсор находится в последней строке, то происходит «свиток» — перемещение содержимого окна вверх с очисткой последней строки.

13 Перевод каретки. Перемещение курсора в крайнюю левую позицию строки в окне. Если установлен соответствующий бит в дескрипторе окна, то совместно с возвратом каретки вызывается и программа обслуживания символа перевода строки.

27 Позиционирование курсора внутри окна. Непосредственно за управляющим кодом должны быть выведены два байта. Первый определяет позицию курсора в строке, второй номер строки. Отсчет ведется от левого верхнего угла окна, начиная с нуля.

3.6. Отключение монитора-отладчика (85h)

Эта функция введена специально для программистов, пишущих для компьютера Scorpion ZS 256, и преследует цель защиты программ от «взлома» внутренним отладчиком. После выполнения этой подпрограммы вызов отладчика из меню сервис-монитора становится невозможным. Кроме этого, включается режим очистки памяти по аппаратному сбросу компьютера, что исключает «снятие» программы из памяти.

Обратим внимание на один важный момент. В «обычном» компьютере использование этой подпрограммы приведет к возврату управления в интерпретатор Бейсик (если только не перехвачен адрес возврата [2]). Такой же эффект произойдет, если в компьютере Scorpion установлена версия монитора ниже 2.2. Подпрограммой 85h можно пользоваться (для отключения отладчика) только, если значение байта 0F1h во втором байке ИЗУ (см. стр.42) отлично от нуля.

Приложения

А. Сообщения монитора

Теневой сервис-монитор обладает большой и сложной системой команд, а, следовательно, достаточно мощным синтаксическим анализатором. В процессе разбора команд и их выполнения могут быть обнаружены неточности и ошибки. В этом случае монитор информирует пользователя и в большинстве случаев предлагает выправить ситуацию. Приведенный ниже список сообщений монитора поможет лучше разобраться в причинах ошибки и исправить ее.

Analyser error Ошибка анализатора (возникает только в профессиональном расширении). Во время работы программы анализатора возникла ошибка, связанная с переполнением стека, делением на ноль или что-то другое. Вслед за сообщением **Analyser error** выдается имя слова, в котором возникла ошибка и, возможно, содержимое стека. Конкретный вид ошибки высвечивается в статусной строке.

Bad address breakpoint Задан неверный адрес ловушки (точнее останова) программы. Для «физической» ловушки адрес не может быть меньше 16384.

Bad disk Ошибка чтения дискеты. Возникает, например, когда сделана попытка посмотреть каталог неформатированной дискеты.

Bad opcode Ошибочный код операции. Во время трассировки программы встретился код, который не соответствует ни одной известной инструкции микропроцессора Z80.

Bad operand Ошибочное значение операнда. Величина операнда команд **SLOW** не может превышать 255.

Bad string Ошибка в задании строкового аргумента (строинга), обычно непарное число кавычек.

Blocks identical Сравнимые (командой **CHECK**) блоки памяти идентичны.

Breakpoint ### При выполнении или трассировке программы встретилась ловушка с номером **###**.

Breakpoint undefined Ловушка не определена. Ошибка возникает при попытке включить (или выключить) ловушку, которая не была ранее задана.

Command abandoned Выполнение команды прервано. Это сообщение информационного характера, команду можно вызвать из стека команд и повторить.

Command not found Команда не найдена. Введенной команды нет в мониторе, либо в тексте присутствует синтаксическая ошибка.

Disk not TRDOS Физические параметры дискеты (размер сектора, например) не соответствуют стандартным для системы TR-DOS.

Division by zero Деление на ноль. Ошибка может возникнуть при выполнении в анализаторе программы пользователя.

Enter address Введите адрес (ввод нового адреса в команде **DASM**, см. стр.20).

Enter name Введите имя (ввод имени дискеты при форматировании).

Enter number Введите число (либо действительно введите число, либо откажитесь от ввода командой **EDIT**).

Enter command Введите команду.

Failed at ### ##:### ## При сравнении блоков памяти обнаружено несоответствие, через двоеточие указаны адреса и значения различных ячеек.

Finish program Конец программы. В процессе трассировки программы выполнена команда микропроцессора **HALT** при отключенных прерываниях. В реальном режиме работы это означает полный останов процессора, выйти из которого можно только аппаратным сбросом.

Forbidden bank RAM Запрещенный номер банка оперативного запоминающего устройства. Ошибка возникает, если в режиме 48 Бейсик попытаться изменить конфигурацию памяти (в этом режиме это сделать невозможно из-за аппаратных блокировок).

Forbidden bank ROM Запрещенный номер страницы постоянного запоминающего устройства. Причины ошибки те же, что и у предыдущей.

Forbidden type word Запрещенный тип слова. Ошибка может возникнуть только в профессиональном расширении, например, при попытке редактирования слова из базового словаря.

Found at ### Найден байт (например, в команде **FIND**), адрес которого выдается в этом сообщении.

Illegal value operand Неправильное значение операнда. Обычно выдается, когда вместо возможных значений **ON** или **OFF** задаются недопустимые численные параметры.

Illegal word name Неправильное имя слова. Ошибка возникает только в профессиональном расширении, например, если дается ссылка к еще неопределенному слову.

Insufficient program space Недостаточно области для программы. Ошибка может возникнуть только в профессиональном расширении, когда не хватает места для размещения в памяти программы пользователя.

Invalid bank RAM Неверный номер банка оперативного запоминающего устройства, который не может быть больше 15.

Invalid bank ROM Неверный номер банка постоянного запоминающего устройства, который не может быть больше 1.

Invalid number breakpoint Неверный номер ловушки. Всего в мониторе можно определить 8 ловушек (1...8).

Invalid status breakpoint Попытка недопустимого действия с ловушкой. Командой **BRK** можно лишь включить (**ON**), либо выключить (**OFF**) ловушку.

Invalid type brackets Несоответствие типов примененных скобок. Закрывающая скобка должна быть того же типа, что и открывающая.

Invalid type breakpoint Неверный тип ловушки. Значение типа может выбираться в диапазоне 0...3.

Known disk Неизвестная дискета. Ошибка возникает в процессе идентификации дискеты.

Known number Непонятный номер. Возможно, попытка ввода числа без префикса, определяющего систему счисления (помните также о системе, установленной командой **BASE**). Эта ошибка может не иметь к вводу чисел никакого отношения, если между параметрами забыт разделитель.

No definition Нет определения для нового слова. Ошибка может быть выдана только в профессиональной версии.

No search string Нет строки для поиска. Аргумент поиска (строка байт) должен быть определен. Клавиша **SS/F** продолжает поиск, начатый командами **FIND** и **SEARCH**, и ищет следующее вхождение аргумента в заданном диапазоне адресов. Если поиск закончен, то аргумент забывается. Для повторного поиска необходимо повторить ввод команды.

Not number Не число. Синтаксический анализатор пытается интерпретировать операнд сначала как мнемонику регистра, затем как выражение, и, наконец, как число, если и это не удастся появляется сообщение об ошибке.

Number too big Число слишком велико. Сообщение выдается, если введено число больше чем 65535.

Operand not found Не найден операнд. В команде задано меньшее количество операндов, чем то, которое должно быть.

Out of range Значение операнда вне возможного для данного случая диапазона.

Out of stack space Недостаточен размер области стека. Сообщение может возникать только в профессиональном расширении.

Print abandoned Прервана команда печати. Если печать прервана, например, нажатием на клавишу **BREAK**, то в некоторых типах принтеров могут остаться распечатанные данные. Для очистки буфера принтера в этом случае его необходимо либо перевести в состояние **OFF LINE**, либо выключить.

Professional program not found Данная функция, команда или режим присутствует только в профессиональном расширении монитора.

Program undefined Программа неопределена. Сообщение выдается только в профессиональной версии монитора, при попытке включить режим анализатора без задания программы анализа.

Quit operand not found Не хватает операнда, обычно это сообщение указывает на нехватку закрывающих скобок.

Stack empty Стек анализатора пуст. Сообщение выдается только в профессиональной версии, если слово пытается обработать значение из стека, которого там нет.

Stack not empty Стек анализатора не пуст. Сообщение выдается только в профессиональной версии, если в конце обработки анализируемого слова на стеке остались неустребованные значения.

Start finish Начальный адрес больше конечного. В командах, работающих с блоками памяти, первым должен указываться начальный (меньший) адрес, а затем конечный (большой).

State of stack Состояние стека анализатора после выполнения слова с помощью команды **EVAL**. Только в профессиональном расширении.

Stop number ### Останов процессора трассировки программы в результате выполнения ключевого слова **st** анализатора. В сообщении выдается номер ключевого слова.

String not found Аргумент поиска (строка символов, байт) не найден.

String too long Аргумент поиска (строка символов, байт) слишком велик (больше допустимых 20 байт).

Syntax error При разборе операнда синтаксический анализатор встретил ошибочное выражение, которое не смог идентифицировать. Ошибка может возникнуть при «слиянии» операндов, когда между ними забыт разделитель.

Too few blocks В команде **MOVE** задан блок нулевой длины.

Too few operands Слишком маленькое значение операнда.

Too many) Слишком много скобок. Сообщение появляется, когда закрывающих скобок больше, чем открывающих.

Too many operands Слишком много операндов для данной команды.

Word not known Неизвестное слово. Сообщение выдается только в профессиональной версии, если при выполнении программы встретилось неопределенное ранее слово.

Word redefined Слово переопределено. Сообщение выдается только в профессиональной версии при задании старому слову новой программы.

Working... Работаю... Сообщение указывает на выполнение команды

Workspace insufficient Задан недопустимый адрес рабочей области монитора. Эта область (используемая при трассировке программ) может располагаться, начиная с адресов 23296...49144.

*****undefined** Данная ловушка не определена.

Б. Технические подробности

Программа монитора полностью занимает банк 2 ПЗУ (**Monitor**), частично 3 (**TR-DOS**) и использует несколько ячеек из 0 и 1 банков. Для своей работы сервис-монитор использует 8 и 9 банки ОЗУ, а также 11 ячеек из стека пользователя (см. табл.16).

Регистр **SP** в мониторе показывает реальное значение стека в прерванной программе, однако 11 байт в стеке ниже **SP** будут утеряны.

Кроме перечисленных ячеек при трассировке в программе пользователя используются 8 ячеек рабочего буфера. Расположение буфера в памяти может быть различным (см. команду **WORKSPACE** на стр. 24), сразу после включения буфер занимает ячейки между адресами 23719...23726 включительно.

Необходимо отметить следующие обстоятельства, которые могут нарушить работу прерванной программы:

1. Если регистр **SP** указывает на область ПЗУ, то прерванную программу не удастся запустить, так как значения, сохраняемые в стеке, будут утеряны. Более того, отсутствие в стеке корректного номера функции приведет к неправильной работе самого монитора.

2. Если в вызываемой программе регистр **SP** используется не для сохранения стека, а для адресации данных, то вызов монитора приведет к разрушению этих данных. Тем самым будет нарушен код программы, и, скорее всего, нормальное продолжение ее выполнения невозможно. Например,

```

DI
LD (ADDRSP),SP
LD SP, DATA
POP BC
POP DE
POP HL
.....
POP IX
LD SP, (ADDRSP)
EI

```

Вызов монитора при выполнении этого фрагмента, скорее всего, испортит данные в области DATA

3. Отсутствие аппаратных средств слежения за изменением содержимого системных регистров не дает 100% гарантии правильного определения конфигурации памяти в момент вызова монитора. К примеру, если программа не использует копию регистра 7ffdh в системной переменной BANKM (23388) номер активного экрана будет определен неверно.

Все вышеприведенное также справедливо и для стандартной программы обработки кнопки **MAGIC** из ПЗУ операционной системы TR-DOS, однако теневой монитор в некоторых случаях может оказаться более «привередливым».

4. Для своей работы сервис-монитор использует 8 и 9 страницы ОЗУ, если информация, содержащаяся в них, будет изменена, то поведение программы не предсказуемо. Для восстановления работоспособности в этом случае необходимо **ВЫКЛЮЧИТЬ ПИТАНИЕ** на 5...10 секунд.

5. Отладчик неправильно работает с адресами 0с000h...0с100h, когда в третьей странице памяти микропроцессора установлен 2-й банк ОЗУ.

Табл.16. Состояние стека при входе в монитор

Смещение относительно стека программы	Размер, байт	Описание
-2	2	Значение регистра PC—адрес возврата из немаскируемого прерывания, формируется аппаратно
-4	2	Значение регистра AF, это и следующие значения заносятся программой входа в монитор, которая расположена в 0 и 1 банках ПЗУ
-6	1	значение регистра R
-7	1	состояние флага IFF2—сохраняется во 2 бите
-8	1	номер функции—используется монитором для определения типа входа
-9	2	значение регистра BC
-11	2	служебная информация, используемая для локализации номера установленного банка ПЗУ

К сожалению, монитор не может самостоятельно отслеживать изменения в используемых им страницах ОЗУ – это отняло бы слишком много времени. Не представляется возможным и проверять всю память после включения питания для определения ее состояния – в процессе работы слишком много ячеек меняют свои значения. Поэтому монитор судит о достоверной информации в памяти по специальному 256-и байтовому полю. Изменение значений ячеек памяти в этом поле приведет к реинсталляции (переустановке) сервис-монитора после «сброса» компьютера.

Для явного задания реинсталляции монитора после «сброса» можно выбрать функцию Set Up/Reset (см. стр.14) в меню.

В. Заметки для программистов

В начале книги говорилось, что для компьютеров Scorpion ZS 256 возможна идентификация каждого конкретного компьютера. Номер компьютера хранится в ПЗУ во 2-ой странице (странице монитора) и занимает 16 байт с адреса 0f0h. Байт 0f1h имеет специальное назначение: в нем хранится номер версии программного обеспечения. Предполагается, что программы могут иметь возможность перестановки на компьютеры с более поздней версией ПЗУ.

Ранее было сказано, что монитор использует для своей работы 8-ю и 9-ю страницы ОЗУ. При нормальной работе программ, предназначенных для 48 К и 128 К памяти, проблем не возникает. Однако в новых разработках могут быть использованы все 256 К памяти, имеющейся в Scorpion'e. Наилучший способ устранения конфликтов с монитором – не использовать указанные банки ОЗУ.

Защита от «взлома» программ может быть реализована при использовании подпрограммы 85h расширенной ОС Бейсик (см. стр.36). Это более предпочтительный вариант, чем приведенный ниже, так как пользователю оставляется возможность работы с диском из сервис-монитора и некоторые другие возможности.*

Если же 8-й и 9-й банки используются, то необходимо принять во внимание некоторые особенности функционирования программного обеспечения компьютера. Отсутствие программы в 8-ом банке приведет к неработоспособности интерпретатора Бейсика. Точнее, той его части, которая обеспечивает обработку ошибок (команда RST 8). Также не будет работать стандартный вывод на печать. И, естественно, перестанет работать сервис-монитор.

Наиболее опасной ситуацией представляется частичное изменение информации в 6-ом банке ОЗУ. При этом может сохраниться частичная работоспособность Бейсика и монитора, но указать, в каком месте может произойти ошибка выполнения и какие действия эта ошибка повлечет, невозможно.

Принимая во внимание все вышесказанное, придется несколько рекомендаций для программ, использующих всю доступную память компьютера.

1. ПЕРЕД работой программ полностью очистите по крайней мере 8

*) Дополнительные сведения Вы можете получить у разработчика монитора по телефону в Санкт-Петербурге 130-19-95 в вечернее время

банк ОЗУ. В этом случае работа монитора становится невозможной, а после сброса компьютера будет произведена реинициализация всех системных ячеек.

2. Работа программы должна заканчиваться либо аппаратным сбросом (имеется в виду то, что пользователь вынужден нажать на кнопку RESET), либо передачей управления на нулевую ячейку при подключении 0-й строки ПЗУ. Это обеспечивает реинициализацию системных переменных монитора.

3. Если программа вышла в 48 режим (48 Бейсик), то ее работа должна завершаться аппаратным сбросом компьютера в любом случае.

Г. Версии

Работа над программой теневого сервис-монитора длится уже более года. За это время было сделано несколько сотен отладочных (Beta) версий. Их номера, выдаваемые функцией About в меню, начинаются с нуля. Привести информацию по каждой из них не представляется возможным. Некоторые функции и команды в этих вариантах содержали ошибки, другие – претерпевали существенные изменения в механизмах работы.

Первая коммерческая версия имеет номер 1 (с различными подвариантами). В ней выявлены ошибки, связанные с неправильной работой процедур печати на принтер. При переводе строки не сбрасывается внутренний счетчик символов, в результате чего появляются дополнительные переводы строк.

Вторая версия (2.0) работает с принтером правильно. Кроме этого, была введена новая команда DASM (см. стр.20). Однако была замечена неприятная ошибка при форматировании диска средствами монитора – неправильно устанавливается номер первого свободного трека. Ошибка исправлена в версии 2.1. Кроме этого, версия 2.1с имеет новый режим в команде DASM – установка/снятие ловушки.

Версия 2.2A снабжена серийным номером разработки (байт 0f1h) и средствами отключения монитора-отладчика. Кроме этого, добавлены средства дополнительного тестирования платы; 6-й бит в команде OPTION.

Работа над программой продолжается. При обнаружении каких-нибудь ошибок и неточностей в работе сервис-монитора, мы будем чрезвычайно признательны Вам за информацию о них. Во избежании непредвиденных ситуаций мы рекомендуем использовать последнюю версию монитора. Переоценка смены версий зависит от количества выявленных ошибок, примерно один раз в полтора месяца.

Д. Литература

1. Зонов С. Ю. Scorpion ZS 256. Руководство пользователя. – СПб., 1993.
2. Ларченко А. А., Родионов Н. Ю. ZX Spectrum для пользователей и программистов. – 2-е изд. СПб., 1993.
3. Диалекты Бейсика для ZX Spectrum/A. В. Болдачев, И. Е. Думов, В. А. Елисеев и др./Под ред. Н. Ю. Родионова и А. А. Ларченко. – СПб.: Питер, 1992.

Е. Карта памяти

На рис.1 представлена карта памяти компьютера Scorpion ZS 256. Следует иметь ввиду, что сервис-монитор допускает использование не всех из аппаратно представленных возможностей. Так, не поддерживается включение в страницу 0 (Page 0) микропроцессора 2-го и 3-го банков ИЗУ и 8-го банка ОЗУ. Имеется ограничение на подключение в страницу 3 микропроцессора 2-го банка ОЗУ.

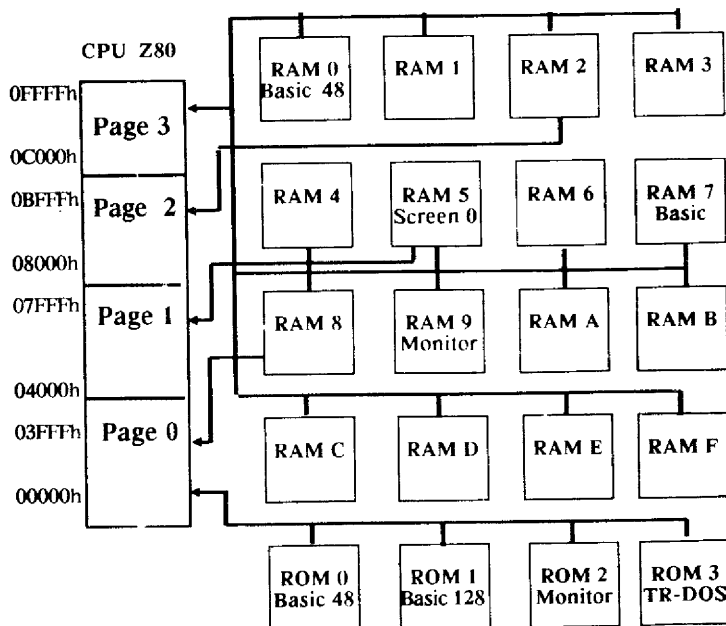


Рис.1. Карта памяти компьютера Scorpion ZS 256

Ж. Индекс

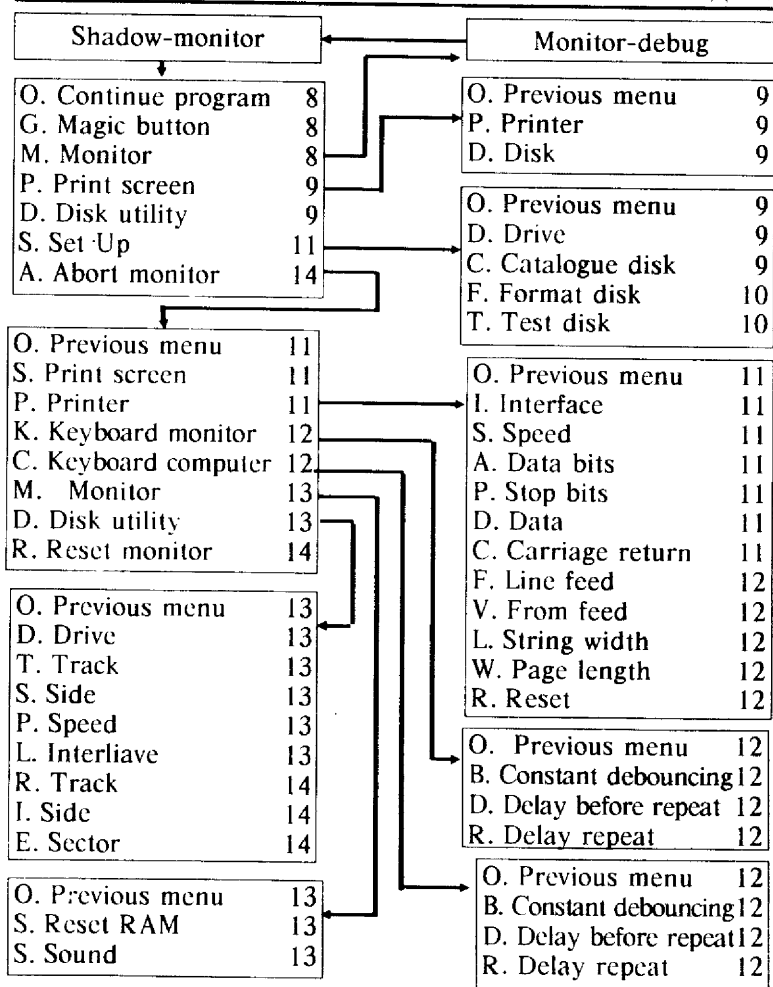


Рис. 2. Карта быстрой ссылки на меню теневого монитора

.data	18
.PC data	18
? data	26
BASE 0 2 8 10 16	24
BREak number, type, addr, int_count[,bank]	21
BRK number,[on off]	22
CAL[][addr],bank]	22
CATalogue [num_drive]	26
CHEck addr_st1, addr_fin1, addr_st2[,bank]	19
DASm [addr]	20
DATA addr, data	11
DELBReak number	22
DISs [addr_st[,addr_end]]	19
DUmP [addr_st[,addr_end]]	19
EX AF	25
EXit	24
EXX	25
FILL addr_st, addr_end, byte[,bank]	18
FIND addr_st,addr_end, data	19
Help	26
IN addr	21
INTerrupt on off	25
Jump [[addr],bank]	22
LCATalogue [num_drive]	26
LDISs [addr_st[,addr_end]]	20
LDUmP [addr_st[,addr_end]]	19
LISTBReak LBReak	22
MEMory addr	19
MOVE addr_st1, addr_fin1, addr_st2[,bank]	18
OPTion bit on off	24
OUT addr,data	21
POP	18
PUSH data	18
RAM bank	18
ROM bank	18
SCREen [on off 5 7]	24
SEarch addr_st, addr_end, bank, data	19
SHow	24
SLOW type	23
WORKspace addr	24

Рис.3. Карта быстрой ссылки к командам отладчика

Оглавление

Введение	3
Часть 1. Сервис-монитор	5
1.1. Вход в сервис-монитор	5
1.2. Работа с принтером	6
1.3. Работа с диском	7
1.4. Установка режимов	7
1.5. Вход в отладчик	8
1.6. Меню сервис-монитора	8
Часть 2. Монитор-Отладчик	15
2.1. Строение экрана	15
2.2. «Горячие» клавиши	15
2.3. Редактор команд	16
2.4. Команды монитора	16
2.4.1. Изменение памяти и регистров	17
2.4.2. Управление памятью	18
2.4.3. Поиск в памяти	19
2.4.4. Реассемблирование и индикация памяти	19
2.4.5. Работа с внешними устройствами	21
2.4.6. Ловушки (точки останова)	21
2.4.7. Выполнение программы	22
2.4.8. Трассировка программы	23
2.4.9. Работа с экраном	24
2.4.10. Выход из монитора	24
2.4.11. Режимы монитора	24
2.4.12. Прочие команды	25
Часть 3. Расширение ОС Бейсик	27
3.1. Вывод на принтер (80h)	27
3.2. Вызов дисковой BIOS (81h)	27
3.3. Установка переменных монитора (82h)	32
3.3.1. Системные переменные монитора (DE=00)	33
3.3.2. Переменные режимов, цвета (DE=02)	33
3.3.3. Переменные дисковой BIOS (DE=04)	34
3.4. Очистка окна пользователя (83h)	35
3.5. Вывод символа в окно пользователя (84h)	36
3.6. Отключение монитора-отладчика (85h)	36
Приложения	37
А. Сообщения монитора	37
Б. Технические подробности	40
В. Заметки для программистов	42
Г. Версии	43
Д. Литература	44
Е. Карта памяти	44
Ж. Индекс	45

Страница для замечаний, размышлений и заметок об ошибках

200 - Почетная монета мира (с) 6 1-1 2