**Блочная и переносимая**

**базовая система ввода вывода**

**(B/P BIOS)**

Авторское право © 1990-93

Cameron W. Cotrill и Harold F. Bower

B/P BIOS, ее документация и ее утилиты являются объектами авторского права © 1990-92 Гарольд Ф. Бауэр {Harold F. Bower} и Кэмерон В. Коттрелл {Cameron W. Cotrill} - Все права защищены.

Авторы настоящим передают этот документ и продукт (BP BIOS и утилиты) под универсальной общественной лицензией GNU, действительной с 1 мая 2001.

Текущие адреса с этой даты:

|  |  |
| --- | --- |
| Harold F. Bower | Cameron W. Cotrill |
| 7914 Redglobe Ct. | 2160 N.W. 159th Place |
| Severn, MD 21144 | Beaverton, OR 97006 |
| USA. | USA. |
| [hfbower@alum.mit.edu](mailto:hfbower@alum.mit.edu) | [ccotrill@thinkstream.com](mailto:ccotrill@thinkstream.com) |

Оба автора хотели бы поблагодарить Аль-Хоули {Al Hawlay}, который сопровождал Ladera Z-Node и любезно предложил услуги Ladera в качестве центрального депозитария и центра поддержки во время разработки и первоначальный развертывания B/P BIOS.

B/P BIOS - исходная работа, спроектированная Кэмероном В. Коттреллом {Cameron W. Cotrill}, с поддержкой методологии, процедурами, и "черновой работой" Гарольда Ф. Бауэра {Harold F. Bower}.

Это руководство было отредактировано на нескольких 8-разрядных машинах, используя операционную систему, описанную здесь с помощью ZDE и WordStar 4.0. Оно было отформатировано, используя WordStar 7.0 на 386 клонах и распечатано на принтере Epson LQ 570.

Торговые марки: Little Board, Ampro Computers; Z80, Z180, Z280, Zilog; DDT, CP/M, Digital Research Inc.; ZCPR3, ZCPR33, ZRDOS, ZDH, Alpha Systems; WordStar, NewWord, MicroPro Int'l; Dbase II, Ashton-Tate; BackGrounder ii, DateStamper, Plu\*Perfect Systems; DosDisk, Z3PLUS, Bridger Mitchell; Turborom, Advent; NSC800, National Semiconductor; SB180, MicroMint; HD64180, Hitachi; XBIOS, Malcom Kemp; ZSDOS, ZDDOS, ZDS, Harold F. Bower - Cameron W. Cotrill – Carson Wilson.

**Благодарности**: Большое спасибо тестерам предварительных версий B/P BIOS за их верность, отзывы и предложения. Это следующие сердечные люди; Пол Чидли {Paul Chidley}, проектировщик YASBEC, Ян Коттрелл {Ian Cottrell}, Джеймс Тале {Jim Thale} и Ларри Мур {Larry Moore}, которые все участвовали в тестировании переносимой версии на YASBEC; и Терри Хейзен {Terry Hazen}, проектировщик платы расширения памяти MDISK для Ampro Little Board, который проверил B/P BIOS на этой платформе.

**Гарантия**: Несмотря на то, что этот продукт был произведен с максимальными усилиями авторов, мы не можем сертифицировать его как подходящий для любой конкретной задачи, для которой вы можете использовать его. Авторы будут прилагать разумные усилия для исправления недостатков, при уведомление о них, в пределах благоразумных финансовых ограничений. Так как этот продукт в настоящее время распространяется в виде исходного кода, специально для того, что бы пользователи могли изменять и адаптировать к своим компьютерным системам, авторы не могут нести ответственности за любые потери данных или неудобства из-за изменений пользователями дистрибутива системы. В духе сообщества Z-System авторы готовы, как свидетельствует этот проект, поделиться знаниями систем, и помочь другим узнать об операционных системах, работая с ними в деталях.

**Содержание**

1. Введение 7

1.1 Об этом руководстве 7

1.2 Условные обозначения 7

1.3 Что такое B/P BIOS? 8

1.4 История B/P BIOS 9

2 Функции B/P BIOS 10

2.1 Байт ввода вывода 10

2.2 Ввод/Вывод на запоминающие устройства большой емкости 11

2.3 Поддержка часов для времени и даты 11

2.4 Поддержка коммутации страниц памяти 11

2.5 Другие функции 12

3 Адаптация B/P BIOS 13

3.1 Принцип действия 13

3.2 Файлы B/P BIOS 15

3.3 Параметры B/P BIOS 17

3.4 Особенности конфигурирования 21

4 Установка B/P BIOS 22

4.1 Установка на дорожки начальной загрузки 23

4.2 Установка образа без коммутации банков памяти 25

4.3 BIOS с переключением банков, установка системы без переключения 26

4.4 Установка образа с полной поддержкой коммутации банков памяти 27

4.5 В случае возникновения проблем... 28

5. Программирование для B/P BIOS 31

5.1 Таблица переходов BIOS 31

5.2 Структуры данных BIOS 48

5.2.1 Область конфигурации 48

5.2.2 Заголовок параметров диска 54

5.2.3 Блок параметров диска 56

5.3 Системные структуры данных 59

5.3.1 Дескриптор окружения 59

5.3.2 Возможности терминала 62

6. Утилиты B/P BIOS 64

6.1 BPBUILD - утилита создания образа системы 64

6.1.1 Использование BPBUILD 64

6.1.2 Информация об экранах меню 65

6.2 BPCNFG - утилита конфигурации 68

6.2.1 Использование BPCNFG 68

6.2.2 Описание экранов меню 68

6.3 BPDBUG - Утилита отладки B/P BIOS 77

6.3.1 Использование BPDBUG 77

6.3.2 Команды BPDBUG 78

6.4 BPFORMAT - утилита форматирования дискет 82

6.4.1 Использование BPFORMAT 82

6.4.2 Конфигурация 83

6.4.3 Сообщения об ошибках BPFORMAT 84

6.5 BPSWAP - Утилита обмена логических обозначений дисков 86

6.5.1 Использование BPSWAP 86

6.5.2 Сообщения об ошибках BPSWAP 87

6.6 BPSYSGEN - утилита генерации системы 88

6.6.1 Использование BPSYSGEN 88

6.6.2 Сообщения об ошибках BPSYSGEN 88

6.7 COPY - универсальная утилита копирования файлов 90

6.7.1 Использование COPY 90

6.7.2 Параметры программы COPY 92

6.8 EMULATE Утилита эмуляции сторонних дисков 95

6.8.1 Использование EMULATE 95

6.8.2 Сообщения об ошибках EMULATE 96

6.9 HDBOOT - утилита загрузки с жесткого диска (заказная) 97

6.9.1 Использование HDBOOT 97

6.9.2 Сообщения об ошибках HDBOOT 97

6.10 HDIAG Форматирование жесткого диска/ Утилита диагностики 98

6.10.1 Использование HDIAG 99

6.10.2 Сообщения об ошибках HDIAG 101

6.11 INIRAMD Утилита инициализации виртуального диска 103

6.11.1 Использование INIRAMD 103

6.11.2 Режим командной строки 103

6.11.3 Сообщения об ошибках INIRAMD 104

6.12 INITDIR Утилита инициализации меток P2Dos-типа 105

6.12.1 Интерактивный режим INITDIR 105

6.12.2 Сообщения об ошибках INITDIR 105

6.13 INSTAL12 Утилита поддержки загрузочных дорожек 107

6.13.1 Использование INSTAL12 107

6.13.2 Сообщения об ошибках INSTAL12 110

6.14 IOPINIT Утилита инициализации пакета IO 113

6.14.1 Использование IOPINIT 113

6.14.2 Сообщения об ошибках IOPINIT 113

6.15 LDSYS Загрузчик файла образа системы 114

6.15.1 Использование LDSYS 114

6.15.2 Сообщения об ошибках LDSYS 115

6.16 MOVxSYS - Утилита загрузочных дорожек системы 116

6.16.1 Использование MOVxSYS 116

6.16.2 Сообщения об ошибках MOVxSYS 117

6.17 PARK - Утилита парковки головок жесткого диска 118

6.17.1 Использование PARK 118

6.17.2 Сообщения об ошибках PARK 118

6.18 SETCLOK утилита установки часов реального времени 119

6.18.1 Использование SETCLOK 119

6.18.2 Сообщения об ошибках SETCLOK 120

6.19 SHOWHD утилита просмотра раздела 122

6.19.1 Использование SHOWHD 122

6.19.2 Сообщение об ошибке SHOWHD 122

6.20 SPINUP Утилита управления двигателем жесткого диска 123

6.20.1 Использование SPINUP 123

6.20.2 Сообщения об ошибках SPINUP 124

6.21 TDD Утилита поддержки умных часов 125

6.20.1 Использование TDD 125

6.20.2 Настройка TDD 126

6.20.3 Сообщения об ошибках TDD 126

6.22 ZSCFG2 Утилита конфигурации ZSDOS2 128

6.22.1 Интерактивный режим ZSCFG2 128

6.22.2 Использование командной строки (режим эксперта) ZSCFG2 129

6.22.3 Описания параметров ZSCFG2 130

6.22.4 Сообщения об ошибках ZSCFG2 135

6.23 ZXD – Утилита вывода файлов для ZSDOS2 136

6.23.1 Использование ZXD 136

6.23.2 Параметры ZXD 136

6.23.3 Настройка ZXD 139

7 ZSDOS версии 2 140

8 ZCPR Версия 4 141

Глоссарий 142

Библиография B/P BIOS 145

Приложение 1 Справочник функций BIOS 147

Приложение 2 Справочник функций ZSDOS2 151

Приложение 3 Спецификации и форматы временных меток 154

Индекс 155

Примечания B/P BIOS 157

(Не используется)

# Введение

Блочная и переносимая (B/P) базовая система ввода вывода (BIOS) является попыткой стандартизировать механизмы многих логических в физические преобразований на микрокомпьютерах, выполняющих Z-System с ZSDOS. При расширении возможностей таких систем стало очевидно, что стандартная BIOS не содержит необходимой функциональности, адекватной стандартизации расширенных вызовов BIOS, ни внутренней структуры, для полной поддержки определений внешних системных параметров. B/P BIOS обеспечивает способ достижения этих целей, в то же время обладает гибкостью для работы с широким перечнем оборудования с гораздо меньшим уровнем системного программирования, чем требовалось ранее.

## Об этом руководстве

Документация B/P BIOS состоит из данного руководства плюс последних добавлений из файла README.2ND с дистрибутивного диска. Данное руководство состоит из следующих разделов:

* [***Функции B/P BIOS***](#_Функции_B/P_BIOS), краткое описание основных функции B/P BIOS в целом, отмечающее преимущества и некоторые ограничения в системе.
* [***Адаптация B/P BIOS***](#_Адаптация_B/P_BIOS_1)содержит подробную информацию об изменении многих параметров для создания индивидуального .REL файла с учетом вашей системы.
* [***Установка B/P BIOS***](#_Установка_B/P_BIOS_2) детализирует установку B/P BIOS в системах без и с использованием коммутации банков памяти в режиме пошаговой инструкции ("how to").
* [***Программирование для B/P BIOS***](#_5._Программирование_для) описывает интерфейсы, структуры данных и рекомендуемые приемы программирования, для обеспечения максимальной эффективности и производительности систем с B/P BIOS.
* [***Утилиты B/P BIOS***](#_6._Утилиты_B/P) описывают назначение, работу и настройку всех предоставленных утилит B/P BIOS и утилит поддержки.
* [***Приложения***](#_Приложение_1_Справочник), обобщают различную техническую информацию.
* [***Глоссарий***](#_Глоссарий_1), определяет много технических терминов, используемых в этом руководстве.
* [***Индекс***](#_Индекс) ключевых слов и фраз, используемых в данном руководстве.

Для тех, кто не заинтересован в технических подробностях, или, кто хочет перевести систему в рабочее состояние с предварительно сконфигурированной версией как можно быстрее, [Раздел 4](#_Установка_B/P_BIOS_1), Установка B/P BIOS, проведет вас через шаги установки, необходимые для выполнения окончательной адаптации к конкретному компьютеру. Остальные разделы касаются функций отдельных программных модулей, содержащихся в B/P BIOS и возможностях утилит, предоставленных для упрощения использования этого продукта.

## Условные обозначения

В данном руководстве используются различные термины и обозначения. В конце данного руководства в [Глоссарии](#_Глоссарий) приведены термины.

Хотя символы кажутся непонятными во-первых, они являются последовательным способом кратко изложить синтаксис программы. Как только вы научитесь читать их, вы можете сказать с первого взгляда как вводить даже самые сложные команды.

При описании синтаксиса программ используются некоторые специальные символы. По соглашению квадратные скобки ([ ]) указывают необязательные элементы командной строки. Вы можете не включать элементы, отображенные между квадратными скобками в вашей команде, но если вы их не укажете, программы обычно заменят их своими значениями по умолчанию. Если в команде используются элементы в скобках, все другие элементы в скобках также должны присутствовать, если эти элементы сами не являются скобками.

Все вспомогательные утилиты разработанные для поддержки B/P BIOS содержат встроенные экраны помощи, которые используют описанные выше соглашения, для отображения краткой информации о синтаксисе. Помочь всегда вызывается вводом после команды с двух символов наклонной черты (//). Так например,

ZXD //

вызывает справку для ZXD, расширенной программы отображения каталоговZSDOS. Кроме того, для интерактивных программ ZSDOS, таких как BPCNFG2, справка содержат более подробные описания сообщений, которые появляются по ходу сессии.

Многие утилиты могут быть вызваны из командной строки с параметрами, которые изменяют поведение программы. В соответствии с соглашением, параметры вводятся после других элементов команды. Например, параметр “P” в команде

ZXD \*.\* P

заставляет утилиту каталогов ZXD перечислить все файлы (\*.\*), и отправить ее вывод на принтер (P). Для удобства один символ косой черты (/), часто можно использовать вместо основных элементов, чтобы указать, что остальная часть командной строки содержит параметры. Таким образом, команда

ZXD /P

идентична по смыслу предыдущему примеру (см. [Раздел 6.23](#_6.23_ZXD_–) для получения более подробной информации о ZXD).

## Что такое B/P BIOS?

B/P BIOS - ряд программных процедур, которые непосредственно управляют микросхемами и другим оборудованием в вашем компьютере и обеспечивают стандартный программный интерфейс с операционной системой, такой как наша ZSDOS/ZDDOS, ZRDOS Echelon Incorporated или даже CP/M 2.2 Digital Research. Эти подпрограммы соответствуют стандартам CP/M 2.2 для базовой системы ввода вывода (BIOS) со многими расширениями. Некоторые основаны на CP/M 3.x (или CP/M Plus), а другие разработаны, чтобы обеспечить необходимые возможности современного программного обеспечения. Корректно написанные модули, содержащие B/P BIOS, позволяют выполнять все стандартные утилиты поддержки, почти все утилиты Z-System, и большинство прикладных программ без изменения.

Умение работать с коммутируемыми банками памяти и без них, система загрузки версий BIOS с единым набором программного обеспечения, для ряда различных компьютеров, плюс увеличение области транзитных программ для прикладных программ в системах с коммутацией банков памяти являются функциями, которых не предлагает ни одна другая система, о которой нам известно.

## История B/P BIOS

Наша предыдущая работа по развитию ZSDOS убедила нас, что мы должны были обратить внимание на машинно-зависимое программное обеспечение для Z80-совместимых компьютеров и разработать ряд стандартных улучшений для того, чтобы максимально эффективно использовать наши машины. Это предположение еще более актуально сегодня с большими жесткими дисками (свыше 100 мегабайт), частой необходимости в больших виртуальных дисках и постоянно сокращающейся области транзитных программ. Попытки добиться гибкости с обычными операционными системами были ограничены 64k адресуемой памятью в Z80-совместимых системах, и требовали принудительного изменения операционной системы, NZCOM и NZBLITZ, в которых быстро могут быть изменены различные конфигурации для удовлетворения требований приложений к памяти.

В середине-конце 1980-х годов, были предприняты некоторые усилия, чтобы использовать коммутацию страниц памяти для нескольких систем совместимых с CP/M 2.2. XBIOS был BIOS с переключением страниц памяти только для MicroMint SB-180 на основе HD64180. Хотя, он имел превосходный вывод на экран с гибким интерфейсом и возможностью работы со множеством периферийных устройств, он имел несколько причуд и заметно ухудшал производительность компьютера. BIOS с коммутацией страниц памяти был также написан для одноплатного S-100 компьютера XLM-180, но потребовал специальных версий многих утилит Z-System и не был произведен в значительном количестве. Другие попытки, такие как Epson portable, пытались создать BIOS с переключением страниц памяти, но не смогли достичь наших всеобъемлющих возможностей: совместимости с существующим программным обеспечением, высокой производительности и портативности.

В 1989 Cam разработал первый прототип B/P BIOS без поддержки переключения страниц памяти на своем TeleTek, в то время как Hal сконцентрировался на расширении ZSDOS и командного процессора. С 1997 B/P BIOS был установлена на:

|  |  |  |
| --- | --- | --- |
| YASBEC | - | Z180 CPU, FD1772 FDC, DP8490 SCSI, 1MB RAM |
| Ampro LB w/MDISK | - | Z80 CPU, FD1770 FDC, MDISK 1MB RAM |
| MicroMint SB-180 | - | HD64180 CPU, SMS9266 FDC, 256KB RAM |
| MicroMint SB180FX | - | HD64180Z CPU, SMS9266 FDC, 512KB RAM |
| Compu/Time S-100 | - | Z80 CPU, FD1795 FDC, 1 MB RAM |
| TeleTek | - | Z80 CPU, NEC765 FDC, 64KB RAM |
| D-X Designs P112 | - | Z182 CPU, SMC FDC37C665 FDC, Flash ROM, 512KB RAM (модификации для 5380 SCSI и GIDE) |

# Функции B/P BIOS

BIOS B/P разработан, чтобы быть полностью совместимым со стандартами CP/M 2.2 для базовой системы ввода вывода, а также обеспечить много расширений, необходимых для функционирования памяти с коммутацией страниц, которая становится очень распространенной с более новыми системами и процессорами. Кроме того, строгие стандарты кодирования, используемые в различных модулях, формирующих BIOS, упрощают проблемы интерфейса с прикладными программами и служат более устойчивой основой для будущих модификаций. Расширения, добавленные к основной функциональности CP/M 2.2, включают много элементов из CP/M 3 Digital Research (или CP/M Plus), но в более логически последовательной форме. Также в версию с коммутацией страниц памяти включены положения для управления до 8 МБ расширенной памяти для приложений использующих коммутацию страниц, электронных дисков и потенциальной многозадачности в будущих версиях. Чтобы обеспечить понимание используемой методологии, давайте теперь исследуем некоторые функции в универсальной B/P BIOS.

## Байт ввода вывода

Как определено Digital Research в их стандартах CP/M 2.2, байт ввода-вывода состоял из логических устройств, называемых TTY, CRT, UC1, CON, и т.д. B/P BIOS расширяет и обобщает эти интерфейсы, используя IOBYTE для определения четырех физических устройств, называемых COM1, COM2, PIO и NUL. Первые два, COM1 и COM2, являются последовательными портами, PIO - параллельный порт, в то время как NUL - "битоприемник", который может заменяться специализированным драйвером или использоваться вместо фактического устройства. Digital Research в CP/M 2.2 обеспечил только ограниченную интерфейсную возможность для устройств символьного ввода-вывода, состоящих из консоли (CON), вспомогательного (auxiliary) ввода и вывода (RDR/PUN) и принтера (LST). Возможность анализа состояния ввода и вывода с этими устройствами чрезвычайно ограничена и была улучшена в CP/M 3. Эти расширенные возможности полностью включены в B/P BIOS с добавлением строгих требований по использованию регистров, чтобы только соответствующие регистры могли быть изменены в соответствующих подпрограммах. Управляя IOBYTE, любые из четырех физических устройств могут использоваться с тремя логическими устройствами консолью (CON), вспомогательным устройством (AUX) и принтером (LST).

Кроме того, B/P BIOS включает функцию, являющуюся модификацией функции CP/M 3, для инициализации (или повторной инициализации) всех устройств и параметров, возвращающую адрес таблицы, которая содержит имена и параметры определенных символьных устройств. Пока не полностью совместимые с эквивалентными из CP/M 3, эти функции согласуются с идеями и функциональными возможностями, необходимыми этой расширенной системе. Включенными в таблицу устройств являются: определение флагов, о возможности устройства к вводу, выводу или обоим, скоростях передачи данных для последовательных устройств (максимум и настройки), последовательном формате данных, если это необходимо и методе квитирования (CTS/RTS, XON/XOFF или отсутствует), а также маски данных ввода и вывода для зачистки ненужных битов символов во время ввода вывода.

## Ввод/Вывод на запоминающие устройства большой емкости

Все версии CP/M BIOS Digital Research определяют только универсальный дисковый драйвер, а реализацию драйверов гибких дисков, жестких дисков, оперативной памяти и ленточных накопителей, оставлена пользователям или разработчикам. В B/P BIOS мы пошли на несколько шагов вперед, чтобы упростить большинство проблем. Во-первых, мы сохранили все стандартные функции и параметры CP/M 2.2, добавили CP/M 3 функции возврата адреса таблицы заголовка параметров диска (DPH) и сброса программного обеспечения, разблокирующего сегмент кода, и добавили новый вектор к таблице переходов BIOS, чтобы обеспечить стандартный метод прямой адресации функций низкоуровневых устройств. Несколько стандартных низкоуровневых функций гибкого диска поддерживаются и используются стандартными утилитами, включая функцию возврата типа используемого дискового контроллера, которая позволяет единой утилите поддержки адаптироваться к большому разнообразию аппаратных платформ. Предусмотрены аналогичные низкоуровневые функции жестких дисков SCSI/SASI и приложения для виртуальных (RAM) дисков в случае реализации специального оборудования. Методы, используемые для реализации этих механизмов доступа могут быть логически расширены для обработки ленточных накопителей или сетевых интерфейсов.

## Поддержка часов для времени и даты

Многие поставщики оборудования добавили поддержку даты и времени как нестандартные расширения в BIOS CP/M 2.2, и еще больше включили такую поддержку в BIOS CP/M 3. Мы решили определить вектор часов CP/M 3 как -стандартные часы, основывающиеся на нашей предыдущей работе по операционной системе ZSDOS. Эта точка входа в BIOS полностью соответствует нашим стандартам ZSDOS и может полностью заменить отдельный драйвер часов при использовании с ZSDOS. Для систем, способных обеспечить возвращение десятых долей секунд, таких как YASBEC и SB-180, стандарт был усовершенствован для поддержки этой возможности.

## Поддержка коммутации страниц памяти

Когда Digital Research добавил поддержку коммутации страниц памяти в CP/M 3, она была, несовместима со стандартами интерфейса BIOS, определенными ранее для CP/M. Метод, используемый в B/P BIOS, совместим с CP/M 2.2 в прямом доступе функций BIOS только за одним незначительным исключением при использовании переключения страниц памяти ZSDOS2 и содержит многие CP/M 3 расширения, добавленные для поддержки переключения страниц памяти с некоторыми изменениями, чтобы не противоречить стандартам, принятым для программного обеспечения Z-System. Исключение из доступа CP/M 2.2 имеет место, когда операционная система должна получить доступ к определенным буферам в *системном банке* памяти. В ZSDOS2, все буферы битов распределения (ALV), буферы регистрации (CSV), и буфер диска находятся в *системном банке* памяти, и непосредственно не доступны транзитным программам. Чтобы компенсировать это, мы добавили в ZSDOS2 команду возврата свободного места на дисках (наиболее распространенная причина для доступа к этим буферам) и адаптировали с учетом этого несколько утилит, чтобы приспособить их к системам с коммутацией банков памяти и без.

В дополнение к примитивам, инициируемым Digital Research, мы добавили функции, прямого доступа к *словам* и *байтам* в расширенных банках памяти, непосредственно получив доступ к подпрограммам, расположенным в альтернативных сегментах памяти, и корректное управление системой при возникновении ошибок. Эти функции делают B/P BIOS намного больше устойчивым и гибким, чем другие продукты. Эти опции реализованы способами, незаметными для системных утилит с тем, чтобы одни и те же функции были доступны в версиях с коммутацией банков памяти и без.

## Другие функции

В B/P BIOS имеется стандартизированный метод идентификации, который может быть использован для определения оборудования, на котором работает программа. Это позволяет приложениям, "адаптироваться" к окружающей среде способом, аналогичным тому, который используется в остальной части сообщества Z-System. Это также минимизирует системные "сбои", при выполнении программы, предполагающей определенные характеристики оборудования, которые могут быть губительными, если выполняются на других системах. Последствия идентификации параметров физической системы наиболее заметны в силу единого пакета утилит, реализующего низкоуровневые функции, такие как форматирование и диагностику, которые функционируют на весьма различных аппаратных платформах. Переносимость такого масштаба редко можно увидеть в других компьютерных системах.

Среда ZCPR 3.4 с расширениями обязательна в системе B/P BIOS. Начиная с добавления адреса системного сегмента и информации о размере CPR, DOS и BIOS, которые были добавлены в среду ZCPR 3.4, B/P BIOS также добавляет *резидентное пространство пользователя*, которое может использоваться, для определения местоположения уникальных подпрограмм для пользовательских приложений способом, подобным, но более последовательным, чем использует NZ-COM. Номер версии среды в 90H идентифицирует среду Z3, как являющуюся совместимой с B/P определениями.

В системах с коммутацией банков памяти прикладные программы могут также быть помещены в альтернативные сегменты памяти, используя информацию о расположении и размере, содержащуюся в стандартных позициях в структуре заголовка BIOS. Эта особенность обеспечивает существенно большую функциональность без ущерба для драгоценного места в области транзитных программ. В то время как схема, используемая в начальном дистрибутиве, подвергается незначительным корректировкам, чтобы ZSDOS2 с коммутацией банков становилась разработанной более надежно, эксперименты и предложения в этой области поощряются.

# Адаптация B/P BIOS

Чтобы настроить B/P BIOS для вашего использования или адаптировать ее к новому оборудованию, вам будут нужны редактор и ассемблер, способный к созданию стандартных перемещаемых объектных файлов в формате Microsoft. Системы использующие процессоры Hitachi HD64180 или Zilog Z180 должны быть ассемблированы с помощью ZMAC или SLR180, которые распознают расширенный набор мнемоник, или с помощью Z80 макроассемблеров с МАКРО-файлом, который разрешает блок расширенных инструкций. Для Z80 и совместимых процессоров, подходящими ассемблерами являются ZMAC и Z80ASM. Любой ассемблер не способный создавать стандартный объектный перемещаемый код Microsoft исключает возможность использования стандартных утилит для правильной установки системы B/P BIOS.

## Принцип действия

Для того чтобы понять необходимость и принципы лежащие в основе B/P BIOS, вы должны понимать, каким образом CP/M 2.2 с изменениями, внесенными Z-System, использует доступное адресное пространство памяти микропроцессора Z80. Для стандартных версий CP/M и совместимых систем, только абсолютные адреса памяти содержатся в *базовой странице*, которая расположена с 0 до 100H. Все адреса выше этого места являются переменными (в определенных пределах). Пользовательские программы обычно запускаются из *области транзитных программ* (TPA), которой доступно пространство оставшееся после выделения памяти всем компонентам операционной системы. Ниже изображена схема распределения памяти с выделенными областями, наряду с некоторыми общими элементами, назначенными каждой области памяти:

Следующее изображение показывает выделенные области памяти вместе с некоторыми общими элементами, расположенными в каждой области:

FFFFH ┌──────────────────┐

│ Буферы Z-System │ ENV, TCAP, IOP, FCP, RCP (~5k)

├──────────────────┤

│ BIOS │ Code + ALV, CSV, Буферы секторов (~5.5k)

├──────────────────┤

│ Операционная │

│ система │ CP/M 2.2, ZRDOS, ZSDOS1 (3.5k)

├──────────────────┤

│ Процессор команд │ CCP, ZCPR3.x (2k)

├──────────────────┤

│ Область │

│ │

│ транзитных │

│ │

│ программ (TPA) │

0100H ├──────────────────┤

│ Базовая страница │ IOBYTE, Jmp WB, Jmp Dos, FCB, Buffer

0000H └──────────────────┘

Поскольку все больше и больше функций добавляется в Z-System, буфера при использовании больших дисков требуют больше места для размещения ALV, и в код BIOS были добавлены дополнительные функции для современных систем, пространство отведенное TPA становится все более дефицитным.

B/P BIOS эту проблему решает способом, легко адаптируемым к различным аппаратным платформам. Она использует дополнительную память для более традиционных задач, чем создание простых виртуальных дисков, она перемещает большую часть накладных расходов, на альтернативные банки памяти. Универсальная схема выглядит следующим образом:

FFFFH ┌──────────┐

│ │

│ BNK1 │

│ │

8000H ├──────────┤ ┌──────────┐ ┌──────────┐ ┌──────────┐

│ │ │ ├┐ │ ├┐ │ ├┐

│ BNK0 │ │ BNK2 ││ │ BNKU ││ │ BNK3 │├┐

│ │ │ ││ │ ││ │ │||

0000H └──────────┘ └┬─────────┘│ └┬─────────┘│ └┬─────────┘││

└──────────┘ └──────────┘ └┬─────────┘│

│ BNKM │

└──────────┘

TPA SYSTEM USER RAM DISK

Как видно из приведенной выше схемы, несколько банков памяти могут быть назначены для различных функциональных областей памяти, каждый банк (за исключением одного, определенного как BNK1) размером 32K переключается в нижней части 32K карты памяти процессора. Банк определенный, как BNK1 всегда присутствует и называется *Общим банком*. Этот банк занимается частями операционной системы (процессором команд, операционной системой, BIOS и таблицами Z-System), которые могут быть доступны из других областей, и которые поэтому всегда должны быть "видны" в памяти процессора. Он также содержит код, для управления механизмами переключения банков B/P BIOS.

Чтобы проиллюстрировать это функциональное разделение, карта памяти основной системы B/P BIOS распределена следующим образом:

FFFFH ┌──────────────────┐

│ Буферы Z-System │

├──────────────────┤

│ Пространство │

│ пользователя │

├──────────────────┤

│ Bios │

├──────────────────┤

│ Операционная │

│ система │

├──────────────────┤

│ Процессор команд │

8000H ├──────────────────┤ ┌──────────────────┐ 8000H

│ │ │ Буфера Bios │

│ Область │ │ Перекл.часть Bios│

│ │ ├──────────────────┤

│ транзитных │ │ Перекл.часть DOS │

│ │ ├──────────────────┤

│ программ (TPA) │ │ Перекл.часть CCP │

│ │ ├──────────────────┤

│ │ │ Реставрация CCP │

0100H ├──────────────────┤ ├──────────────────┤ 0100H

│ Базовая страница │ │ Копия Базов. стр.│

0000H └──────────────────┘ └──────────────────┘ 0000H

TPA (BNK0/BNK1) Системный банк (BNK2)

Концепция B/P BIOS, выделяющая один байт для номера банка, позволяет непосредственно контролировать до 8 Мегабайт. В схеме нумерации сделаны некоторые предположения, главным из которых является, что банк BNK0 располагается в нижней части физического ОЗУ, BNK1 следующий банк дополняющий ОЗУ, остальные следуют с последовательно увеличивающейся нумерацией. Несколько примеров могут проиллюстрировать этот процесс. YASBEC предлагает несколько вариантов карты памяти. Модули с MEM 1, 2 или 3 PAL декодер присваивают первые 128k физической памяти ПЗУ начальной загрузки, таким образом, BNK0 установлен в 4 (Банки 0-3 являются ПЗУ). MEM 4 PAL использует только первые 32K (Физический Банк 0) для ПЗУ, что означает, что BNK0 присвоен 1, BNK1 к 2 и т.д. до максимума в 1 мегабайт, где BNKM равняется 31.

С другой стороны, Ampro Little Board, укомплектованный MDISK, полностью удаляет ПЗУ начальной загрузки из карты памяти, оставляя максимум 1 Мбайт непрерывного пространства ОЗУ. В этой системе BNK0 установлен в 0 и BNKM к 31 для платы MDISK, полностью оборудованной до 1 Мбайт.

Область, начинающаяся после BNK1, упоминается как *Системный банк*. Он начинается с номера банка, присвоенного BNK2 и заканчивается номером банка непосредственно перед присвоенный номером банка пользователя BNKU, если он присутствует, или BNK3 если область банка пользователя не определена.

Если она присутствует, один или более 32K банков памяти могут быть определены как BNKU для уникальных пользовательских программ или областей хранения. Эта область начинается с незадействованного номер банка и заканчивается номером незадействованного банка непосредственно перед BNK3. BNK3 определяет верхнюю область физической памяти, которая наиболее часто используется для виртуального диска, обеспечивающего быстрое временное рабочее пространство в виде эмулируемого диска.

B/P BIOS содержит механизмы защиты в виде программных проверок, чтобы убедиться, что обеспечиваются критические участки карты памяти. В случае систем без коммутации банков памяти производится проверка, чтобы убедиться, что размер системы не настолько большой, что BIOS может перезаписать зарезервированные в памяти области Z-System (RCP, IOP, и т.д.). Если обнаружено возможное переполнение, сообщение

**++ mem ovfl ++**

будет выдаваться, при запуске системы. В системах с коммутацией банков на экран будет выведено это сообщение, если вершина частей системы в системном банке превысит 32K - размер банка. Для большинства систем это пространство все еще разрешает дискам в нескольких сотен мегабайт быть размещенными.

Поскольку общие части компонентов операционной системы должны оставаться видимым для приложений, аналогичная проверка делается для уверенности, что нижний адрес, используемый процессором команд является равным или больше чем 8000H. Этот фактор проверяется программами MOVxSYS и BPBUILD с выдачей предупреждения, в первом случае, или проверкой достоверности записи в последнем случае.

## Файлы B/P BIOS

Эта BIOS разделена на несколько файлов, часть из которых в значительной степени зависят от конкретного оборудования, используемого на компьютере, и некоторые являются общими, и не должны изменяться, чтобы собрать рабочую систему. Широко используется условное ассемблирование, чтобы адаптировать результирующий файл BIOS к требуемой конфигурации. Основной файл, BPBIO-xx.Z80, определяет, какие файлы используются для сборки образа BIOS под управлением включаемого файла, DEF-xx.LIB. Именно этот файл, выбирает функции и содержит аппаратно-зависимые мнемонические эквивалентности. Поддерживая максимальный возможный код в общих модулях, которые не требуют никаких изменений, версии B/P BIOS относительно просто приспособить для различных машин. Независимые модули, используемые в системе B/P BIOS:

|  |  |  |
| --- | --- | --- |
| BOOTRAM.Z80  BOOTROM.Z80 | - | Требуются только в приложениях ПЗУ начальной загрузки (BOOT ROM) |
| BYTEIO.Z80 | - | Символ IO через IOBYTE используя IIO-хх процедуры |
| DEBLOCK.Z80 |  | Процедуры разблокирования диска |
| DPB.LIB | - | Определения форматов дискет 3.5/5.25" (если автоматический выбор) |
| DPB8.LIB | - | Определения форматов дискет 8"/HD (если автоматический выбор) |
| DPB2.LIB | - | Дополнительные определения дискет (опционально, если автоматический выбор) |
| DPBRAM.LIB | - | Фиксированные определения форматов дискет (если не автоматический выбор) |
| DPH.LIB | - | Таблица заголовков параметров дисков и определения дискет |
| FLOPPY.Z80 | - | Высокоуровневое управление гибким диском |
| SECTRAN.Z80 | - | Процедуры трансляции секторов |
| SELFLP1.Z80 | - | Процедура выбора дискет (если не автоматический выбор) |
| SELFLP2.Z80 | - | Процедура выбора дискет (если не автоматический выбор) |
| SELRWD.Z80 | - | Общие процедуры чтения/записи |
| Z3BASE.LIB | - | ZCPR 3.x файл с мнемоническими эквивалентностями для настройки среды |

Другие файлы в различной степени зависят от оборудования. Этим модулям, требующим настройки под различные наборы оборудования систем, присвоены имена, которые заканчиваются универсальным обозначением "-xx" для идентификации конкретной версии. Адаптация этих модулей варьируется от простой настройки командной строки до полного переписывания. Версии B/P BIOS, созданные на сегодняшний день определены как:

|  |  |  |  |
| --- | --- | --- | --- |
| "-18" | - | MicroMint SB-180 | (64180 CPU, 9266 FDC, 5380 SCSI) |
| "-FX" | - | MicroMint SB180FX | (Z180 CPU, 9266 FDC, 53C80 CSI) |
| "-YS" | - | YASBEC | (Z180 CPU, 1772 FDC, DP8490 SCSI) |
| "-DX" | - | D-X Designs P112 | (Z182 CPU, 37C665 FDC, Flash ROM) |
| "-AM" | - | Ampro Little Board | (Z80 CPU, 1770 FDC, 1MB MDISK) |
| "-CT" | - | Compu/Time S-100 board set | (Z80 CPU, 1795 FDC, 1MB Memory) |
| "-TT" | - | Teletek | (Z80 CPU, 765 FDC) |
| "-XL" | - | XL M-180 SBC | (HD64180 CPU, 9266 FDC, PIO SCSI) |

Файлы, связанные с определенными версиями оборудования или, требующие, адаптации:

|  |  |  |
| --- | --- | --- |
| BPBIO-xx.Z80 | - | Основной файл, для адаптации имен включаемых файлов |
| CBOOT-xx.Z80 | - | Подпрограммы “Холодного” старта, приглашение входа в систему |
| DEF-xx.LIB | - | Мнемонические эквивалентности для настройки параметров, режима, скорости, и т.д. |
| DPBHD-xx.LIB | - | Определения разделов жесткого диска (опционально) |
| DPBM-xx.LIB | - | Определения RAM диска (опционально) |
| DPHHD-xx.LIB | - | Определения DPH жесткого диска (опционально) |
| DPHM-xx.LIB | - | Определения DPH RAM диска (опционально) |
| FDC-xx.Z80 | - | Процедуры низкого уровня интерфейс/драйвер дискеты |
| HARD-xx.Z80 | - | Процедуры низкого уровня интерфейс/драйвер жесткого диска |
| IBMV-xx.Z80 | - | Процедуры поддержки переключения банков памяти (если с переключением) |
| ICFG-xx.Z80 | - | Конфигурационный файл для скорости, физических дисков, и т. д. |
| IIO-xx.Z80 | - | Процедуры и определения символа ввода вывода |
| RAMD-xx.Z80 | - | Процедуры интерфейс/драйвер RAM диска (опционально) |
| TIM-xx.Z80 | - | Процедуры счетчик/таймер и драйвер часов ZSDOS |
| WBOOT-xx.Z80 | - | Процедуры “Теплого” старта и повторной инициализации |

## Параметры B/P BIOS

Наиболее логичной отправной точкой в начале конфигурации является редактирование файла DEF-xx.LIB, для выбора нужных параметров. Этот файл - основное руководство по выбору параметров для вашей системы, и некоторый тщательный выбор здесь минимизирует размер BIOS и увеличивает функциональность. Некоторые более важные параметры и краткое их описание:

**MOVCPM** - Интегрироваться в загрузчик MOVCPM “типа”? **Если** система будет интегрирована в систему MOVCPM, дескриптор среды, содержащийся в подпрограмме CBOOT всегда перемещается в позицию как часть процесса “Холодного” старта. Если задано значение NO, будет осуществлена проверка, чтобы видеть, загружен ли уже дескриптор среды, и копия BIOS не будет загружена, если такой имеется.

*Примечание*: При ассемблировании BIOS для инсталляции загрузочной дорожки (MOVCPM установлен в YES), много параметров удаляется для экономии места и номер версии BIOS становится 1.1.

**BANKED** - Эта BIOS с переключением страниц? Если установлено значение YES, модуль управления банками, IBMV, включается в ассемблирование, и большая часть кода перемещается в *системный банк*. Обратите внимание, что системы с переключением страниц не могут быть размещены на системных дорожках, или интегрированы в образ MOVCPM.

**IBMOVS** - Возможны ли прямые межбанковских перемещения? Если задано значение YES, прямая передача данных между банками возможна для процессоров Zilog Z180/Hitachi 64180. Если NO, буфер передачи размером 256 байт помещается в верхнюю общую память и межбанковские перемещения будут осуществляться передачей байтов через этот буфер.

**ZSDOS2** - Собрать для системы ZSDOS2 с переключением банков? Если YES, буферы ALV и CSV будут помещены в *системный банк*, невидимый для обычных программ. У этого есть побочный эффект. Многие программы CP/M, осуществляющие обработку файлов (выдачу списка каталога, DATSWEEP, MEX, и т.д.), которые не знают об этой функции, будут сообщать неверные размеры. Преимущество состоит в том, что не расходуется TPA для больших жестких дисков. Установите параметр в NO, если вы хотите иметь строгую совместимость CP/M 2.2.

**FASTWB** - Восстанавливать процессор команд из *системного банка* ОЗУ? Если имеет значение YES, “Теплый” старт будет восстанавливать процессор команд из зарезервированной области в *системном банке* ОЗУ, а не из загрузочных дорожек. Для максимальной пользы от B/P BIOS, всегда пытайтесь установить его в YES. В системах без расширенной памяти, он должен быть установлен в NO.

**MHZ** - Установить скорость процессора к ближайшему целому в мегагерцах (например для частоты 9.216 МГц, равную 9). Введенное здесь значение используется во многих системах для вычисления значений времени и/или параметров скорости последовательной передачи данных.

**CALCSK** - Вычислить таблицу чередования секторов дискеты? Если NO, таблица чередования секторов используется для каждого формата дискет включенного в изображение. Расчет чередования секторов, как правило, более эффективен с точки зрения размера, хотя и немного медленнее, однако эти факторы настолько малы, что практически неизмеримы.

**HAVIOP** - Включить код IOP в таблицу переходов? Если процедура IOPINIT удовлетворяет ваши требования инициализации IOP, вы можете выключить этот параметр, установив NO и сэкономить немного места. Он обычно будет отключен при создании системы для интеграции MOVCPM для экономии места.

**INROM** - Это альтернативный банк в ПЗУ? Значение NO для систем на базе обычных дисков. Пожалуйста, свяжитесь с авторами, если вам нужна дополнительная информация о системных компонентах на основе ПЗУ.

**BIOERM** - Печатать сообщения об ошибках BIOS? Установите его в YES, если вы желаете прямую печать сообщений BIOS об ошибках гибкого диска. Однако, если вы создаете BIOS для размещения на дорожках начальной загрузки, то у вас не будет, вероятно, хватать места и необходимо выключить его. Установите значение в NO, чтобы просто возвращать нормальный флаг ошибки успеха/сбоя без распечатки сообщения.

**FLOPY8** - Включать форматы 8"/HD дискет? Некоторые системы (SB-180, Compu/Time) могут обрабатывать 5,25" и 8" диски. Если ваше оборудование поддерживает возможность и вы хотите использовать 8" диски, а также обычные 3,5" и 5,25" дискеты, установив его YES будут добавлены форматы, содержащиеся в DPB8.LIB и задействована логика условного ассемблирования. Будущие системы могут использовать гибкие диски "High-Density" 3.5" и 5.25", которые используют более высокие скорости передачи данных. Этот флаг также будет контролировать их определения.

*Примечание*: Если AUTOSL будет установлен в NO, этот вариант, вероятно, заставит BIOS быть больше, чем необходимо, так как эти дополнительные форматы могут быть не доступны.

**MORDPB** - Использовать больше DPB дискет (в дополнение к нормальным 4-5.25" и дополнительным 8")? Если YES, будет включен файл DPB2.LIB. Многие форматы являются шаблонами и могут быть заполнены любыми неконфликтными форматами, которые вы пожелаете.

*Примечание*: Если AUTOSL будет установлен в NO, этот вариант, вероятно, заставит BIOS быть больше, чем необходимо, так как эти дополнительные форматы могут быть не доступны.

**MORDEV** - Включать драйверы устройств дополнительных символов? Установив это значения в YES, определяемые пользователем драйверы добавятся к таблице символов IO, и ассемблируется код соответствующего драйвера. Системы, с платами расширения, такие как SB-180 и YASBEC могут теперь воспользоваться преимуществами дополнительных последовательных и параллельных интерфейсов в базовом BIOS. Значение NO ограничит код 4 основными драйверами.

*Примечание*: При ассемблировании BIOS для инсталляции на загрузочные дорожки (MOVCPM установлен в YES), много параметров удаляется для экономии места и номер версии BIOS становится 1.1. в файлах дистрибутива.

**BUFCON** - Использовать буфер опережающего ввода для консоли? Если задано значение YES, добавляется код для создания и управления буфером опережающего ввода для драйвера консоли. Это устройство будет контролироваться либо прерываниями (в таких системах как YASBEC и SB-180), либо фоновым опросом (в Ampro и Compu/Time). Это означает, что символы, вводимые в то время, когда компьютер делает что-либо другое не будут потеряны, но будут сохраняться, пока не потребуются.

**BUFAUX** - Использовать буфер опережающего ввода для вспомогательного (Auxiliary) порта? Как с BUFCON выше, выбор YES будет добавлять код для создания и управления буфером опережающего ввода для вспомогательного устройства. Так как порт AUX обычно используется для модемных соединений, буферирование ввода минимизирует потерю символов от удаленного конца.

**AUTOSL** - Выбрать гибкие форматы автоматически? Если установлено в YES, выбор гибких дисков будет использовать алгоритм из SELFLP2.Z80, чтобы идентифицировать формат диска из включенных файлов DPB (DPB.LIB, опционально DPB8.LIB и опционально DPB2.LIB) и регистрировать диск, если соответствие найдено. Не должно быть **никаких** конфликтных определений, включенных в различные файлы для того, чтобы функционировать должным образом. Посмотрите примечания в различных файлах, для уточнения ограничений. Если установлено в NO, будет включен единственный файл DPBRAM.LIB, который может быть адаптирован, чтобы содержать только фиксированный формат или форматы, желаемых дисководов. Это уменьшает размер кода, но менее гибко.

**RAMDSK** - Включать код для RAM-диска? Если установлено в YES, любая память выше системного или пользовательского банка может быть использована для виртуального диска (по умолчанию это диск M:) путем включения файла RAMD-xx.Z80. Параметры для определения размера и конфигурации также включены в файлы DPHM-xx.LIB и DPBM-xx.LIB. В системах без расширенной памяти, или для экономии места, при построении системы для загрузочных дорожек, он может быть отключен путем установки NO.

**HARDDSK** - Включить драйвера жесткого диска SCSI? Установите значение YES, если вы хотите включить возможность доступа к жестким дискам. В система только с дискетами, установка NO сведет к минимуму код BIOS.

**HDINTS** - (Системно-зависимый) В некоторых системах, таких как YASBEC, могут использоваться управляемые прерыванием контроллеры жестких дисков, использующие возможности передачи DMA. Если вы хотите использовать этот тип драйвера, определенный в файле HARDI-xx.Z80 вместо обычной процедуры опроса, включенной в HARD-xx.Z80, установите этот параметр к TRUE. В большинстве случаев этот драйвер потребует больше места в области транзитных программ, так как подпрограмма обработки прерываний должна быть в общей памяти.

**CLOCK** - Включать кода драйвера часов ZSDOS? Если задано значение YES, вектор в BIOS+4EH будет содержать ZSDOS-совместимый драйвер часов с физическим кодом, содержащимся в модуле TIM-xx.Z80. Если NO, обращение к BIOS+4EH возвращает код ошибки.

**TICTOC** - (Системно-зависимый) Использовать счетчик псевдо пульса? Эта функция используется в системах, таких как Ampro Little Board и Compu/Time SBC880, которые не имеют схемы прерываний для управления часами реального времени. Вместо этого в код включается серия ловушек, (опрос статуса символа IO, опрос статуса дискеты) для проверки переполнения 1-секундного счетчика. Это менее предпочтительно, чем системы основанные на прерываниях, но достаточно если отсутствует другой доступный способ. Значение NO, если не требуется.

**QSIZE** - размер в байтах буферов опережающего ввода, контролируемых BUFCON и BUFAUX.

**REFRSH** - Активировать функцию Dynamic Refresh (динамическое обновление) процессоров Z180/HD64180? В некоторых компьютерах, с помощью этих процессоров, таких как YASBEC, обновление не требуется и только замедляет обработку. Установите значение NO, если вам не нужна эта функция. Если ваш процессор использует динамическую память, или нуждается в сигнале для других целей (например SB180 использует Refresh для DMA гибких дисков), установите значение YES.

**Z3** - Включить код инициализации ZCPR? Поскольку среда Z3 является обязательной в B/P BIOS (который теперь "принадлежит" окружающей среде), эта опция имеет мало значения.

При ассемблировании версии B/P BIOS с переключением банков, должна быть сделана идентификация различных банков памяти с тем, чтобы различные системные компоненты "знали", где они расположены. Описание этих областей присутствует в [Разделе 3.1](#_3.1_Принцип_действия) выше. Значение BNK0 должно быть первым банком ОЗУ в системе, если другое декодирование не сделано. Должны быть установлены следующее константы с помощью директивы приравнивания:

|  |  |  |
| --- | --- | --- |
| BNK0 | - | Первый 32K банк ТРА (переключается в/из) |
| BNK1 | - | Второй 32K банк ТРА (Общий банк) |
| BNK2 | - | Начало области системного банка (BIOS, DOS, CPR) |
| BNKU | - | Начало последовательности банков для пользовательских приложений |
| BNK3 | - | Начало дополнительных банков (первый банк для использования электронного диска) |
| BNKM | - | Назначенный максимальный номер банка |

## Особенности конфигурирования

При компиляции версии B/P BIOS для компоновки в файл IMG размер получаемого образа не является большой проблемой, таким образом, вы не должны волноваться о незначительном вопросе размера. Однако, для интеграции в системы записываемые на дорожки начальной загрузки дискет, ограничение очень реально, так как необходимо обеспечить, чтобы CPR/DOS/BIOS и загрузочный сектор(ы) могли поместиться на зарезервированные системные дорожки. Как правило, существует предел около 4.5K для компонентов BIOS. Когда флаг MOVCPM будет установлен в YES для этого типа, ассемблером будут выдаваться предупреждения, если образ превышает 4352 байта (максимум для систем с 2 загрузочными записями) и 4480 байтов (максимум для систем с одной загрузочной записью). Достижение этих пределов часто требует отключения многих функций.

Первое, что вы должны сделать перед компиляцией BIOS - это создать резервную копию всего диска, затем скопировать только необходимые файлы на рабочий диск для редактирования. Установив нужные параметры, редактируйте определения оборудования в ICFG-xx.Z80 для отражения физических характеристик ваших накопителей гибких и жестких дисков, а также любых других подходящих элементов. Затем отредактируйте логические характеристики для жесткого и виртуального накопителей (если таковые имеются) в DPBHD-xx.LIB и DPBM-xx.LIB. Если вы не предполагаете использовать любой из стандартных форматов дискет или хотите изменить их, редактируйте DPB.LIB и/или DPB2.LIB (если используете автоматический выбор) или DPBRAM.LIB, если вы используете фиксированные форматы гибких дисков. Наконец отредактируйте файлы DPH, чтобы назначить логическим дискам, желаемые номера в диапазоне “A”..”P”.

Решите, хотите ли вы генерировать систему, используя файл образа, разработанный для поддержки B/P BIOS (BPBUILD/LDSYS), или для интеграции на загрузочные дорожки дискет. Если последнее, вы, вероятно, не сможете иметь все параметры включенными. Например, для MicroMint SB-180, следующие параметры должны быть отключены: BANKED, ZSDOS2, BIOERM, FLOPY8, MORDPB, BUFAUX и обычно либо CLOCK или RAMDSK. В качестве помощи для уменьшения размера, условная компиляция, основываясь на флаг MOVCPM автоматически подавляет все форматы, кроме двухсторонних гибких дисков из DPB.LIB. При настройке для загрузки с дорожек дискеты (флаг MOVCPM установлен в значение TRUE), во время компиляции будут выводится предупреждения, если размер превышает доступный для одно или двух-секторной загрузочной записи. Используя метод BPBUILD/LDSYS, вы можете изменить почти все системные параметры, даже делая различные системы для последующей динамической загрузки.

Если вы используете версию B/P BIOS, уже установленную для вашего типа компьютера, вы теперь готовы скомпилировать, построить систему и выполнить ее. Единственной оставшейся задачей является необязательная адаптация текста баннера в файле CBOOT-xx.Z80 и повторная перекомпиляция .REL файла.

Для тех, кто преобразует стандартную версию B/P BIOS к системе с новым оборудованием, мы рекомендуем, чтобы вы начали с системы только для дискет в режиме без переключения банков, затем расширяли ее. Самый простой способ проверить новые версии для использования, является режим образа системы (IMG файл), а затем перейти к установке на загрузочные дорожки, если это необходимо. Усовершенствования, которые могут быть добавлены после тестирования предыдущих версий могут включать: добавление жестких, виртуальных дисков и наконец переключения банков памяти.

# Установка B/P BIOS

Дискета(ы) дистрибутива, на которой предоставлена B/P BIOS, сконфигурирована для начальной загрузки на унифицированном оборудовании для заказанной версии. Стандартный последовательный терминал 9600 бит/с позволит вам сразу поднять минимальную систему с использованием гибких дисков без переключения банков в рабочее состояние. В связи с разнообразием различных системных конфигураций и ограничения размера в некоторых версиях, только накопители гибких дисков могут быть использованы в качестве первоначального загрузочного диска. В случаях, когда остается место на загрузочных дорожках, также предоставляется ограниченная поддержка жестких дисков, и в некоторых конфигурациях, имеется даже поддержка виртуального диска.

После начальной загрузки с установленной системы или с загрузочных дорожек дистрибутивного диска, отформатируйте одну или несколько новых дискет и скопируйте содержимое дистрибутивной дискеты на резервную дискету(ы). Скопируйте загрузочные дорожки с основной дискеты на резервную, используя BPSYSGEN (см. [6.6](#_6.6_BPSYSGEN_-)). Удалите основную дискету для сохранности и работайте только с копией, которую вы только что сделали.

Демонстрационный файл автозагрузки STARTUP.COM на дистрибутивном диске автоматически выполняет последовательность инструкций, во время загрузки системы. Он содержит различные инструкции, которые далее адаптируют систему и загружают части операционной системы, которые слишком большие, чтобы поместиться на загрузочных дорожках. Последовательность инструкций по умолчанию следующая:

|  |  |  |
| --- | --- | --- |
| LDDS | <-- | Загрузка подпрограммы временных меток файлов DateStamper типа |
| LDR SYS.RCP,SYS.FCP,SYS.NDR | <-- | Загрузка сегментов среды ZCPR 3 для резидентного процессора команд, пакета управления потоками и именами каталогов |
| IOPINIT | <-- | Инициализация пакета процессора ввода-вывода |
| TD S | <-- | Подсказка для даты и времени, в качестве альтернативы установки времени можно использовать TDD ([6.21](#_6.21_TDD_Утилита)) или SETCLOK ([6.18](#_6.18_SETCLOK_)) |
| IF ~EX MYTERM.Z3T | <-- | Если файл MYTERM.Z3T не существует... |
| TCSELECT MYTERM.Z3T | <-- | …выберите, какой терминал у вас есть, создание файла MYTERM.Z3T |
| FI | <-- | ... конец IF |
| LDR MYTERM.Z3T | <-- | Загрузка заданных определений терминала |

Если вы хотите изменить эти первоначальные инструкции, например, чтобы инициализировать виртуальный диск с помощью INIRAMD, добавить функционирование временных меток файлов с помощью INITDIR или PUTDS и скопировать некоторые файлы с помощью COPY, они могут быть добавлены с помощью ALIAS, VALIAS, SALIAS или других совместимых файлов доступных в областях ZSYSTEM или ZCPR33 на Z-Nodes.

После первоначальной инициализации системы и выполнения системы по умолчанию с загрузочных дорожек, вы можете расширить процесс генерации систем различного назначения для получения наибольшей отдачи от вашей системы. Возможны многие виды установок, самым простым из которых является система без переключения банков памяти, размещаемая в основной памяти, использующая только 64K системной памяти. Такая система использует обычный процессор команд, например, ZCPR3.x и операционную систему без переключения банков, как наша ZSDOS версии 1. Системы без переключения банков могут быть установлены на загрузочные дорожки диска, или созданы в виде файла с образом для динамической загрузки, с помощью утилиты LDSYS (см. [6.15](#_6.15_LDSYS_Загрузчик)).

Системы с переключением банков памяти должны быть созданы с помощью утилиты BPBUILD (см. [6.1](#_6.1_BPBUILD_-)) и загружены с помощью LDSYS (см. [6.15](#_6.15_LDSYS_Загрузчик_1)). Методы управления различными банками памяти для формирования полнофункциональной операционной среды довольно сложные и лучше обрабатываются нашими утилитами. Многие файлы образов могут быть созданы и загружены в случае необходимости адаптации системы для достижения оптимальной производительности. Следующие разделы подробно описывают эти различные виды установок.

## Установка на дорожки начальной загрузки

Для большинства существующих CP/M совместимых компьютеров, чтобы начать выполнение дисковой операционной системы, программа должна быть помещена в определенную область гибкого или жесткого диска. Обычно, первые две или три дорожки на диске зарезервированы с этой целью и называются “Boot Tracks” (дорожки начальной загрузки). Поскольку место обеспеченное таким образом, как правило ограничено, установка полной B/P BIOS, или системы с переключением банков не возможна. Вместо этого для запуска компьютера используется уменьшенная система, эквивалентная работающей в настоящий момент, и служит в качестве операционной системы, для последующей загрузки более крупных систем, при необходимости.

Если вы используете предварительно настроенную версию B/P BIOS для вашего оборудования, вы можете просто продолжить использовать загрузочные сектора системы с дистрибутивного диска(ов) путем копирования системы, согласно инструкции приведенной в [Разделе 4](#_Установка_B/P_BIOS) выше, используя BPSYSGEN (см. 6.6). Если вы решили изменить или иначе настроить загрузочный сектор системы, вы должны скомпилировать исходный код B/P BIOS, настроив некоторые параметры в файле DEF-xx.LIB для обеспечения правильного типа системы. Для компиляции системы размещаемой на загрузочных дорожках наиболее важными параметрами являются:

MOVCPM Установите YES

BANKED Установите NO

ZSDOS2 Установите NO

Один вид системы с переключением банков пригоден для установки на дорожки начальной загрузки, при наличии дополнительной памяти, и подпрограммы вашего B/P BIOS поддерживают такую функцию. Эта функция загружает процессор команд из банков памяти, а не из загрузочной дорожки диска, как правило, содержит меньше кода (занимает меньше места на загрузочной дорожке) и выполняется быстрее. Он может быть настроен с помощью следующих настроек:

FASTWB Установки YES, если необходимо, NO, если теплый старт с диска

Некоторые из функций, которые обычно должны быть отключены для уменьшения размера системы устанавливаются следующим образом:

MORDPB Установите NO

DPB8 Установите NO

MORDEV Установите NO

Когда по крайней мере определения этих параметров и любых других, которые вы хотите изменить (см. [Раздел 4](#_4_Установка_B/P)) внесены в файлы компонентов системы, скомпилируйте свой файл BPBIO-xx.Z80 в стандартный файл Microsoft REL. Этот выходной файл может использоваться, чтобы наложить часть BIOS с помощью утилиты генерации системы MOVxSYS.COM (см. [6.16](#_6.16_MOVxSYS_-)), присутствующей на вашем дистрибутивном диске или аналогичной программы, имеющейся на вашем компьютере. Утилита MOVxSYS или ее аналог (MOVCPM, MOVZSYS, и т.д.) является специальной программой, настроенной для вашего конкретного оборудовании, содержащей все компоненты операционной системы, которые будут размещены на загрузочных дорожках, наряду с выполнением процедуры изменения внутренних адресов, чтобы соответствовать указанному размеру памяти.

Чтобы добавить новый BIOS, который вы только что скомпилировали, запустите INSTAL12 (см. инструкцию в [6.13](#_6.13_INSTAL12_Утилита)), указав программу MOVxSYS или аналогичную для вашего компьютера и следуйте подсказкам для наложения нового BIOS. После сохранения программой INSTAL12 перемещаемого или абсолютного файла вы готовы создать загрузочный диск, содержащий модифицированную систему.

Если вы используете команду INSTAL12 для установки сегментов системы на MOVxSYS или эквивалентную программу, необходимо сначала создать файл системы в абсолютном формате. Поскольку функциональная часть вашей новой программы является идентичной оригинальной MOVxSYS или аналогичной, используйте метод объяснённый в вашей первоначальной документации для создания новой системы. Команда MOVxSYS имеет вид:

MOVxSYS nn \* <-- Заменить MOVxSYS своей версией

где “nn” - размер системы (обычно 51 для умеренной системы начальной загрузки). Звездочка говорит программе сохранять изображение в памяти и не писать его в файл на диске. Вы можете теперь использовать BPSYSGEN, чтобы записать новый образ на системные дорожки вашей загрузочной дискеты. Сделайте это, выполнив BPSYSGEN без параметров и введите только возврат каретки в ответ, на вопрос относительно источника образа.

Если вы используете команду INSTAL12 /A, чтобы установить сегменты, заменяющие системные, поверх файла образа системы, или используете утилиту, которая записала новое изображение в файл на диске, используйте BPSYSGEN, чтобы записать файл с образом на системные дорожки вашего загрузочного диска. Правильная команда

BPSYSGEN filename

где filename - это имя файла на диске, который вы только что создали, выполнив MOVxSYS или аналогичную, с выводом в файл на диске, или с помощью INSTAL12 на существующий файл с образом.

Если система записана на жесткий диск, и ваш компьютер поддерживает загрузку с жесткого диска, например, YASBEC, вы обычно должны изменить загрузочный сектор по умолчанию от загрузочного сектора гибкого диска по умолчанию, содержащийся в MOVxSYS или аналогичной. Эта перестройка осуществляется программой HDBOOT (см. [6.9](#_6.9_HDBOOT_-)), которая должна быть настроена для конкретного оборудования используемого в системе.

После того, как вышеупомянутые действия были корректно завершены, адаптируйте систему в загрузочных секторах, чтобы отразить необходимые начальные настройки с помощью BPCNFG (см. [6.2](#_6.2_BPCNFG_-)). Такие элементы, как желаемое имя файла автозагрузки, номера банков (критически важно, если используется FASTWB) и используемые типы дисков обычно настраиваются в этом месте. Когда вы закончили этот шаг, протестируйте свою новую систему, нажав кнопку сброс или выключив и включив питание и у вас все должно заработать!

## 4.2 Установка образа без коммутации банков памяти

Системы без коммутации банков могут быть установлены в виде файла с образом, в отличие от базовой установки на загрузочные дорожки, описанной выше в [Разделе 4.1](#_4.1_Установка_на). Для создания файла с образом, вы должны иметь REL или ZRL версии процессора команд (рекомендуется ZCPR3.x или аналогичный), операционной системы (рекомендуется ZSDOS.ZRL), и версии REL B/P BIOS для вашей системы со значением параметра MOVCPM установленным NO в файле DEF-xx.LIB. Другие параметры в этом файле могут быть установлены, как описано выше для системы на загрузочных дорожках. Так как файлы образов, не сильно ограничены в размерах, как при установке на загрузочные дорожки, как правило, больше компонентов может быть активировано, таких как сообщения об ошибках, виртуальный диск, дополнительные разделы жесткого диска и дополнительные форматы дискет. В этом варианте основным недостатком является использование больших жестких дисков, которые значительно сокращают область транзитных программ, так как все параметры диска должны располагаться в защищенной верхней памяти выше BIOS.

После того, как BIOS скомпилирована, должен быть создан файл с образом. Это достигается с помощью утилиты BPBUILD (см. [6.1](#_6.1_BPBUILD_-_1)). Задайте имена файлов в меню 1, чтобы отразить только файлы, без переключения банков (или минимальный BIOS с переключением банков, если параметр FASTWB установлен в YES), и пусть BPBUILD делает свою работу. Поскольку стандартные сегменты системы без переключения банков, как правило, устанавливается для "стандартных" CP/M 2.2 размеров, вы можете ответить “Y” на запрос “autosize”, для получения максимальной области транзитных программ в результирующей системе. Когда BPBUILD завершит свою работу, файл, обычно с расширением по умолчанию .IMG, будет помещен на текущий диск/область пользователя и вы готовы выполнить следующий шаг для подготовки образа без коммутации банков.

Как в случае установки на загрузочные дорожки, рассмотренном выше, должны быть настроены несколько системных параметров, прежде чем образ сможет быть благополучно загружен и выполнен. Это осуществляется путем вызова BPCNFG с указанием имени файла с образом в качестве параметра, или указанием настраиваемого образа в интерактивном меню (см. [6.2](#_6.2_BPCNFG_-_1)). Установите все элементы которые вы хотите настроить в операционной системе, в частности, номера банков (если активен FASTWB), параметры дисков и присвоения. Когда это успешно завершено, вы готовы, загрузить и выполнить только что созданную систему.

Установка файла с образом (по умолчанию расширение файла .IMG) чрезвычайно проста (см. [6.15](#_6.15_LDSYS_Загрузчик_2)). Необходима только утилита LDSYS.COM. Если расширение файла по умолчанию .IMG не было изменено, только имя файла с образом должно передаваться LDSYS при выполнении:

LDSYS IMGFILE <-- где IMGFILE является именем вашего файла с образом

Рабочие параметры исполняющейся в настоящий момент системы сначала проверят возможность загрузки файла с образом. Если есть возможность продолжить, файл с образом загружается, размещаясь в соответствующих ячейках памяти, и дается команда начать выполнение с помощью вызова вектора холодного старта B/P BIOS. Холодный старт ([Функция 0](#Ф0) BIOS) выполняет заключительную установку, выводит на экран любое желаемое приглашение и передает управление процессору команд с любым указанным файлом автозагрузки для использования замены процессора команд ZCPR3.x.

Так как файл с образом системы без коммутации банков, вероятно, имеет небольшие отличия по сравнению с содержимым загрузочных секторов, обычно тот же файл автозагрузки может использоваться, для завершения настроек начальной инициализации. Если требуется использовать другой файл, файл с образом может быть изменен, указанием иного файла с помощью BPCNFG.

## 4.3 BIOS с переключением банков, установка системы без переключения

С помощью системы B/P BIOS может быть создан и загружен образ системы, который помещает часть BIOS только в системный банк, сохраняя операционную систему без переключения банков памяти и поэтому максимально совместимую с существующими прикладными программами. Таким образом, обычно несколько тысяч байт может быть освобождено для транзитных программ, хотя большие диски и/или увеличение числа логических дисков по-прежнему уменьшает пространство TPA из-за необходимости хранения информации вектора распределения в общей памяти.

Для подготовки такой системы, при необходимости, просто отредактируйте нужные файлы BIOS, уделив особое внимание файлу DEF-xx.LIB, в котором должны быть установлены следующие параметры:

MOVCPM Установите NO

BANKED Установите YES

ZSDOS2 Установите NO

Поскольку переключаемые банки памяти доступны для этого типа установки, вы, возможно, захотите активизировать доступную функцию “Fast Warm Boot” (быстрый теплый старт) для увеличения производительности системы. Чтобы активировать эту возможность, установите следующий параметр:

FASTWB Установите YES

Когда редактирование завершено, скомпилируйте BIOS в файл Microsoft .REL с помощью подходящего ассемблера, например ZMAC и создайте образ системы с помощью BPBUILD (см. [6.1](#_6.1_BPBUILD_-_2)), измените имя файла BIOS в меню 1 на имя недавно созданного файла BIOS. Затем, при необходимости, настройте условия по умолчанию с помощью BPCNFG (см. [6.2](#_6.2_BPCNFG_-_2)), и вы готовы активировать новую систему способом пригодным для всех файлов с образами, вызывая LDSYS с параметром файла с образом:

LDSYS BBSYS <-- где BBSYS имя вашего файла с образом

Как для системы не использующей переключение банков, описанной выше в [Разделе 4.2](#_4.2_Установка_образа), для файла автозагрузки, который используется для начальной инициализации при загрузке системы, новые требования не установлены, так как процессор команд и дисковая операционная система не поддерживают переключение банков, и области данных, необходимые прикладным программам, не помещаются в *системный банк*. Как для любых файлов с образами, дополнительные функции, такие как расширенные сообщения об ошибках BIOS, более полный перечень форматов дискет и виртуальный диск могут быть активированы в настройках системы перед компиляцией, поскольку отсутствуют ограничения размера, присущие системам для размещения на загрузочных дорожках.

## 4.4 Установка образа с полной поддержкой коммутации банков памяти

Для создания системы, максимально использующей преимущества переключаемой памяти, необходимы специальная операционная система с поддержкой коммутации банков и процессор команд. Они были представлены в первоначальной версии этого пакета в виде файлов ZSDOS20.ZRL и Z40.ZRL соответственно. Они используют функции переключения банков B/P BIOS и размещают максимально возможное количество исполняемого кода и данных в системном банке. Важным фактором для увеличения области транзитных программ является размещение битовых массивов распределения диска в системном банке, которое означает, что добавление больших жестких дисков, или нескольких дисков требует лишь минимального расширение резидентной части BIOS.

BIOS с полной поддержкой переключения банков создается путем редактирования файлов B/P BIOS по мере необходимости, для настройки системы под ваши желания. Убедитесь, что параметры в DEF-xx.LIB установлены следующим образом:

MOVCPM Установите NO

BANKED Установите YES

ZSDOS2 Установите YES

Скомпилируйте полученный B/P BIOS в файл в формате Microsoft .REL, создайте файл с образом с помощью BPBUILD (см. [6.1](#_6.1_BPBUILD_-_3)), и настройте полученный файл с образом используя BPCNFG (см. [6.2](#_6.2_BPCNFG_-_3)). Если вы уверены, что все настройки по умолчанию установлены, активируйте файл, введя:

LDSYS FBANKSYS <-- где FBANKSYS имя вашего файла с образом

Некоторые различия могут существовать в файле автозагрузки (Startup), используемом в системе с полной поддержкой переключения банков. Как правило, изменения означают удаление некоторых элементов, таких как модуль записи временных меток файла для ZSDOS1 без коммутации банков, который не требуется ZSDOS2 и Z40 с полной поддержкой переключения банков. Для ZSDOS2 необходимо указать только тип часов. Кроме того, поскольку замена процессора команд Z40 содержит наиболее часто используемые команды, собранные из многих пакетов резидентного процессора команд (RCP), обычно не требуется загрузка RCP. Ниже представлен простой файл автозагрузки, содержащий все необходимое во время разработки B/P системы с полной поддержкой переключения банков:

|  |  |  |
| --- | --- | --- |
| ZSCFG2 CB | <-- | Установка ZSDOS 2 часов BIOS+4EH |
| LDR SYS.FCP,SYS.NDR | <-- | Загрузка сегментов среды ZCPR 3 для управления потоками и именами каталогов |
| IOPINIT | <-- | Инициализация пакета процессора ввода-вывода |
| TD S | <-- | Подсказка для даты и времени, в качестве альтернативы установки времени можно использовать TDD ([6.21](#_6.21_TDD_Утилита_1)) или SETCLOK ([6.18](#_6.18_SETCLOK__1)) |
| IF ~EX MYTERM.Z3T | <-- | Если файл MYTERM.Z3T не существует... |
| TCSELECT MYTERM.Z3T | <-- | …выберите, какой терминал у вас есть, создание файла MYTERM.Z3T |
| FI | <-- | ... конец IF |
| LDR MYTERM.Z3T | <-- | Загрузка заданных определений терминала |

Поскольку требования к системе с полной поддержкой коммутации банков значительно отличаются от требований к системе без коммутации банков, мы рекомендуем вам использовать другое имя для файла автозагрузки. Например, STARTUP.COM это имя по умолчанию, используемое в системах на загрузочных дорожках для начальной инициализации и файлов с образами без коммутации банков, в то время как STARTB может быть подходящим именем для сценария выполняющемся при загрузке системы с полной поддержкой коммутации банков. Имя необходимого файла автозагрузки может быть легко изменено в системах на загрузочных дорожках или в образе системы с помощью выбора 1 в BPCNFG (см. [6.2](#_6.2_BPCNFG_-_4)).

Вариант, доступный для запуска из файла большого образа, состоит в настройке файла автозапуска для выполнения системой с загрузочной дорожки, содержащего единственную команду. Команда просто вызывает LDSYS с необходимым файлом с образом с переключением банков в качестве параметра:

LDSYS BANKSYS <-- где BANKSYS.IMG имя вашего файла с образом

В этом случае последовательность команд инициализации, перечисленных выше, не выполняются системой на загрузочной дорожке, и выполнятся только содержащиеся в файле автозапуска для BANKSYS.IMG. Другие варианты имеются в большом количестве и оставлены сообществу изобрести новые комбинации и последовательности.

## 4.5 В случае возникновения проблем...

Хотя мы попытались изложить процедуры для большинства установок, которые мы считали осуществимыми, могут быть случаи, когда вы случайно окажетесь в ситуации, когда вам, кажется, что вы потеряли возможность поднять систему и заставить ее работать.

**Проблема**: при загрузке .IMG файл с помощью LDSYS, на экране отображается баннер LDSYS, адреса системы, и останавливается с последним экраном, отображающем:

"...loading banked system"

**Решение**: Что-то неправильно настроено в BIOS, поскольку все строки после последней отображаемой печатаются из вновь загруженной BIOS. Одной из наиболее распространенных причин возникновения этой проблемы является неправильные настройки номера банка. Используйте скрытый выбор 1 в меню BPCNFG (см. [6.2](#_6.2_BPCNFG_-_5)), чтобы убедиться, что правильные номера банков были установлены для TPA и системных банков. Другой распространенной причиной возникновения этой проблемы является неправильная настройка порта консоли, или параметра IOBYTE, который направляет данные консоли в устройства не предназначенные для этого. Используйте меню 2 BPCNFG для правильной установки IOBYTE и параметров консоли.

**Проблема**: Вы загружаетесь с или загружаете систему B/P BIOS с жесткого диска, и сразу после запуска, система пытается обратиться к дисководу 0.

**Решение**: Наиболее распространенной причиной этого симптома является то, что искомый жесткий диск и гибкий диск не были поменяны местами, чтобы определить раздел жесткого диска, в качестве дисковода A:. Используйте BPCNFG (см. [6.2](#_6.2_BPCNFG_-_6)), меню 5 для обмена дисков в нужной конфигурации. Аналогичная ситуация может возникать, если сразу после загрузки активирован жесткий диск, когда необходим дисковод для гибких дисков как диск A:.

**Проблема**: компьютер, кажется, загружается удовлетворительно, но после выполнения нескольких программ или любой программы, которая выполняет "Теплый" старт (или ввод Ctrl-C), система входит в "Вымышленную страну" и должна быть перегружена кнопкой сброс.

**Решение**: эта проблема чаще всего вызвана невозможностью открыть и загрузить процессор команд. Наиболее вероятно это вызвано компиляцией B/P BIOS с присвоением параметру FASTWB в DEF-xx.LIB значения YES, когда система не имеет расширенной памяти, или неправильно настроены номера банков. Проверьте настройки номеров банков, используя скрытую функцию BPCNFG, меню 1 (см. [6.2](#_6.2_BPCNFG_-_7)).

**Проблема**: При холодном старте с жесткого диска (от питания или сброса), система переходит в дисковод перед отображением знак первоначальных сообщений, и остается зарегистрированной на дискете.

**Решение**: Это происходит чаще всего из-за вашего упущения запустить утилиту HDBOOT на систему загружаемую с жесткого диска после записи ее с помощью BPSYSGEN. Как правило системы созданные с помощью MOVxSYS содержат загрузочный сектор гибкого диска, который загрузит первоначальную операционную систему с дискеты. Утилита HDBOOT (см. [6.9](#_6.9_HDBOOT_-_1)) изменяет эту запись на указанном устройстве жесткого диска, таким образом, что операционная система загружается с жесткого диска. Запустите HDBOOT на нужный жесткий диск, а затем используйте BPCNFG (см. [6.2](#_6.2_BPCNFG_-_8)), чтобы убедиться, что логические диски позиционируются как надо (меню 5).

**Проблема**: При начальной загрузке системная консоль или ничего не выводит на экран или распечатывает странные символы.

**Решение**: Чаще всего это происходит из-за неправильных настроек параметров текущей консоли, по всей вероятности скорости передачи данных или тактовой частоты процессора. Загрузите хорошую систему, затем используйте BPCNFG (см. [6.2](#_6.2_BPCNFG_-_9)) для коррекции настроек параметров проблемной системы. Обратите особое внимание на меню 1 (тактовая частота процессора) и меню 2 (IOBYTE и скорости передачи данных последовательного порта).

**Проблема**: При запуске системы ZSDOS 2 с полной поддержкой переключения банков, некоторые программы «зависают» или «блокируют» систему при выходе.

**Решение**: Одним из наиболее распространенных источников этого симптома является прикладная программа, в которой автор использовал код, который предполагает, что BDOS и процессор команд имеют определенный размер, или имеют фиксированные ссылки на адреса страницы 0. Они наиболее часто могут возникнуть при использовании образов систем, построенных при ответе YES на запрос Autosizing в BPBUILD (см. [6.1](#_6.1_BPBUILD_-_4)). Чтобы компенсировать такие неблагополучные программы, вы можете использовать двухступенчатый процесс сборки:

1. Используйте BPBUILD для создания файла IMG, отвечая YES на Autosizing при выходе. Это максимизирует TPA, размещая резидентную BIOS в памяти как можно выше.
2. Выполните BPBUILD снова с параметром имени, которое Вы только что задали файлу, созданному выше. Это загрузит определения из файла IMG. Сразу выходите с помощью ввода возврата каретки, и ответьте NO на Autosizing и YES для размещения сегментов системы в стандартных местах. Эта процедура сохраняет адреса BIOS неизменными, но переместит начальные адреса BDOS и процессора команд вниз, если это возможно, чтобы моделировать "стандартные" размеры, используемые в CP/M 2.2.

# 5. Программирование для B/P BIOS

Для большинства существующих задач, программирование для B/P BIOS ничем не отличается от стандартного для BIOS СP/М 2.2. Даже адаптация программ СP/М 3 к B/P BIOS не должна представлять особой сложности благодаря близкому сходству с соответствующими расширенными функциями. Однако, мощность и современный интерфейс B/P BIOS, заключается в использовании комбинированных функций для создания переносимого программного обеспечения в широком спектре аппаратных платформ, используя все функции B/P BIOS во взаимодействии. В этом разделе описываются интерфейсы, доступные для системного программиста с использованием B/P BIOS и функции, доступные, чтобы облегчить прямой доступ к гибким и жестким дискам для специализированных программ унифицированным способом.

С нашей точки зрения одним из архитектурных недостатков CP/M Plus был странный способ реализации прямого доступа BIOS. Мы разработали B/P BIOS по возможности максимально совместимой с CP/M 2.2, но обеспечили расширенную функциональность, необходимую в приложениях с использованием переключения банков памяти. Для этого, интерфейс прямых вызовов BIOS как можно близко следует соглашениям CP/M 2.2.

Следующие страницы по программированию предполагают некоторое знакомство с базовыми основами CP/M и знание языка ассемблер для Z80/Z180, поскольку это выходит за рамки намерений данного руководства, и наших литературных навыков, представить учебник по программированию на ассемблере. Если вам нужна дополнительная помощь в этой области, пожалуйста, обратитесь к аннотированной библиографии для ссылок на соответствующие руководства.

## 5.1 Таблица переходов BIOS

Таблица переходов BIOS состоит из 40 переходов к различным функциям BIOS и обеспечивает основную функциональность. Она включает в себя полный перечень точек входа CP/M 2.2, большую часть CP/M 3 (или CP/M Plus) (хотя некоторые отличаются порядком параметров и/или использованием регистров), и новые точки входа, необходимые для поддержки переключаемых банков памяти унифицированным и логическим способом.

Точки входа BIOS состоят из таблицы абсолютных 3-байтовых переходов в начале исполняемого образа. Параметры передаются в BIOS в регистрах, необходимых для конкретной операции. Чтобы избежать проблем совместимости с будущими версиями, некоторые из основных правил для использования вызовов BIOS включают: альтернативные или индексные регистры при вызове BIOS не должны изменяться и все регистры перечисленные в документации сохраненные и незадействованные должны быть возвращены в вызывающую программу в состоянии до вызова. Точки входа в BIOS следующие:

|  |  |  |  |
| --- | --- | --- | --- |
| **Функция 0**(xx00) | | **Холодный старт** | |
| Вход: | Нет | Вывод: | Нет  Выполнение восстанавливает CPR |
| Использование: все регистры | |

Выполняет инициализацию "Холодный" старт при первом выполнении. Параметр перехода позже перезаписывается и указывает на таблицу переходов устройства IOP. Причина этого заключается в том, что код, выполняющий инициализацию часто помещают в области памяти, которые впоследствии используются для хранения системной информации с целью сохранения памяти. Попытка повторного выполнения кода инициализации в этом случае натолкнётся на байты данных вместо исполнимых инструкций, и система гарантированно "откажет".

Среди других функций, во время начального выполнения кода холодного старта осуществляются: создание начальной среды Z3, если это необходимо, инициализация любых системных сегментов Z3, таких как расширенный путь, пакет управления потоком, буфер имен каталогов, а так же: настройка специфичных для системы значений, таких как размещение буферов вектора распределения для виртуальных и жестких дисков и выполнение подпрограммы инициализации устройств (см. Функцию 21). Подпрограмма холодного старта обычно выходит путем сцепления с функцией теплого старта (Функция 1) для установки векторов на странице 0 в банке памяти TPA.

|  |  |  |  |
| --- | --- | --- | --- |
| **Функция 1**(xx03) | | **Теплый старт** | |
| Вход: | Нет | Вывод: | Нет  Выполнение возвращается к CPR |
| Использование: все регистры | |

Эта функция повторно инициализирует операционную систему и возвращается к процессору команд после его перезагрузки с загрузочных дорожек диска по умолчанию или переключаемой памяти, если BIOS была собрана с параметром Fast Warm Boot (быстрый теплый старт).

Если не изменен некорректно ведущим себя резидентным системным расширением (RSX) или другой работающей транзитной программой, вектор "Теплого" старта в памяти по адресу 0 указывает на этот вектор. Корректные программы не изменяют этот адрес, но должны, вместо этого, изменить адрес назначения вектора перехода в заголовке BIOS. Есть единственное исключение из этого правила в случае NZCOM, где вектор "Теплого" старта указывает на NZBIOS, а не на реальную BIOS. В этом случае, адрес реальной BIOS должен быть определен отдельно (см. Функцию 30).

|  |  |  |  |
| --- | --- | --- | --- |
| **Функция 2**(xx06) | | **Опрос состояния ввода консоли** | |
| Вход: | Нет | Вывод: | A=0FFH если символ готов, NZ  A=0 если символ готов, Z |
| Использование: AF | |

Эта функция возвращает флаг, указывающий, был ли символ введен с устройства консоли, выбранной с помощью IOBYTE на странице 0 банка TPA. Статус возврата часто используется транзитными программами, чтобы определить, пытается ли пользователь запустить или остановить выполнение программы.

|  |  |  |  |
| --- | --- | --- | --- |
| **Функция 3**(xx09) | | **Ввод символа с консоли** | |
| Вход: | Нет | Вывод: | A= маскированный входной символ |
| Использование: AF | |

Эта функция ожидает символ, который будет введен из устройства консоли, выбранной с помощью IOBYTE на странице 0 банка TPA и возвращает его вызывающей подпрограмме. Согласно строгим стандартам CP/M 2.2, должен быть обнулен старший значащий бит входного байта, но это может быть изменено маской ввода для консольного устройства.

|  |  |  |  |
| --- | --- | --- | --- |
| **Функция 4**(xx0C) | | **Вывод символа на консоль** | |
| Вход: | C=Символ для отправки на  консоль | Вывод: | Нет |
| Использование: AF | |

Эта функция отправляет указанный символ на устройство консоли, определенной IOBYTE на странице 0 банка TPA. Перед отправкой символа она будет ждать готовность устройства, при необходимости, и будет маскировать биты, как указано в конфигурации символьного устройства для устройства вывода.

|  |  |  |  |
| --- | --- | --- | --- |
| **Функция 5**(xx0F) | | **Вывод символа на устройство печати** | |
| Вход: | C=Символ для отправки на  устройство печати (принтер) | Вывод: | Нет |
| Использование: AF | |

Эта функция будет отправлять указанный символ на устройство печати (принтер) определенное с помощью IOBYTE на странице 0 банка TPA. Она будет ждать готовность устройства перед отправкой символа, при необходимости, и будет маскировать его как указано в конфигурации символьного устройства для устройства вывода.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Функция 6** (xx12) | | **Вывод символа на вспомогательное устройство** | | |
| Вход: | C=Символ для отправки на  вспомогательное устройство | | Вывод: | Нет |
| Использование: AF | |

Эта функция отправляет указанный символ на вспомогательное устройство, определенное IOBYTE на странице 0 банка TPA. Перед отправкой символа она будет ждать готовность устройства, при необходимости, и будет маскировать биты, как указано в конфигурации символьного устройства для устройства вывода.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Функция 7**(xx15) | | **Ввод символа со вспомогательного устройства** | | |
| Вход: | Нет | | Вывод: | A= маскированный входной символ |
| Использование: AF | |

Эта функция прочитает символ со вспомогательного устройства ввода, определенного с помощью IOBYTE на странице 0 банка TPA. Она будет ждать символ, который будет получен и маскирует его, как определено в конфигурации символьного устройства для устройства ввода.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Функция 8**(xx18) | | **Установка головки на 0 дорожку** | | |
| Вход: | Нет | | Вывод: | Нет. Головки выбранного диска перемещаются к дорожке 0 |
| Использование: Все основные регистры | |

Эта функция перемещает головку на выбранном диске на 0 дорожку. В B/P BIOS эта операция не выполняет полезных действий и является просто возвратом. Ожидание записи и позиционирование головок обрабатывается отдельным драйвером устройства (для выбранного диска).

|  |  |  |  |
| --- | --- | --- | --- |
| **Функция 9** (xx1B) | | **Выбор логического диска** | |
| Вход: | C= Необходимый диск (A..P) | Вывод: | (успех) A <> 0, NZ  HL = Адрес DPH  (нет диска) A = 0 Ноль (Z)  HL = 0 |
|  |  | Использование: Все основные регистры | |

Эта функция выбирает указанный логический диск в качестве текущего диска, к которому относятся дисковые операции. Если операция выполнена успешно, возвращается адрес *заголовка параметров диска* (DPH) для последующего определения параметров устройства. Если операция невозможна по любой причине (несуществующий диск, неизвестные или плохие носители, и т.д., и т.п.), возвращается указатель с нулевым значением, означающий, что к диску нельзя получить доступ через BIOS.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Функция 10**(xx1E) | | **Выбор дорожки** | | |
| Вход: | BC = Номер нужной дорожки | | Вывод: | Нет. Сохраняется номер дорожки |
| Использование: Регистры не используются | |

Эта функция сохраняет указанный логический номер дорожки для будущих дисковых операций. Последнее значение, сохраненное с помощью этой функции будет использоваться при чтении и записи с диска.

*Примечание*: Хотя для этой функции определено 16-разрядное значение, в большинстве драйверов используется только младший байт (8 бит).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Функция 11**(xx21) | | **Выбор сектора** | | |
| Вход: | BC = Номер нужного сектора | | Вывод: | Нет. Сохраняется номер сектора |
| Использование: Регистры не используются | |

Эта функция сохраняет указанный номер логического сектора для будущих дисковых операций. Последнее значение, сохраненное с помощью этой функции будет использоваться при чтении и записи с диска.

*Примечание*: Хотя для этой функции указано 16-разрядное значение, во всех драйверах гибких дисков и большинстве драйверов жестких и виртуальных дисков используется только младший байт (8 бит).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Функция 12**(xx24) | | **Установка адреса DMA для передачи** | | |
| Вход: | BC = Адрес начала буфера | | Вывод: | Нет. Сохраняется адрес DMA |
| Использование: Регистры не используются | |

Эта функция сохраняет указанный адрес для использования в качестве источника или назначения для будущих дисковых операций. Последнее значение, сохраненное с помощью этой функции будет использоваться при чтении и записи с диска. В системах с переключением страниц, выбранный для передачи банк может быть изменен Функцией 28.

|  |  |  |  |
| --- | --- | --- | --- |
| **Функция 13**(xx27) | | **Чтение выбранного сектора** | |
| Вход: | Нет | Вывод: | A = 0, Z если нет ошибок  A = не ноль, NZ если есть ошибки |
| Использование: Все основные регистры | |

Эта функция читает 128-байтный логический сектор с дорожки диска, установленный Функциями 9-11 в буфер расположенный по адресу определенный с помощью Функции 12. При возврате, регистр A=0, если работа была успешной, ненулевое значение, если произошла ошибка.

|  |  |  |  |
| --- | --- | --- | --- |
| **Функция 14**(xx2A) | | **Запись выбранного сектора** | |
| Вход: | C = 1 для немедленной записи  C = 0 для записи с буферизацией | Вывод: | A = 0, Z если нет ошибок  A = не ноль, NZ если есть ошибки |
| Использование: Все основные регистры | |

Эта функция записывает 128-байтный логический сектор на дорожке диска, установленный Функциями 9-11 в буфер расположенный по адресу определенный с помощью Функции 12. Если регистр C=1, будет выполнена непосредственная запись и сброс буферов BIOS. Если C=0, запись может быть задержана до деблокирования.

|  |  |  |  |
| --- | --- | --- | --- |
| **Функция 15**(xx2D) | | **Опрос состояния устройства печати** | |
| Вход: | Нет | Вывод: | A=0FFH, NZ если готов принять символ  A=0, Z если принтер занят |
| Использование: AF | |

Эта функция возвращает флаг, указывающий, готов ли принтер принять символ. Она использует IOBYTE на странице 0 банка TPA, чтобы определить какое физическое устройство доступно.

|  |  |  |  |
| --- | --- | --- | --- |
| **Функция 16**(xx30) | | **Преобразование номера сектора** | |
| Вход: | BC = Логический номер сектора  DE = Адрес таблицы преобразования | Вывод: | HL = Физический номер сектора |
| Использование: Все основные регистры | |

Эта функция переводит логический номер сектора в регистре BC (в настоящее время используется только C) в физический номер сектора, используя таблицу преобразования, полученную из DPH расположенного по адресу в DE.

Здесь заканчивает часть таблицы переходов BIOS строго совместимая с CP/M 2.2. Следующий ряд входов примерно следует переходам используемым в CP/M 3, но с исправлением того, что по нашему мнению, имело недостатки и несоответствия в параметрах вызова и структурах.

|  |  |  |  |
| --- | --- | --- | --- |
| **Функция 17**(xx33) | | **Опрос состояния вывода консоли** | |
| Вход: | Нет | Вывод: | A=0FFH, NZ если консоль готова принять символ  A=0, Z если нет символа |
| Использование: AF | |

Эта функция возвращает флаг, указывающий, готово ли консольное устройство, выбранное IOBYTE на странице 0 банка TPA, принять новый выходной символ.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Функция 18**(xx36) | | **Опрос состояния дополнительного устройства ввода** | | |
| Вход: | Нет | | Вывод: | A=0FFH, NZ если ввод AUX ожидает символ  A=0, Z если символ не готов |
| Использование: AF | |

Эта функция возвращает флаг, указывающий, находится ли устройство вспомогательного ввода, выбранное с помощью IOBYTE на странице 0 банка TPA, в ожидании символа.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Функция 19**(xx39) | | **Опрос состояния дополнительного устройства вывода** | | |
| Вход: | Нет | | Вывод: | A=0FFH, NZ если готово принять символ  A=0, Z если устройство AUX занято |
| Использование: AF | |

Эта функция возвращает флаг, указывающий, готово ли вспомогательное устройство вывода, выбранное с помощью IOBYTE на странице 0 банка TPA принять новый символ для вывода.

|  |  |  |  |
| --- | --- | --- | --- |
| **Функция 20**(xx3C) | | **Вернуть адрес таблицы устройств IO** | |
| Вход: | Нет | Вывод: | HL = Адрес таблицы устройств IO |
| Использование: HL | |

Эта функция примерно соответствует аналогичной функция CP/M Plus несмотря на то, что несколько отличаются определения битов. Символ таблицы IO состоит из четырех устройств: СОМ1, СОМ2, PIO и NUL. Каждое имеет маску ввода и вывода, настройки скорости передачи данных и флаги протокола. Не все заданные настройки (например, ACK/NAK и XON/XOFF сигналы квитирования и т.д.) могут быть полностью реализованы в каждой версии, но доступны для дальнейшего расширения и использования.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Функция 21**(xx3F) | | **Инициализация устройств ввода-вывода** | | |
| Вход: | Нет | | Вывод: | Нет. Выполнение инициализации |
| Использование: Все основные регистры | |

Эта функция инициализирует настройки символьного ввода-вывода и других функций, которые могут быть изменены с помощью утилиты конфигурирования. Это расширенная версия соответствующей функции СР/М Plus. Ее основное предназначение восстановить конфигурацию IO, системные параметры, такие как тактовая частота, состояния ожидания, и т.д., после изменения программами, которые имеют прямой доступ к аппаратным средствам, такими как многие программы модемов и утилита конфигурации, BPCNFG (см. [6.2](#_6.2_BPCNFG_-_10)).

|  |  |  |  |
| --- | --- | --- | --- |
| **Функция 22**(xx42) | | **Вернуть адрес таблицы DPH** | |
| Вход: | Нет | Вывод: | HL = Адрес таблицы DPH |
| Использование: HL | |

Эта функция возвращает указатель в таблице 16-битных указателей заголовков параметров дисков для дисков A-P. Запись Null (0000H) означает, что диск не определен в этой логической позиции.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Функция 23**(xx45) | | **< Зарезервирована для чтение-записи нескольких секторов>** | | |
| Вход: | Нет | | Вывод: | Нет |
| Использование: Регистры не используются | |

Эта функция зарезервирована в начальном выпуске B/P BIOS и просто осуществляет возврат.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Функция 24**(xx48) | | **Сброс буферов на диски** | | |
| Вход: | Нет | | Вывод: | Нет. Запись данных из буферов |
| Использование: Все основные регистры | |

Эта функция записывает все отложенные данные на диск из разблокированных буферов, как указано в Функции 14 выше. Эта функция должна вызываться в критических областях, при переключении задач, или смене носителей, если существует вероятность, что операционная система не обнаружит изменения.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Функция 25**(xx4B) | | **Выполнить возможное межбанковое перемещение** | | |
| Вход: | HL=Адрес источника DE=Адрес назначения BC=Число перемещаемых байт | | Вывод: | Нет. Перемещение данных |
| Использование: Все основные регистры | |

Эта функция перемещает конкретное количество байт между указанными адресами. Для межбанковых перемещений банки источника и назначения, должны быть, ранее определены с помощью вызова XMOVE (Функция 29). Обратите внимание, что реализация этой функции B/P инвертирует использование регистровых пар DE и HL по сравнению с эквивалентной функцией CP/M 3.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Функция 26**(xx4E) | | **Получить/Установить дату и время** | | |
| Вход: | DE=Адрес буфера из 6 байт  C=0 (получить время и дату)  C=1 (установить время и дату) | | Вывод: | A = 1 успешно A = 0 если ошибка или нет часов |
| Использование: Все основные регистры | |

Эта функция обеспечивает программам интерфейс для драйвера часов реального времени BIOS. Функция использует 6-байтовую строку даты/времени в формате ZSDOS, отличающуюся от формата Digital Research, используемого в CP/M Plus для этой функции. Кроме того, эта функция должна соответствовать дополнительным требованиям DateStamper™ при выходе, регистр E должен содержать информацию расположенную по адресу (DE+5), и HL должен указать на содержимое (DE)+5. Если фактическая аппаратная реализация часов поддерживают изменение 1/10 секунды, текущее значение счетчика 1/10 секунд может быть возвращено в регистре D.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Функция 27** (xx51) | | **Выбор банка памяти** | | |
| Вход: | A= Требуемый банк памяти | | Вывод: | Нет Банк подразумевает диапазон 0..7FFFH |
| Использование: AF | |

Эта функция выбирает банк памяти, указанный в регистре А и активизирует его в диапазоне адресов 0-7FFFH. Поскольку символ IO может использоваться, при выборе банка с помощью этой функции отличного от TPA (который содержит IOBYTE), B/P BIOS автоматически получает IOBYTE из банка TPA, чтобы обеспечить, соответствие символа IO с нужными устройствами.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Функция 28** (xx54) | | **Выбор банка памяти для DMA** | | |
| Вход: | A= Банк памяти для передачи DMA | | Вывод: | Нет. Номер банка сохраняется для последующих операций IO |
| Использование: Регистры не используются | |

Эта функция выбирает банк памяти, с которым осуществляются дисковые операции IO. Функция 12 (Установка адреса DMA для передачи) действует в сочетании с этим выбором для дисковых операций IO.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Функция 29**(xx57) | | **Установка банков источника и назначения для перемещения** | | |
| Вход: | B=Номер банка назначения  C= Номер банка источника | | Вывод: | Нет. Номера банков сохраняются для осуществления перемещения |
| Использование: Все основные регистры | |

Эта функция устанавливает номера банков источника и назначения для последующего перемещения (Функция 25). После того, как перемещение выполнено, банки источника и назначения автоматически сбрасываются к значениям банка TPA.

Здесь заканчиваются переходы "типа" CP/M Plus и начинаются уникальные дополнения к таблице переходов B/P BIOS, для поддержки переключения банков, прямого IO и взаимодействие через интерфейс.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Функция 30**(xx5A) | | **Возвращение адреса BIOS** | | |
| Вход: | Нет | | Вывод: | A = Номер версии BIOS (Hex) BC = Начальный адрес BIOS DE = Адрес конфигурации BIOS HL = Адрес таблицы устройств |
| Использование: Все основные регистры | |

Эта функция возвращает различные указатели на внутренние области данных BIOS и номер версии BIOS, как обозначено выше. Версия BIOS может использоваться, чтобы определить применимость системного программного обеспечения и используется различными утилитами поддержки, для минимизации возможности повреждения данных и/или как индикатор поддерживаемых функций.

Начальный адрес таблицы переходов BIOS, возвращаемый в регистре BC, часто используется для обеспечения правильной индексации структур данных B/P, если "оболочка" BIOS была добавлена, например, при запуске NZCOM. Несмотря на то, что переход “Теплый старт” расположенный по адресу 0000H, обычно указывает на BIOS BASE+3, это не всегда надежно, тогда как эта функция всегда возвращает истинное значение B/P BIOS.

Регистры DE и HL возвращают указатели, которые имеют значение для программ, которые изменяют или настраивают различные параметры BIOS. Указатель на область конфигурации BIOS должен использоваться в утилитах в отличие от индексации от самого начала таблицы переходов BIOS, так как дополнение таблицы переходов или внесение других данных повлияет на начальный адрес области конфигурации. Указатель в HL доступен для использования в системах, которые могут содержать более четырех устройств в символе ввода-вывода. Этот указатель позволяет осуществлять обмен устройств, чтобы разместить нужные устройства в первых четырех позициях таблицы, делая их доступными для выбора через IOBYTE. После того, как внесены любые изменения, связанные с устройствами, с помощью вызова Функции 21 BIOS инициализации устройств IO следует активировать функцию.

**ПОДФУНКЦИИ ГИБКОГО ДИСКА**

Функция 31 позволяет низкоуровневый доступ к гибким и жестким дискам (через SCSI-интерфейс), указав тип драйвера и нужную функцию. Хотя некоторые типы оборудования не поддерживают все указанные параметры, в частности для гибких дисков, данная архитектура поддерживает все типы, хотя конкретные системы могут игнорировать определенные функции. Таким образом программа поддерживает единый формат, например, для NEC765, SMC9266, WD1770/1772/179x и других типов контроллеров с разными интерфейсами. Функции гибких дисков осуществляется путем ввода значения 1 в регистр B (тип флоппи-драйвера) и требуемого номера функции в регистр C, затем перехода к, или вызова BIOS входа перехода с номером 31.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Функция 31**(xx5D)  Подфункция дискеты 0 | | **Задать режим чтения/записи контроллера гибких дисков** | | |
| Вход: | A = 0 Двойная плотность  FF Одинарная плотность  B = 1 (Гибкий диск)  С = 0 (Номер подфункции) | | Вывод: | Нет |
| Использование: AF | |

Эта процедура устанавливает режим работы плотности контроллера дискеты для чтения и записи данных. Она предполагает, что подфункции 1 (установить размер и мотор) и 3 (установить сектор) вызываются первыми.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Функция 31**(xx5D)  Подфункция дискеты 1 | | **Установить размер дискеты и параметры мотора** | | |
| Вход: | A = 0 300 об/мин (нормальный)  FF 360 об/мин (8"/HD)  D = 0 (Мотор включен)  FF (Управление мотором)  E = 1 Размер диск   B = 1 (Гибкий диск)  С = 1 (Номер подфункции) | | Вывод: | Нет |
| Использование: AF | |

Эта процедура устанавливает некоторые из физических параметров для дисковода гибких дисков. Обычные 5.25" и 3.5" дисководы емкостью 400 или 800 Кб или менее вращается со скоростью 300 об/мин. Многие из новых дисководов могут увеличить скорость до 360 об/мин, эту скорость, используют старые 8" дисководы. Эта скорость используется для 5.25" дисководов высокой плотности (HD) 1.2 МБ (IBM формат). Регистр A используется для обозначения максимальной скорости доступной на указанном дисководе. Регистр D используется, чтобы указать, является ли двигатель всегда включенным, или он будет периодически запускаться и останавливаться. Обычно BIOS использует это значение для периода задержки, перед записью если двигатель остановился, чтобы позволить ему набрать скорость, тем самым уменьшая вероятность повреждения данных на дискете. Регистр E используется для указания размера физического носителя: 0 = жесткий диск, 001B = 8" дисковод, 010В = 5.25" дисковод и 011B = 3.5". Эта подфункция не возвращает значений.

Хотя не все эти функции могут поддерживаться на любом конкретном компьютере, интерфейс используемый в программах всегда должен передать необходимые параметры для совместимости.

*Примечание*: эта процедура предполагает, что подфункция 2 (выбор головки и привода) вызывается первой. Вызывайте ее перед вызовом функции 0 (установки режима).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Функция 31**(xx5D)  Подфункция дискеты 2 | | **Задать номер головки и номер гибкого диска** | | |
| Вход: | A = Номер диска (биты 0,1)  Номер головки (бит 2)  B = 1 (Гибкий диск)  С = 2 (Номер подфункции) | | Вывод: | Нет |
| Использование: AF | |

Эта процедура вводится с регистром A, содержащим номер устройства гибких дисков закодированный в битах 0 и 1 (устройство 0 = 00, 1 = 01 ... 3 = 11) и головку в бите 2 (0 = головка 0, 1 = головка 1). Эта подфункция не возвращает значений. Вызывайте эту подфункцию прежде чем большинство других для сведения к минимуму проблем с доступом к гибким дискам.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Функция 31**(xx5D)  Подфункция дискеты 3 | | **Задать режим чтения/записи контроллера гибких дисков** | | |
| Вход: | A = Номер физического сектора   D = Размер физического сектора  E = Номер последнего сектора  B = 1 (Гибкий диск)  С = 3 (Номер подфункции) | | Вывод: | Нет |
| Использование: AF | |

Эта процедура устанавливает информацию, необходимую для корректного доступа к указанному сектору однозначно для множества различных типов контроллеров. На входе, регистр A содержит необходимый физический номер сектора, D содержит размер сектора, где 0 = сектор 128 байт, 1 = 256 … 3 = 1024 сектор байт и Е содержит номер последнего сектора на стороне. Обычно значение регистра E не используется в контроллерах Western Digital, но необходимо для 765 и 9266. Эта подфункция не возвращает значений.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Функция 31**(xx5D)  Подфункция дискеты 4 | | **Задание временных параметров накопителя** | | |
| Вход: | A = Время шага в мс  D = Время отвода головки после операции чтения/записи в мс  E = Время между подводом головки к поверхности и началом чтения/записи в мс  B = 1 (Гибкий диск)  С = 4 (Номер подфункции) | | Вывод: | A = 0 Время отвода головки после операции чтения/записи в мс |
| Использование: AF | |

Эта подфункция устанавливает различные временные значения, используемых для выбранного физического накопителя. При вызове, регистр A содержит время шага в миллисекундах. В BIOS этот значение округляется до ближайшего времени контроллера, даже если указанное значение не совпадает. Регистр D содержит время отвода головки после операции чтения/записи в миллисекундах, и регистр E время между подводом головки к поверхности и началом чтения/записи в мс.

*Примечание*: В контроллерах типа Western Digital, только время шага является универсальной переменной. В этих системах, время, передаваемое BIOS округляется до ближайшего фиксированного времени шага, например 2, 3, 5, или 6 миллисекунд для WD1772 или 6, 10, 20 или 30 мс, используемых в более старых модификациях WD1770 и WD1795. Эта подфункция не возвращает значений.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Функция 31**(xx5D)  Подфункция дискеты 5 | | **Перемещение головки к дорожке 0** | | |
| Вход: | B = 1 (Гибкий диск)  С = 5 (Номер подфункции) | | Вывод: | A = 0, Z если успешно  A <> 0, NZ если с ошибкой |
| Использование: AF  Важно: Необходимы подфункции 1, 2 и 4 | |

Эта подфункция отвечает за движения головки на выбранном диске к дорожке 0 (домой). Только успех/неудача определяется по значению в регистре A. Другие регистры не могут быть изменены этой подфункцией (в частности BC).

*Примечание*: Эта функция требует, чтобы подфункции 1 (установить размер дискеты и параметры мотора), 2 (задать номер головки и номер гибкого диска) и 4 (задание временных параметров накопителя) были вызваны первыми для определения физических характеристик диска.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Функция 31**(xx5D)  Подфункция дискеты 6 | | **Поиск дорожки** | | |
| Вход: | A = Номер дорожки  D = 0FFH С проверкой  0 Без проверки  E = 0 Один шаг  <> 0 Два шага  B = 1 (Гибкий диск)  С = 6 (Номер подфункции) | | Вывод: | A = 0, Z если успешно  A <> 0, NZ если с ошибкой |
| Использование: AF  Важно: Необходимы подфункции 2, 3 и 4 | |

Эта подфункция перемещает головку(и) для выбранного диска к заданной дорожке на носителе. Если флаг двойного шага (Регистр E) будет установлен в ненулевое значение, то контроллер выдаст два импульса шага для каждого приращения или уменьшения дорожки, который требуется. После поиска чтобы убедиться, что желаемый трек был найден, если флаг проверки (регистр D) установлен в ненулевое значение, предпочтительно 0FFH, будет выполняться функция чтения ID. Только регистры AF изменяются этой подфункцией.

*Примечание*: Эта функция требует, чтобы подфункции 2 (задать номер головки и номер гибкого диска), 3 (задать режим чтения/записи контроллера гибких дисков) и 4 (задание временных параметров накопителя) были вызваны первыми для определения физических характеристик диска.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Функция 31**(xx5D)  Подфункция дискеты 7 | | **Чтение сектора дискеты** | | |
| Вход: | HL = Адрес буфера назначения  B = 1 (Гибкий диск)  С = 7 (Номер подфункции) | | Вывод: | A = 0, Z если успешно  A <> 0, NZ если с ошибкой |
| Использование: AF, HL  Важно: Необходимы подфункции 0, 1, 2, 4 и 6 | |

Этот подфункция считывает физический сектор данных из выбранного диска и помещает его в буфер по указанному адресу. Важно, предоставление буфера соответствующего размера для этой задачи. Значение в регистре A будет указывать, успех или неудачу выполнения функции, как указано в приведенной выше таблице. Только регистры AF и HL могут быть изменены этой функцией.

*Примечание*: Эта функция требует, чтобы подфункции 0 (задать режим чтения/записи контроллера гибких дисков), 1 (установить размер дискеты и параметры мотора), 2 (задать номер головки и номер гибкого диска), 4 (задание временных параметров накопителя) и 6 (поиск дорожки) были вызваны первыми для того чтобы установить физические и логические характеристики передачи данных.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Функция 31**(xx5D)  Подфункция дискеты 8 | | **Запись сектора дискеты** | | |
| Вход: | HL = Адрес буфера источника  B = 1 (Гибкий диск)  С = 8 (Номер подфункции) | | Вывод: | A = 0, Z если успешно  A <> 0, NZ если с ошибкой |
| Использование: AF, HL  Важно: Необходимы подфункции 0, 1, 2, 4 и 6 | |

Эта подфункция записывает данные из буфера, начиная с заданного адреса в сектор на дорожке с помощью головки выбранных другими подфункции. Нулевое значение в регистре A вместе с установкой флага Z указывает, является ли выполнение операции успешным или нет. Только регистры AF и HL могут быть изменены этой функцией.

*Примечание*: Эта функция требует, чтобы подфункции 0 (задать режим чтения/записи контроллера гибких дисков), 1 (установить размер дискеты и параметры мотора), 2 (задать номер головки и номер гибкого диска), 4 (задание временных параметров накопителя) и 6 (поиск дорожки) были вызваны первыми для того чтобы установить физические и логические характеристики передачи данных.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Функция 31**(xx5D)  Подфункция дискеты 9 | | **Считать первую корректную информацию ID на дорожке** | | |
| Вход: | B = 1 (Гибкий диск)  С = 9 (Номер подфункции) | | Вывод: | A = 0, Z если успешно  A <> 0, NZ если с ошибкой |
| Использование: AF  Важно: Необходимы подфункции 0 и 2 | |

Эта подфункция читает первую допустимую информацию заголовка (ID), встретившуюся на дорожке. У этой функции нет входных параметров, за исключением типа драйвера накопителя и номера подфункции. Возвращающийся флаг, указывает наличие или отсутствие ошибки. Ошибка означает, что узнаваемый ID сектора не может быть прочитан на диске. В большинстве случаев, это происходит из-за неправильного настройки плотности в BIOS.

*Примечание*: Эта функция требует, чтобы подфункции 2 (задать номер головки и номер гибкого диска), и 3 (задать режим чтения/записи контроллера гибких дисков) были вызваны в первую очередь для того, чтобы установить физические характеристики диска.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Функция 31**(xx5D)  Подфункция дискеты 10 | | **Вернуть состояние привода** | | |
| Вход: | B = 1 (Гибкий диск)  С = 10 (Номер подфункции) | | Вывод: | A = Байт статуса последней операции  BC = Тип контроллера  HL = Адрес байта статуса |
| Использование: AF, BC, HL  Важно: Необходима подфункции 2 | |

Эта функция возвращает статус текущего выбранного диска. У этой функции нет входных параметров, за исключением типа драйвера накопителя и номера подфункции. На выходе возвращается необработанный без маски байт состояния привода, или последней операции в зависимости от типа контроллера, вместе с двоичным числом, представляющим тип контроллера гибких дисков (например, 765, 9266, 1772, и др.).

*Примечание*: Эта процедура предполагает, что подфункция 2 (задание номер головки и номер гибкого диска), была вызвана вызова этой процедуры, для выбора физических параметров.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Функция 31**(xx5D)  Подфункция дискеты 11 | | **Форматирование дорожки гибкого диска** | | |
| Вход: | HL = Указатель на блок данных  D = Число секторов на дорожке  E = Число байт межсекторного промежутка GAP3  B = 1 (Гибкий диск)  С = 11 (Номер подфункции) | | Вывод: | A = 0, Z если успешно  A <> 0, NZ если с ошибкой |
| Использование: AF, BC, DE, HL  *Примечание*: Необходима подфункция 10 для определения типа контроллера | |

Этот подфункция форматирует полную дорожку на одной стороне дискеты. Она предполагает, что режим, головка/привод, дорожка и сектор уже установлены. На входе HL указывает на данные, требуемые контроллеру для форматирования дорожки. Они отличаются для различных контроллеров, поэтому, нужно вызвать RETDST (подфункцию 10), чтобы определить тип контроллера перед подготовкой структуры данных. Также на входе регистр D должен содержать число секторов на дорожке, и регистр E должен содержать число байт, используемых для промежутка 3 в формате дискеты. На выходе, A = 0 и установленный нулевой флаг (Z), если операция была успешно завершена, A <> 0 и очищенный нулевой флаг (NZ), если произошла ошибка. Эта процедура может изменить все основные регистры (AF, BC, DE, HL).

*Примечание*: Эта процедура предполагает, что подфункция 10 (Вернуть состояние привода) вызывается первой для определения типа контроллера и задания правильной информации в блоке данных формата.

**ПОДФУНКЦИИ ЖЕСТКОГО ДИСКА**

Эти функции доступны для прямого доступа к жестким дискам, подключенным с помощью интерфейса типа SCSI. Они доступны путем загрузки требуемого номера функции в регистр C, Загрузка 2 (драйвер SCSI) в регистре B и перехода к, или вызова BIOS входа перехода с номером 31. Поскольку этот интерфейс не является стандартизированным, как функции гибких дисков в порядке обработки SASI, а также устройств SCSI, интерфейс имеет только основные функции с точными операциями, указанными пользователем в блоке дескриптора команд, передаваемом с функцией 2. Хотя этот способ возлагает большую нагрузку на пользовательские программы, он обеспечивает большую гибкость, при изменении функций для новых дисков SCSI.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Функция 31**(xx5D)  Подфункция жесткого диска 0 | | **Установить адрес области данных для прямого SCSI IO** | | |
| Вход: | HL = Адрес области данных  B = 2 (Жесткий диск)  С = 0 (Номер подфункции) | | Вывод: | A = Число байт в блоке команды |
| Использование: AF | |

Этот подфункция задает адрес области данных пользователя для прямого ввода/вывода SCSI и возвращает число байт, доступных в блок дескриптора команды SCSI. Область данных должен быть длиной минимум 512 байт и используется для хранения данных для записи и для получения данных, чтение из выбранного диска. Этот размер области данных является обязательным, поскольку всегда возвращаются 512 байт от прямого доступа для того, чтобы обрабатывать широкий спектр типов контроллеров, признанных в драйверах B/P BIOS. Число байт, доступных в блоке дескриптора команд в физическом драйвере обычно составляет 10 для обработки расширенных команд SCSI, но может быть сокращено до 6 в ограниченных приложениях.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Функция 31**(xx5D)  Подфункция жесткого диска 1 | | **Задание физической и логический диск** | | |
| Вход: | HL = Байт устройства (5.2.1)  B = 2 (Жесткий диск)  С = 1 (Номер подфункции) | | Вывод: | A = Число байт в блоке команды |
| Использование: AF | |

Эта подфункция устанавливает бит в BIOS для доступа к физическому устройству SCSI и номер логического устройства в блоке команды SCSI (Байт 1, биты 7-5).

Формат байта устройства, предусмотренного для этой процедуры определен в [Раздел 5.2.1](#_5.2.1_Область_конфигурации_3) конфигурация данных, CONFIG+61 и доступен из заголовка расширенных параметров диска DPH-1. На выходе этой процедуры, байт возвращается с битом установленным в единицу в точной позиции (бит 7 = устройство 7... бит 0 = устройство 0) для выбранного устройства с помощью команды SCSI.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Функция 31**(xx5D)  Подфункция жесткого диска 2 | | **Задание физического и логического диска** | | |
| Вход: | DE = Указатель блока дескриптора команд  A = 0 если нет записи данных  A = FF если данные для записи  B = 2 (Жесткий диск)  С = 1 (Номер подфункции) | | Вывод: | A = Бит 1 статус, установленные флаги  H = Значение байта сообщения  L = Значение байта статуса |
| Использование: AF, BC, DE, HL  *Примечание*: Необходимы подфункции 0 и 1 | |

Эта подфункция выполняет действия, необходимые при выполнении этой команды в указанный блоке дескриптора команд. Флаг указанный в регистре A означает должны ли данные пользователя быть записаны этой командой. Если установлено ненулевое значение, данные из области, заданной с помощью функции 0 будут позиционироваться для операции записи SCSI. В конце процедуры 512 байт всегда передаются из буфера ввода-вывода BIOS в указанную область пользователя, установленную подфункцией 0. Это может быть неэффективно, но только таким способом мы смогли обеспечить широкий спектр различных SASI/SCSI контроллеров с ограничением кода в разумных пределах. При завершении этой функции возвращается байт состояния маскированный с помощью бита проверки, 1-й бит. Кроме того для дополнительной идентификации любой ошибки предусмотрены полный байт состояния и байт сообщения SCSI операции.

*Примечание*: Эта процедура предполагает, что блок дескриптора команд правильно настроен для типа контроллера жесткого диска в B/P BIOS, и что выбранный диск правильно описан (при необходимости) в блоке определений BIOS. Возникновении ошибки приводит к выходу из программы и теплому старту. Предполагается, что пользователь вызывает функции 0 (установить адрес области данных для прямого SCSI IO) и 1 (задание физического и логического диска) перед использованием этой подфункции.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Функция 32**(xx60) | | **Установка номер банка** | | |
| Вход: | A = Номер банка | | Вывод: | Нет |
| Использование: Регистры не используются | |

Эта функция задает номер банка для последующей функции 33 перехода к процедуре в другом банке памяти.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Функция 33**(xx63) | | **Перейти к (HL) в альтернативном банке** | | |
| Вход: | HL = Адрес для выполнения в банке заданном функцией 32 | | Вывод: | <Неизвестно> Вызываемая процедура устанавливает статус возврата |
| Использование: Все основные регистры (предположительно) | |

Эта функция активизирует банк памяти номер которого, ранее определен функцией 32, затем вызывает процедуру по адресу размещенному в HL. После завершения, выполнение возвращается в банк из которого она была вызвана, к адресу находящемуся на вершине стека.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Функция 34**(xx66) | | **Очистка переключателя в стеке** | | |
| Вход: | HL = Адрес возобновления выполнение в банке вызова | | Вывод: | Нет. Выполнение возобновляется по адресу в HL в банке вызова. |
| Использование: Регистры не используются | |

Эта функция используется для выхода по ошибке из процедуры в другом банке и возврату в банк вызова.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Функция 35**(xx69) | | **Загрузить A,(HL) из альтернативного банка** | | |
| Вход: | HL = Адрес требуемого байта  С = Номер банка | | Вывод: | A = Байт из C:HL. |
| Использование: AF | |

Эта функция получает байт (8 битов) из указанного банка и адреса. Банк временно переключается в контекст для доступа (при необходимости), затем возвращаются начальные условия. В течение короткого промежутка времени доступа прерывания временно недоступны.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Функция 36**(xx6C) | | **Загрузить DE,(HL) из альтернативного банка** | | |
| Вход: | HL = Адрес требуемого слова  С = Номер банка | | Вывод: | A = Слово из C:HL. |
| Использование: AF, DE | |

Эта функция получает слово (16 битов) из указанного банка и адреса. Банк временно переключается в контекст для доступа (при необходимости), затем возвращаются начальные условия. В течение короткого промежутка времени доступа прерывания временно недоступны.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Функция 37**(xx6F) | | **Сохранить (HL),A в альтернативный банк** | | |
| Вход: | HL = Адрес байта назначения  С = Номер банка  A = Байт для сохранения в C:HL | | Вывод: | Нет |
| Использование: AF | |

Эта функция сохраняет байт (8 битов) по указанным адресу и банку. Банк временно переключается в контекст для доступа (при необходимости), затем возвращаются начальные условия. В течение короткого промежутка времени доступа прерывания временно недоступны

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Функция 38**(xx72) | | **Сохранить (HL),DE в альтернативный банк** | | |
| Вход: | DE = Слово сохраняемое из C:HL  HL = Адрес назначения слова  С = Номер банка | | Вывод: | Нет. Слово помещенное в C:HL |
| Использование: AF | |

Эта функция сохраняет слово (16 битов) в указанные адрес и банк. Банк временно переключается в контекст для доступа (при необходимости), затем возвращаются начальные условия. В течение короткого промежутка времени доступа прерывания временно недоступны

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Функция 39**(xx75) | | **Вернуть номер текущего банка памяти** | | |
| Вход: | Нет | | Вывод: | A = Текущий активный банк в диапазоне адресов 0..7FFFH |
| Использование: AF | |

Эта функция возвращает банк памяти в настоящее время в контексте в диапазоне адресов 0..7FFFH. Она может быть использован в роли "где я" в прикладных программах для отслеживания обращений к памяти.

## 5.2 Структуры данных BIOS

### 5.2.1 Область конфигурации

Большая часть возможностей адаптации B/P BIOS для ваших конкретных функциональных требований обусловлена стандартизированным расположением многих дискретных элементов данных и возможностью легко найти и изменить их, независимо от используемой конкретной аппаратной платформы. Функция 30 BIOS (возвращение адреса BIOS), в регистровой паре DE сообщает начальный адрес области конфигурации. В этом разделе мы рассмотрим каждый из указанных элементов, их функции, и какие части данных должны жестко контролироваться, чтобы обеспечить функционирование предоставленных утилит, а также гарантировать мобильность других программ.

**CONFIG-6 - BIOS ID. Символьная строка, 6 байтов**

Это символьная строка должна начинаться с трех символов ASCII в верхнем регистре “B/P”, за которыми следуют три символа идентификации версии. По состоянию на март 1997 года были определены идентификаторы для следующих систем:

"B/P-YS" YASBEC

"B/P-AM" Ampro Little Board 100

"B/P-18" MicroMint SB-180

"B/P-CT" Compu/Time S100 Board Set

"B/P-TT" Teletek

"B/P-XL" Intelligent Computer Designs XL-M180

"B/P-DX" D-X Designs Pty Ltd P-112

**CONFIG+0 - IOBYTE. Байт**

Этот байт содержит начальные определения байта размещенного со смещением 3 в базовой странице (0003H) во время "холодного" старта и определяет, какое из четырех заданных символьных устройств ввода-вывода будет использоваться в качестве консоли, вспомогательного устройства и принтера. Значение по умолчанию может быть изменено с помощью утилиты BPCNFG чтобы отразить изменение конфигурации устройств, или повторной компиляцией BIOS.

Определения битов в этом байте следующие:

Бит 7 6 5 4 3 2 1 0

│ │ │ │ │ │ │ │

│ │ │ │ │ │ └─┴──── Консольное устройство

│ │ │ │ └─┴──────── Вспомогательное устройство ввода

│ │ └─┴──────────── Вспомогательное устройство вывода

└─┴──────────────── Устройство печати (принтер)

**CONFIG+1 – Системный диск. Байт**

Этот байт содержит диск, который будет доступен после "холодного" старта и, как предполагается, содержит файлы операционной системы. Это двоичное значение, где A = 0, B = 1...P = 15.

**CONFIG+2 - Флаги параметров BIOS. Байт**

Этот байт состоит из индивидуально обозначенных битов, которые показывают, какие опции активны в скомпилированной BIOS. Зарезервированные биты не должны быть определены без предварительного согласования с разработчиками системы, чтобы устранить конфликты с запланированными улучшениями. Байт в настоящее время определен следующим образом:

|  |  |  |
| --- | --- | --- |
| Бит 7 6 5 4 3 2 1 0  │ │ │ │ │ │ │ │ |  |  |
| │ │ │ │ │ │ │ └──── | 0 = BIOS без банков | 1 = BIOS с переключением банков |
| │ │ │ │ │ │ └────── | 0 = Банк в ОЗУ | 1 = Банк в ПЗУ |
| │ │ │ │ │ └──────── | 0 = Фиксированные DPB | 1 = Назначаемые DPB |
| │ │ │ │ └────────── | 0 = ALV/CSV в TPA | 1 = ALV/CSV в банке (ZSDOS2) |
| │ └─┴─┴──────────── | <Зарезервирован> |  |
| └────────────────── | 0 = не заблокирован | 1 = заблокирован, не может быть перезагружен |

Следующие пять байтов определяют карту памяти системы с переключением банков блоками по 32k. Для полного описания распределения банков пожалуйста, обратитесь к [Разделу 4](#_4_Установка_B/P_1). В системах, без переключения банков все, кроме банка виртуального диска должны все быть установлены в 0. Если память не доступна для повторного назначения в качестве виртуального диска, этот байт также должен быть установлен в 0.

**CONFIG+3 - Пользовательский банк. Байт**

Этот байт содержит номер банка, зарезервированный для приложений пользователя.

**CONFIG+4 - Банк TPA. Байт**

Этот байт отражает номер банка, зарезервированный для области транзитных программ в диапазоне адресов 0..7FFFH. Следующий порядковый номер банка обычно - *общий банк*, который всегда остается в контексте в диапазоне адресов 8000..FFFFH и содержит операционную систему, BIOS и таблицы Z-System.

**CONFIG+5 - Системный банк. Байт**

Этот байт отражает номер банка, содержащий любой исполняемый код и данные, определенные для системного банка.

**CONFIG+6 – Банк виртуального диска. Байт**

Этот байт отражает начальный номер банка, доступный для использования в качестве виртуального диска. Предполагается, что вся ОЗУ от этого банка до максимального номера банка непрерывна и доступна в качестве виртуального диска.

**CONFIG+7 - Максимальный номера банка. Байт**

Этот байт отражает номер последнего доступного банка ОЗУ в системе. Во многих системах ему могут быть присвоены различные числа в зависимости от количества микросхем памяти, установленных в системе.

**CONFIG+8 - Начальная страница обшей памяти. Байт**

Этот байт отражает начальную страницу из общей области в системах, которые не полностью соответствуют архитектуре B/P BIOS с банками памяти 32k, но могут быть сделаны до некоторой степени совместимыми. Этот байт должен быть по крайней мере 80H, но может быть выше, если это необходимо.

**CONFIG+9 – Размер DPB. Байт**

Этот байт содержит длину *блока параметров диска* (DPB) определенную в BIOS. Поскольку требуется больше информации, чем 15 байтов назначенных Digital Research в CP/M 2.2, используется расширенный формат. Все повторные назначения данных параметров диска должны использовать этот байт, чтобы определить размер записей.

**CONFIG+10 – Количество DPB в общем ОЗУ. Байт**

**CONFIG+11 - Количество DPB в системном банке. Байт**

Эти два байта указывают полный комплект форматов гибких дисков, доступных в BIOS. В большинстве случаев, один из этих двух байтов будет содержать нулевое значение для всех резидентных входов блоков параметров диска либо в общей области, или в системном банке. Однако, доступна возможность, с помощью этих двух байтов разделить определения в пользовательских версиях, без поддержки предоставляемой системными утилитами.

**CONFIG+12 – Указатель первого DPB в общем ОЗУ. Слово**

**CONFIG+14 - Указатель первого DPB в системном банке. Слово**

Эти два слова указывают на первый DPB в последовательном списке в соответствующих сегментах памяти для блоков параметров диска, определенных в предыдущих байтах. В большинстве случаев одно из этих двух слов будет иметь нулевой указатель (0000H) соответствующий отсутствию данных, как описано выше в байтах количества.

**CONFIG+16 - Начальная команда запуска. Строка**

Эта строка содержит первую команду, которая будет инициироваться при холодном старте. Она загружается в *буфер нескольких команд*, определенный в дескрипторе окружения (см [5.2.4](#_5.2.4_Дескриптор_окружения)) и вызывает файл с указанным именем с расширением “COM”. Строка может иметь до восьми символов и должна оканчиваться нулем (конец строки с бинарным 0). Определение строки следующее:

Байт - Число символов (0…8)

Строка - 8 байт ASCII символов (Обычно в верхнем регистре)

Байт - Символ окончания строки (бинарный 0)

**CONFIG+26 - Указатель дескриптора окружения. Слово**

Это слово указывает на первый байт расширенного окружения Z34, которое должно начинаться на границе страницы (xx00H). Смотрите Раздел 5.3.1 для полного описания дескриптора окружения и уникальных особенностей B/P BIOS.

**CONFIG+28 – Флаг пользовательского банка/номер банка. Байт**

Этот байт может использоваться в качестве флага для обозначения определяется ли банк пользователя. Банк 0 не может быть использован в качестве пользовательского банка по указу авторов системы. Таким образом, если этот байт содержит двоичный 0, ни один банк пользователя не доступен.

**CONFIG+29 – Указатель начала банков области пользователя. Слово**

Это слово содержит адрес первого свободного байта в переключаемой области пользователя, если она существует. Подпрограммы, загруженные в пользовательский банк, должны содержать стандартную структуру заголовка RSX, чтобы соединить последовательные программы и обеспечить примитивную функцию управления памятью.

**CONFIG+31 - Тактовая частота процессора в мегагерцах. Байт**

Этот байт должен содержать скорость процессора, округленную до ближайшего целого мегагерца. Он может использоваться в циклах синхронизации в прикладных и служебных программах для адаптации к тактовой частоте компьютера и обеспечения приблизительного времени. Этот байт отражается в дескрипторе окружающей среды (см. 5.2.4), а также для программ, которые “знают ” Z-System.

**CONFIG+32 - Дополнительно требуемые состояния ожидания. Байт**

Этот байт использует полубайты, чтобы отразить число состояний ожидания, необходимых для доступа к памяти и портам ввода-вывода, если эти функции можно настроить в программном обеспечении. Процессор Z80/Z180 при доступе к портам ввода-вывода вставляет одно состояние ожидания. Этот байт не учитывает этот факт и отражает состояния ожидания *в дополнение к* любым встроенным в аппаратные средства. Для более старых процессоров, таких как Z80, эти байты обычно не имеют никакого эффекта, так как дополнительные состояния ожидания должны быть добавлены с помощью аппаратных средств.

**CONFIG+33 - Значение таймера перезагрузки. Слово**

Во многих системах, прерывания или значения таймера устанавливаются программно-конфигурируемыми таймерами обратного отсчета. 16-разрядное значение в этом месте зарезервировано для задания значения таймера и может быть использовано для "тонкой настройки", чтобы позволить системе поддерживать корректное время в присутствии тактовых частот, которые могут отклониться от прецизионных частот, необходимых для точных часов.

**CONFIG+35 - Физические параметры гибкого диска. Таблица**

Эта таблица состоит из четырех записей из 5 байт, которые содержат информацию о максимум четырех физических дисках. Каждая запись определена следующим образом:

**Байт 0** - Байт обеспечивает начальный адрес XDPH[[1]](#footnote-1). Отображение битов:

|  |  |  |
| --- | --- | --- |
| Бит 7 6 5 4 3 2 1 0  │ │ │ │ │ │ │ │ |  |  |
| │ │ │ │ │ └─┴─┴──── | Размер диска 000=жесткий диск, 001=8", 010=5.25", 011=3.5" | |
| │ │ │ │ └────────── | 0 = Односторонний | 1 = Двухсторонний |
| │ │ │ └──────────── | <Зарезервирован> |  |
| │ │ └────────────── | 0 = Мотор всегда включен | 1 = Необход. управление мотором |
| │ └──────────────── | 0 = 300 об/мин | 1 = 360 об/мин (8" и HD) |
| └────────────────── | <Зарезервирован> |  |

**Байт** 1 - Время шага в миллисекундах.

**Байт** 2 - Время отвода головки после операции чтения/записи в миллисекундах.

**Байт** 3 - Время между подводом головки к поверхности и началом чтения/записи в миллисекундах.

**Байт** 4 - Число дорожек (цилиндров) на диске.

Биты обозначенные в байте 0, как зарезервированные должны быть равными 0, так как этот байт содержит начальные значения, хранящиеся в XDPH, во время использования указанных накопителей. Для контроллеров, которые не используют имеющуюся информацию (например, контроллерам Western Digital не нужен 3 байт) эти значения могут быть установлены в любое произвольное значение, но должны оставаться в структуре для предотвращения изменения последующего адреса.

**CONFIG+55 - Время включения мотора в 0,1 секунды. Байт**

Это время может использоваться в некоторых типах контроллеров гибких дисков, чтобы оставить двигатели дисковода, вращающимися в течение указанного времени после последнего доступа, чтобы избежать задержек связанных с раскруткой шпинделя до требуемой скорости. Некоторые контроллеры, особенно серия Western Digital 17xx и 19xx, не поддерживают эту функцию. В этом случае байт может быть установлен в любое произвольное значение, но обязательно должен присутствовать.

**CONFIG+56 - Время раскрутки мотора в 0,1 секунды. Байт**

Это время является задержкой, которая используется BIOS перед попыткой доступа к гибкому диску, когда он предполагает, что двигатель находится в остановленном состоянии. Использование этой задержки сводит к минимуму вероятность повреждения данных при записи на диск, который вращается с неправильной скоростью.

**CONFIG+57 - Максимальное количество повторных попыток. Байт**

Этот байт указывает количество попыток, которые будут сделаны при доступе к гибкому диску перед возвратом кода ошибки. В некоторых случаях, таких как диагностические программы, желательно установить это значение равным 1, чтобы выявить программные ошибки, или те, которые успешны не с первой, а на последующих попытках. Рекомендуемое значение 3 или 4 основано на нашем опыте. Большие значения могут привести к чрезмерно длительным задержкам при обнаружении ошибок.

**CONFIG+58 - Указатель на таблицу векторов прерываний. Слово**

Это слово содержит адрес начала таблицы векторов прерываний, которая, если используется, содержит указатели на сервисные подпрограммы. Точное определение таблицы не стандартизировано и может значительно отличаться в разных системах. Этот указатель предназначен только для обеспечения легкого и стандартизированного метода размещения таблицы для переопределения сервисов или характеристик системных функций.

**CONFIG+60 – Тип контроллера SCSI. Байт**

Для поддержки широкого спектра различных контроллеров, включая более старые модели SASI, этот байт определяется как содержащий байт-код для конкретной используемой модели. В большинстве случаев, этот байт не оказывает какого-либо эффекта в BIOS, но может иметь значительное влияние на программы диагностики жесткого диска, или разработанные пользователем утилиты. Любые дополнения к этой таблице должны быть согласованы с авторами, чтобы обеспечить, что стандартные утилиты поддержки продолжают функционировать. Текущие определения:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 - | Owl |  | 5 - | Conner SCSI |
| 1 - | Adaptec ACB-4000A |  | 6 - | Quantum SCSI |
| 2 - | Xebec 1410A/Shugart 1610-3 (SASI) |  | 7 - | Maxtor SCSI |
| 3 - | Seagate SCSI |  | 8 - | Syquest SCSI |
| 4 - | Shugart 1610-4 (Минимальное подмножество SCSI) |  | 9 - | GIDE (IDE/ATA) |

**CONFIG+61 - Физические параметры жесткого диска. Таблица**

Эта таблица состоит из трех 9-байтовых записей, определяющих до трех физических жестких дисков. Хотя определение SCSI допускает больше устройств, три было сочтено достаточным для большинства систем. Если необходимы дополнительные диски, пожалуйста свяжитесь с авторами для уточнения способа подключения их, не нарушающего работу любой из стандартных утилит или интерфейсов. Каждая из трех записей определяется следующим образом:

**Байт** - Логической и физической адрес:

|  |  |  |
| --- | --- | --- |
| Бит 7 6 5 4 3 2 1 0 |  | |
| │ │ │ │ │ │ │ │ |  | |
| │ │ │ │ │ └─┴─┴────  │ │ │ │ │ | Номер физического устройства (000-110B, 111B зарезервировано для хоста) | |
| │ │ │ │ └────────── | <Зарезервирован> | |
| │ │ │ └──────────── | 0 = Диск отсутствует | 1 = Диск присутствует |
| └─┴─┴────────────── | Номер логического устройства (000-111B) для контроллеров, способных обрабатывать несколько дисков | |

Слово - Количество физических цилиндров на диске

Байт - Количество используемых физических головок на диске

Слово - Начальная дорожка для предварительной записи

Слово - Начальная дорожка для предварительной компенсации при записи

Байт - Время шага. Этот байт может содержать абсолютное значение в миллисекундах или код, основанный на определениях конкретного контроллера

Для многих из новых контроллеров, последние три пункта, возможно, не имеют никакого смысла, в этом случае им могут быть присвоены любые произвольные значение. Кроме того, в новых дисках, физические характеристики, такие как количество цилиндров и головок может быть скрыто в электронике диска с повторным предоставлением значений, предоставляемых контроллером с помощью различных команд SCSI. Аналогично трем последним записям, в этом случае, им могут быть присвоены любые произвольные значения.

**CONFIG+88 (Зарезервированные байты)**

Пять байт зарезервированы для будущего расширения.

**CONFIG+93 - Определения символьных устройств. Таблица**

Эта таблица состоит из четырех или больше 16 байтных (8 байтных в B/P версиях до 1.1) записей завершающихся нулевым (двоичный нуль) байтом. Каждая запись определяет имя и характеристики символьного устройства ввода-вывода в системе. Первые четыре из них непосредственно доступны для выбора через IOBYTE: консоль, вспомогательные устройства ввода-вывода и принтер. Можно определить другие записи и обменять их с первыми четырьмя, чтобы сделать доступными в системе.

Записи определены следующим образом:

Строка – Имя из четырех символов ASCII: COM1, PIO1, NULL, и др.

Байт - Возможные скорости передачи данных:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Бит 7 6 5 4 3 2 1 0 |  | | | |
| │ │ │ │ │ |
| │ │ │ │ └────────── | Текущая настройка скорости передачи данных | | | |
| └─┴─┴─┴──────────── | Максимальная скорость (бит в секунду): | | | |
|  | 0000 = None | 0001 = 134.5 | 0010 = 50 | 0011 = 75 |
| 0100 = 150 | 0101 = 300 | 0110 = 600 | 0111 = 1200 |
| 1000 = 2400 | 1001 = 4800 | 1010 = 9600 | 1011 = 19200 |
| 1100 = 38400 | 1101 = 76800 | 1110 = 115200 | 1111 = Фикс. |

Слово - Указатель на подпрограмму статуса ввода

*Примечание*: В версии В/Р BIOS до 1.1 последние четыре указателя расположены не в этом месте, но были доступны с помощью указателя возвращаемого функцией 30 BIOS.

### 5.2.2 Заголовок параметров диска

Заголовок параметров диска (DPH) является структурой логических данных, необходимой для каждого диска в дисковой операционной системе совместимой с CP/M. Она состоит из серии восьми указателей, которые содержат адреса других элементов, необходимых DOS, а также некоторое пространство временной памяти. Адрес DPH, связанный с заданным диском возвращается после успешного выбора с помощью функции 9 BIOS. Если при выборе возникают ошибки, или диск не существует, возвращается пустой указатель (0000H).

Для BIOS B/P было необходимо добавить еще четыре байта к каждому DPH, которые содержат дополнительную информацию о физических и логических параметрах, а также информацию о флагах. Эти дополнительные байты называются расширенным DPH или XDPH. Хотя расширение во многом напоминает, добавленное в CP/M 3, реализовано иначе. XDPH добавляется перед началом DPH и может быть доступен путем уменьшения возвращаемого адреса. По соглашению, для DPH в исходном коде B/P BIOS зарезервированы определенные последовательности меток для определенных типов устройств: DPH00-DPH49 используются для дисководов гибких дисков, DPH50-DPH89 для разделов жесткого диска и DPH90-DPH99 для виртуальных дисков.

Весь блок DPH/XDPH требуется для каждого логического диска в системе B/P BIOS. В то время как некоторые указатели, такие как указатель на буфер каталога (Directory), может быть общим для нескольких дисков для большинства систем, другие элементы будут указывать на уникальные области.

Для прикладных программ доступны следующие элементы DPH/XDPH, как индексация от адреса DPH:

**DPH-4 - Флаг блокировки формата. Байт**

Нулевое значение указывает, что формат диска не фиксирован, но может быть изменен. Если BIOS была скомпилирована с параметром Auto-select, BIOS просканирует много различных форматов, чтобы идентифицировать диск. Если значение 0FFH помещено в этот байт, он указывает, что формат фиксирован и не может быть изменен. Это обычно имеет место для виртуальных и жестких дисков, а также для посторонних форматов гибких дисков, которые были выбраны в режиме эмуляции. Если параметр Auto-select не был выбран во время компиляции BIOS, у всех дисководов гибких дисков в этой позиции байт также будет установлен 0FFH, показывая, что форматы не могут быть изменены.

**DPH-3 - Тип дисковода. Байт**

Этот байт - с побитовым отображением и содержит флаги, указывающие на многие параметры диска. Для дисководов гибких дисков этот байт содержит копию первого байта в *таблице физического диска* (См. [5.2.1](#_5.2.1_Область_конфигурации), CONFIG+35) с двумя зарезервированными байтами устанавливаемыми во время процесса выбора диска. Байт определен следующим образом:

|  |  |  |
| --- | --- | --- |
| Бит 7 6 5 4 3 2 1 0  │ │ │ │ │ │ │ │ |  |  |
| │ │ │ │ │ └─┴─┴──── | Размер диска 000=жесткий диск, 001=8", 010=5.25", 011=3.5" | |
| │ │ │ │ └────────── | 0 = Односторонний | 1 = Двухсторонний |
| │ │ │ └──────────── | 0 = Диск с одним шагом | 1 = Диск с двойным шагом |
| │ │ └────────────── | 0 = Мотор всегда включен | 1 = Необход. управление мотором |
| │ └──────────────── | 0 = 300 об/мин | 1 = 360 об/мин (8" и HD) |
| └────────────────── | 0 = Двойная плотность | 1 = Одинарная плотность |

Для разделов жесткого диска и виртуального диска, этот байт не используется и все значение содержат нули, указывая тип жесткого диска.

**DPH-2 - Идентификационный номер драйвера. Байт**

В базовой конфигурации B/P BIOS используются три типа драйверов. Нулевое значение указывает на несуществующий драйвер, другие значения используются для прямого обращения к диску с кодом, соответствующим типу устройства. Основные определенные типы драйверов существуют для гибкого диска (1), жесткого диска через интерфейс SCSI (2), и виртуального диска (3). Если вы хотите расширить эту таблицу для включения адаптированных драйверов, проконсультируйтесь с авторами, чтобы избежать возможных конфликтов с планируемыми расширениями.

**DPH-1 - Номер физического диска/устройства. Байт**

Этот байт содержит номер физического диска или устройства, размещающего логический диск. Для дисководов гибких дисков он обычно будет в диапазоне от 0 до 3 для четырех дисков. У жестких дисков может быть несколько DPH, совместно использующих тот же номер физического диска, в то время как это поле игнорируется для единственного виртуального диска, поддерживаемого в дистрибутивной версии B/P BIOS.

*Примечание*: байт номера физического диска для жестких дисков состоит из двух полей для облегчения обработки устройств SCSI. До семи устройств (устройство 111B зарезервировано для хост-компьютера) могут быть определены каждый из которых может иметь до 8 логических устройств. Байт настраивается следующим образом:

Физический номер диска

Бит 7 6 5 4 3 2 1 0

│ │ │ │ │ │ │ │

│ │ │ │ │ └─┴─┴──── Физический диск (000-110B, 111B - зарезервировано для хоста)

│ │ │ │ └────────── <Зарезервирован>

│ │ │ └──────────── 0 = Устройство недоступно, 1 = Устройство активно

└─┴─┴────────────── Номер логического устройства (000-111B)

**DPH+0/1 – Указатель таблицы смещений (XLT). Слово**

Это слово содержит указатель на таблицу смещения. Она редко используется для жестких и виртуальных дисков, но требуется для дисководов гибких дисков. Если BIOS была скомпилирована, используя параметр CALCSK (вычислить смещение), адрес указывает на байт, абсолютное значение которого является численным значением смещения (обычно в диапазоне от 1 до 6) используемого для доступов к диску. Этот термин часто заменяется чередованием и синонимичен с этой целью. Если значение байта является отрицательным, это означает, что секторы на диске записаны в смещенном виде и операции чтения и записи должны быть последовательными. Если значение положительное, то вызывается алгоритм для вычисления номера физического сектора на основе требуемого логического сектора и фактора смещения. Для систем, собранных без CALCSK (вычислить смещение), это слово указывает на таблицу до 26 байт, которые должны быть проиндексированы требуемым физическим номером сектора (0..Максимальный номер сектора) для получения номера соответствующего сектора диска.

**DPH+2 – Рабочие слова DOS. 3 слова**

Эти три слова DOS использует во время работы. Никакие фиксированные значения не присвоены, не являются значениями для данных, хранивших там любое значения.

**DPH+8/9 - Указатель на буфер каталога (DIRBUF). Слово**

Это слово указывает на 128-байтовую область данных, которая используется для поиска в каталоге. Обычно эта область общая для всех DPH в системе и часто обновляется DOS в нормальных условиях эксплуатации.

**DPH+10/11 - Указатель на DPB. Слово**

Это слово указывает на другую структуру данных, которая детализирует многие логические параметры выбранного диска или раздела. Ее структура описана в [Разделе 5.2.3](#_5.2.3_Блок_параметров_2) ниже. Диски того же типа и логической конфигурации могут совместно использовать определения DPB, таким образом, велика вероятность найти указатели на DPB в различных структурах DPH, указывающих на ту же область.

**DPH+12/13 - Буфер контрольной суммы диска (CSV). Слово**

Это слово указывает на рабочую буферную зону ОЗУ для дисководов со съемными носителями, используемую для обнаружения изменений дисков. Обычно эта функция используется только для дисководов гибких дисков и отключена, содержа нулевое слово (0000H) для жестких и виртуальных дисков. Для дисководов гибких дисков необходима область ОЗУ с одним байтом для каждых четырех записей каталога (128-байтовый сектор) (См. [5.2.3](#_5.2.3_Блок_параметров_1), DPH+11/12). Эта рабочая область не может быть совместно использована несколькими дисками.

Следует отметить, что в системах B/P BIOS с поддержкой переключения банков с ZSDOS2, буфер контрольной суммы помещается в системный банк и непосредственно не доступен прикладным программам.

**DPH+14/15 – Буфер вектора распределения (ALV). Слово**

Это слово указывает на битовый буфер, содержащий один бит для каждого блока распределения на обусловленном диске (См. [5.2.3](#_5.2.3_Блок_параметров), DPB+5/6). Бит "1" в этом буфере означает, что соответствующий блок данных на устройстве уже выделен файлу, в то время как "0" означает, что блок свободен. Этот буфер уникален для каждого логического диска и не может быть совместно использован несколькими дисками.

Следует отметить, что в системах B/P BIOS с поддержкой переключения банков с ZSDOS2, буфер ALV помещается в системный банк и непосредственно не доступен прикладным программам. Поскольку доступ к буферу ALV часто необходим для вычисления свободного места на дисках, ZSDOS2 содержит дополнительную функцию для возвращения свободного места на диске. Использование этого вызова позволяет приложениям получить доступ к информации без прямого доступа к структуре данных.

### 5.2.3 Блок параметров диска

Блок параметров диска (DPB) - структура данных, определенная Digital Research для CP/M, определяет логическую конфигурацию хранения информации на внешних накопителях. Она была расширена в B/P BIOS для обеспечения дополнительной информации, чтобы обеспечить повышенную гибкость и возможности. Расширение называется расширенным DPB или XDPB, и добавляется перед началом фактической структуры DPB. Адрес DPB может быть получен из указателя DPH, возвращенного BIOS или DOS после выбора диска (См. [5.2.2](#_5.2.2_Заголовок_параметров_1) выше). Все DPB находятся в обшей области памяти и доступны прикладным программам как в системах с переключением так и без переключения банков. По соглашению DPB помечаются таким же образом как DPH: DPB00-DPB49 используются для дисководов гибких дисков, DPB50-DPB89 для разделов жесткого диска и DPB90-DPB99 для виртуальных дисков.

Расположение блока параметров диска, как индексировано от доступного указателя DPB следующее:

**DPB-16 – Строка ASCII обозначения. 10 байт**

Эта строка служит обозначений, которые могут быть напечатаны прикладными программами, такими как наша BPFORMAT. Эта строка может содержать до 10 смешанных буквенно-цифровых символов ASCII, но последний допустимый символ должен иметь установленный в “1” самый старший бит (бит 7).

**DPB-6 – Тип формата байт 0. Байт**

Этот байт содержит некоторые сведения о формате диска и логическую последовательность информации на физическом носителе. Биты в байте имеют следующие значения:

|  |  |  |
| --- | --- | --- |
| Бит 7 6 5 4 3 2 1 0  │ │ │ │ │ │ │ │ |  |  |
| │ │ │ │ │ └─┴─┴──── | Размер диска: 000=жесткий диск, 001=8", 010=5.25", 011=3.5" | |
| │ │ └─┴─┴────────── | **Тип дорожки:** |  |
| │ │ | 000 = Односторонний | 001 = Зарезервировано |
| │ │ | 010 = Sel by Sec, Cont | 011 = Sel by Sec, Sec # Same |
| │ │ | 100 = S0 All, S1 All | 101 = S0 All,S1 All Reverse |
| │ │ | 110 = Sel by Trk LSB | 111 = Зарезервировано |
| │ └──────────────── | **Дорожка 0 сторона 0:** |  |
| │ | 0 = Двойной плотности | 1 = Одинарной плотности |
| └────────────────── | **Дорожки с данными:** |  |
|  | 0 = Двойной плотности | 1 = Одинарной плотности |

Для жестких и виртуальных дисков, этот байт содержит все нулевые биты, для обозначения фиксированного носителя и формата.

**DPB-5 – Тип формата байт 1. Байт**

Этот байт содержит дополнительные сведения о формате. Биты имеют следующие значения:

|  |  |  |
| --- | --- | --- |
| Бит 7 6 5 4 3 2 1 0  │ │ │ │ │ │ │ │ |  |  |
| │ │ │ │ │ └─┴─┴──── | Размер сектора: 000=128, 001=256, 010=512, 011=1024 | |
| │ │ └─┴─┴────────── | Размер размещения: 000=1K, 001=2K, 010=4K, 011=8K, 100=16K | |
| │ │ | *Примечание*: Это должно соответствовать определению в DPH | |
| │ └──────────────── | <Зарезервирован> |  |
| └────────────────── | 0 = Нормальная скорость (300 об/мин) | |
|  | 1 = Дискета 8" или HD (360 об/мин), или жесткий диск | |

Для жестких дисков, дистрибутив версии B/P BIOS и вспомогательные утилиты предполагают, что размер сектора всегда равен 512 байт. Остальные биты должны быть установлены согласно описания.

**DPB-4 - Коэффициент смещения. Байт**

Этот байт - двоичное значение со знаком, указывающее коэффициент смещения, который будет использоваться во время форматирования, чтения и записи. Он обычно используется только с дисководами гибких дисков и обычно устанавливается равным -1 (0FFH) для жестких и виртуальных дисков, чтобы указать, что чтение и запись должны производиться без смещения. Если параметр CALCSK (вычислить смещение) действовала во время компиляции BIOS, указатель смешения в DPH (BPH+0/1) указывает на этот байт. Если используется таблица смещения, этот байт не имеет никакого эффекта и должен иметь значение 80H.

**DPB-3 - Начальный номер сектора. Байт**

Этот байт содержит номер первого физического сектора на каждой дорожке. Так как большинство дисковых операционных систем использует последовательную нумерацию секторов начинающуюся с 0, это значение обеспечивает начальное смещение для коррекции логических номеров секторов в физические.

**DPB-2 – Количество физических секторов на дорожке. Байт**

Этот байт содержит число физических (а не логических) секторов на каждой дорожке. Например, CP/M вычисляет сектора на основе 128-байтовых распределений, которые используются на 8" гибких дисках одинарной плотности. Один из популярных пятидюймовых форматов использует пять 1K физических секторов, которые приравниваются к 40 логическим секторам CP/M. Этот байт содержит число 5 в случае 1K физических секторов.

**DPB-1 - Количество физических дорожек на стороне. Байт**

Этот байт содержит число физических дорожек на стороне, также называемое числом цилиндров. Оно отражает диск, в противоположность возможностям привода и используется, чтобы установить требования для двойного продвижения дисководов гибких дисков. В случае диска с 40 дорожками, помещенного в дисковод с 80 дорожками, этот байт содержал бы 40, в то время как параметр диска в разделе конфигурации содержит 80 в качестве числа дорожек на диске. Этот байт не имеет никакого значения для разделов жесткого диска или виртуальных дисков и должен быть обнулен, несмотря на то, что приемлемо любое произвольное значение.

**DPB+0/1 - Логических секторов на дорожке (SPT). Слово**

Это значение-число логических 128-байтных секторов на каждой дорожке диска. Это эквивалентно количеству физических секторов раз Размер физического сектора MOD 128.

**DPB+2 - Коэффициент смещения блока (BSH). Байт**

**DPB+3 - Маска блока данных (BLM). Байт**

**DPB+4 - Маска размера блока (EXM). Байт**

Эти три байта содержат значения, используемые операционной системой для вычисления дорожек и секторов для доступа к логическим дискам. Их значения подробно описаны в различных руководствам по программированию CP/M и ZSDOS и не должно изменяться без знания их последствий.

**DPB+5/6 – Размер диска (DSM). Слово**

Это слово содержит номер последнего выделенного блока на диске. Он совпадает с числом выделенных блоков - 1. Например, если используются 4k блоки распределения и определен дисковод 10 Мегабайт, это слово будет содержать 10000000/4000 - 1 или 2499.

**DPB+7/8 - Максимальная запись каталога (DRM). Слово**

Это слово содержит номер последней записи в каталоге и совпадает с числом записей - 1. Например, если необходимо 1024 входов в каталоге, это слово должно содержать 1024 - 1 = 1023.

**DPB+9/10 - Распределение 0 и 1 (AL0, AL1). 2 байта**

Эти два байта содержат начальные распределения, хранящиеся в первых двух байтах буфера ALV (см. [5.2.2](#_5.2.2_Заголовок_параметров), DPH+14/15) во время первоначального выбора диска. В основном он используется, чтобы указать, что секторы каталога уже выделены и недоступны для хранения данных. Они являются значениями с побитовым отображением и используются в формате старший байт младший байт, в отличие от нормально используемого формата младший байт старший байт для хранения слова в процессорах типа Z80. Биты выделяются начиная со старшего бита первого байта до младшего байта, затем со старшего бита до младшего бита второго байта по одному биту на выделенный блок или часть их используется для каталога. Для расчета битов сначала вычисляется количество записей в блоке распределения, затем требуемое количество записей делится на это число. Любой остаток требует дополнительного выделения бит.

Например, если используются 4K блоки распределения, каждый блок способен к 4096/32 байта на запись = 128 Записей каталога. Если требуется 512 записей, то необходимы 512/128 = 4 блока выделения, который определяет, что байт распределения 0 будет 11110000B (0F0H), и байт распределения 1 будет 00000000B.

**DPB+11/12 - Проверка размера (CKS). Слово**

Это слово используется только для съемных носителей (обычно только дисководы гибких дисков) и указывает количество секторов для которых вычисляется контрольная сумма для обнаружения изменения дисков. Ему должно быть присвоено значение 0000H для жестких и виртуальных дисков чтобы избежать потери времени на перерегистрацию после каждого теплого старта.

**DPB+13/14 - Смещение дорожки (OFF). Слово**

Это слово указывает число логических дорожек, которые пропускаются, прежде чем начнется существующий DPB. Оно обычно используется, чтобы зарезервировать дорожки начальной загрузки (обычно 1 - 3) или разделить большие диски на меньшие логические единицы, пропуская дорожки, используемые для других определений диска.

## 5.3 Системные структуры данных

### 5.3.1 Дескриптор окружения

Дескриптор окружения (Environment), называемый просто ENV, является основой того, что теперь известно как Z-System. Последние дополнения к системе выполненные Джо Райт {Joe Wright} и Джей Сейджем {Jay Sage} заменили в ENV некоторые относительно бессмысленные элементы с системно-зависимой информацией, такой как расположение компонентов операционной системы. Вследствие этого, ENV не просто функция замены командного процессора ZCPR 3.4, но и является ресурсом операционной системы, который позволяет другим программам, таким как ZCPR 3.4 получать доступ к этой информации.

B/P BIOS требует присутствия ENV и использует несколько элементов информации, содержащихся в нем. ZSDOS2 с переключением банков и замена командного процессора Z40 используют еще больше функций ENV. Несколько остающихся байтов были переопределены для поддержки систем основанных на B/P BIOS. Для обозначения определений элементов данных B/P BIOS, был зарезервирован новый тип, 90H. Используя этот байт "Типа", пользовательские программы могут получить доступ и воспользоваться преимуществами новых определений и функций.

Шаблон для дескриптора окружения, используемый B/P BIOS, который принимает значения из файла Z3BASE.LIB, включённого в дистрибутивный диск следующий:

Предостережение

При переходе от более старого операционного окружения, которое использует загружаемые пользователями файлы окружения, такие как SYS.ENV, вы должны или удалить такие инструкции загрузки или изменить ENV файлы, чтобы отразить точные определения загруженной системе. Игнорирование этого факта может вызвать много нежелательных побочных эффектов!

ENV: JP 0 ; Предшествующий переход (адрес CBIOS если NZCOM)

DEFB 'Z3ENV' ; Идентификатор окружения

DEFB 90H ; Тип Env (>=90H означает расширенное B/P ENV

; с/областью пользователя)

DEFW EXPATH ; Внешний путь (PATH)

DEFB EXPATHS ;

DEFW RCP ; Пакет резидентных команд (RCP)

DEFB RCPS ;

DEFW IOP ; Пакет ввода/вывода (IOP)

DEFB IOPS ;

DEFW FCP ; Пакет команд потока (FCP)

DEFB FCPS ;

DEFW Z3NDIR ; Именованные каталоги (NDR)

DEFB Z3NDIRS ;

DEFW Z3CL ; Командная строка (CL)

DEFB Z3CLS ;

DEFW Z3ENV ; Окружение (ENV) - фактический начальный адрес

DEFB Z3ENVS ;

DEFW SHSTK ; Стек интерпретатора (SH)

DEFB SHSTKS ;

DEFB SHSIZE ;

DEFW Z3MSG ; Буфер сообщений (MSG)

DEFW EXTFCB ; Внешний fcb (FCB)

DEFW EXTSTK ; Внешний стек (STK)

DEFB 0 ; Тихий флаг (1=тихо, 0=не тихий)

DEFW Z3WHL ; Байт колеса (WHL)

DEFB 9 ; Частота процессора (MHZ)

DEFB 'P'-'@' ; Буква максимального диска

DEFB 31 ; Максимальный номер пользователя

DEFB 1 ; 1 = принять DU:, 0 = не принять

DEFB 0 ; Выбор CRT ()

DEFB 0 ; Выбор принтера ()

DEFB 80 ; CRT 0: Ширина

DEFB 24 ; Число строк

DEFB 22 ; Число текстовых строк

; В расширенном ENV CRT 1 заменен системной информацией

;; DEFB 132 ; .CRT 1: Ширина

;; DEFB 24 ; Число строк

;; DEFB 22 ; Число текстовых строк

; PONMLKJIHGFEDCBA

IF RAMDSK

DEFW 0001000001111111B ; Виртуальный диск обозначается M:

ELSE

DEFW 0000000001111111B ; Вектор допустимых дисков

ENDIF

DEFB 0 ; (Зарезервировано)

DEFB 80 ; PRT 0: Ширина

DEFB 66 ; Число строк

DEFB 58 ; Число текстовых строк

DEFB 1 ; Флаг FF (1=возможен прогон страницы)

;========================================================================

; Определения Prt1 используются для векторов резидентного пространства пользователя

;; DEFB 96 ; PRT 1: Ширина

;; DEFB 66 ; Число строк

;; DEFB 58 ; Число текстовых строк

;; DEFB 1 ; Флаг FF (1=возможен прогон страницы)

DEFB USPCS ; Свободное пространство пользователя (записей)

USRSP: DEFW USPC ; Начальный адрес резидентного пространства

; пользователя (xx00H/xx80H)

DEFB USPCS ; Размер резидентного пространства пользователя

; в 128-байтовых записях

;========================================================================

; В Расширенном ENV Принтеров 2 и 3 не стало, заменены системной информацией

;; DEFB 132 ; PRT 2: Ширина

;; DEFB 66 ; Число строк

;; DEFB 58 ; Число текстовых строк

;; DEFB 1 ; Флаг FF (1=возможен прогон страницы)

;;

;; DEFB 132 ; PRT 3: Ширина

;; DEFB 88 ; Число строк

;; DEFB 82 ; Число текстовых строк

;; DEFB 1 ; Флаг FF (1=возможен прогон страницы)

DEFW CPR ; Начальный адрес CCP

DEFB [DOS-CPR]/128 ; Размер CCP в 128-байтовых записях

DEFW DOS ; Начальный адрес BDOS (xx00H или xx80H)

DEFB [BIOSJT-DOS]/128 ; Размер буфера BDOS в 128-байтовых записях

DEFW BIOSJT ; Начальный адрес BIOS (NZBIO если выполняется NZCOM)

DEFB 'SH ' ; Имя файла переменной Shell

DEFB 'VAR' ; Расширение файла переменной Shell

DEFB ' ' ; Файл 1

DEFB ' ' ;

DEFB ' ' ; Файл 2

DEFB ' ' ;

DEFB ' ' ; Файл 3

DEFB ' ' ;

DEFB ' ' ; Файл 4

DEFB ' ' ;

DEFB 0 ; Общедоступная область диска (ZRDOS +)

DEFB 0 ; Общедоступная область пользователя (ZRDOS +)

; ENV 128 байт длиной

### 5.3.2 Возможности терминала

В дополнение к основному дескриптору окружения, описанному выше, шаблон структуры записей возможностей терминала (TCAP, также известный как TERMCAP), добавляется в зарезервированное пространство для, и определения возможностей компьютерного терминала по умолчанию. TERMCAP по умолчанию полностью совместим с подпрограммами VLIB, используемыми в сообществе Z-System после VLIB4D. Шаблон структуры записи:

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; Эти данные TermCap для новой Z-System соответствуют спецификации VLIB4D и

; более полно описывает терминал и его возможности. Измените поля со зна-

; чениями на параметры вашего терминала, или используйте его в качестве шаб-

; лона для внешнего определения, загружаемого из файла начального запуска.

TCAP: DEFB ' ' ; Имя терминала (13 байтов, заканчивается

; пробелом)

B13: DEFB GOELD-ENV2 ; Смещение GOELD в графическом разделе

B14: DEFB 10000000B ; Бит 7 = Расширенный TCAP, остаток

; не определен

; B15 b0 Выделение 0 = Пониженная яркость, 1 = Негативное изображение

; B15 b1 Задержка при 0 = Нет, 1 = Задержка 10 секунд

; Включении питания

; B15 b2 Нет переноса 0 = Перенос строки, 1 = Нет переноса, если

; символ записан в

; последнюю позицию

; экрана

; B15 b3 Нет прокрутки 0 = Прокрутка, 1 = Нет прокрутки, если

; символ записан в

; последнюю строку

; экрана

; B15 b4 ANSI 0 = ASCII, 1 = ANSI

B15: DEFB 00000000B ; Негативное изображение, перенос строки,

; прокрутка, ASCII

;

; Дополнительные байты команды перемещения курсора

;

DEFB 'E'-'@' ; Курсор вверх

DEFB 'X'-'@' ; Курсор вниз

DEFB 'D'-'@' ; Курсор вправо

DEFB 'S'-'@' ; Курсор влево

DEFB 0 ; CL Задержка при очистке экрана (0 до 255)

DEFB 0 ; CM Задержка при перемещение курсора

DEFB 0 ; CE Задержка при очистке до конца строки

; Строки начинаются здесь

DEFB 0 ; (CL) В начало экрана и очистить экран

DEFB 0 ; (CM) Перемещение курсора

DEFB 0 ; (CE) Очистка до конца строки

DEFB 0 ; (SO) Включить негативное изображение

DEFB 0 ; (SE) Выключить негативное изображение

DEFB 0 ; (TO) Инициализация терминала

DEFB 0 ; (TE) Восстановление терминал в нормальный

; режим

; Расширения стандартного Z3TCAP

DEFB 0 ; (LD) Удалить строку

DEFB 0 ; (LI) Вставить строку

DEFB 0 ; (CD) Очистка от курсора до конца экрана

; Параметры настройки атрибутов

DEFB 0 ; Установить атрибуты

DEFB 0 ; Строка атрибутов

; Чтение элементов с экрана

DEFB 0 ; Report Cursor Pos'n (ESC Y Pn Pn)

DEFB 0 ; Read Line Under Cursor

GOELD: DEFB 0 ; Включить/Выключить задержку

; Графические строки смещены от значение Задержки

DEFB 0 ; Включить графический режим

DEFB 0 ; Выключить графический режим

DEFB 0 ; Выключить курсор

DEFB 0 ; Включить курсор

; Графические символы

DEFB '\*' ; Верхний левый угол [\*]

DEFB '\*' ; Верхний правый угол [\*]

DEFB '\*' ; Нижний левый угол [\*]

DEFB '\*' ; Нижний правый угол [\*]

DEFB '-' ; Горизонтальная линия [-]

DEFB '|' ; Вертикальная линия [|]

DEFB '\*' ; Полный блок (hashed block) [\*]

DEFB '#' ; Hashed Block (big X) [#]

DEFB '+' ; Верхнее пересечение (перевернутое "T") [+]

DEFB '+' ; Нижнее пересечение ("T") [+]

DEFB '+' ; Среднее пересечение (пересечение линий) [+]

DEFB '+' ; Правое пересечение ("Т" повернутое влево)[+]

DEFB '+' ; Левое пересечение ("Т" повернутое вправо)[+]

DEFB 0

DEFB 0

ENVEND:

# 6. Утилиты B/P BIOS

Мы разработали и адаптировали многие процедуры поддержки, чтобы помочь вам в использовании и адаптации установок B/P BIOS. Многие из них следуют общепринятым функциям, и их использование может быть легко воспринято. Другие являются адаптацией процедур, которые мы разработали для поддержки нашей предыдущей операционной системы ZSDOS, или взяты из подпрограмм доступных в общественном достоянии. Остальные были написаны специально для поддержки данного продукта.

Каждая процедура поддержки, как правило, соответствует общепринятым стандартам утилит Z-System, включая встроенную справку и повторное выполнение с помощью команды GO.

Справка доступна, вводом имени необходимой процедуры за которым следуют два символа косой черты (т.е. LDSYS //). Дополнительная информация приводится в следующих разделах данного руководства.

Некоторые функции B/P BIOS поддерживаются программами зависимыми от версии. Они отличаются для разных систем и не должны быть смешаны, если вы поддерживает несколько различных типов компьютеров. Свяжитесь с нами, если вы должны создать такую индивидуальную версию одной из этих программ. Далее описаны подпрограммы, которые ограничены определенной системой или которые, как упомянуто, требуют адаптации.

## 6.1 BPBUILD - утилита создания образа системы

Цель BPBUILD состоит в том, чтобы создать загружаемый образ системы. Входные файлы, необходимые этой утилите включать вывод компиляции файла BPBIO-xx (в формате Microsoft REL), REL или ZRL образ операционной системы (ZSDOS без переключения банков или ZSDOS2 (ZS203.ZRL или ZS227G.ZRL с полной поддержкой переключения банков рекомендуются), и REL или ZRL образ командного процессора (ZCPR33.REL без переключения банков, Z41.ZRL для систем с полной поддержкой переключения банков рекомендуются).

BPBUILD способен встраивать различные типы компонентов операционной системы в исполняемое изображение. Среди протестированных систем настроенный BIOS, ZRDOS, CP/M 2.2, ZCPR2, ZCPR3x и другие. Одно из ограничений в любом сегменте возникает при использовании именованных COMMON для размещения различных частей системы в карте памяти процессора. Большинство ассемблеров и компоновщиков ограничены в числе различных именованных COMMON, которые могут поддерживаться. Компоновщик, включенный в BPBUILD, может обработать только следующие имена COMMON: \_ENV\_, \_MCL\_, \_MSG\_, \_FCB\_, \_SSTK\_, \_XSTK\_, \_BIOS\_, BANK2, B2RAM и RESVD.

### 6.1.1 Использование BPBUILD

BPBUILD работает с многоуровневыми меню и может быть вызван, чтобы создать замену для существующего файла с образом, или со значениями по умолчанию, чтобы создать полностью новый файл образа. Все файлы необходимые для создания требуемого образа: BIOS, DOS и CPR должны находиться на текущем диске области пользователя или быть доступными через общий атрибут при работе под ZSDOS. Синтаксис для BPBUILD следующий:

BPBUILD - Создать систему по умолчанию

BPBUILD fn[.ft] - Сделать/изменить в указанную систему

(По умолчанию расширение .IMG)

BPBUILD // - Вывод этого сообщения

Нужно отметить, что, если ZSDOS2 выбирается в качестве DOS, буферы битов распределения для жестких дисков будут расположены в системном банке.

### 6.1.2 Информация об экранах меню

После запуска BPBUILD вам будет представлен экран главного меню. Из этого экрана вы можете выбрать одну из трех основных категорий, чтобы адаптировать выходное изображение. В любой момент конфигурации нажав Ctrl-C или ESC можно выйти из меню и вернуться в командный процессор.

┌───────────────────────────────────────────────────────────────────────────┐

│ Main │

│ │

│ │

│ 1 File Names │

│ │

│ 2 BIOS Configuration │

│ │

│ 3 Environment │

│ │

│ Selection : │

└───────────────────────────────────────────────────────────────────────────┘

#### 6.1.2.1 Экран 1 - Имена файлов

Выбор первого варианта в главном меню представит экран, перечисляющий текущие имена файлов для трех входных файлов, необходимых для построения образа и имя, которое будет применено к выходному файлу. Если BPBUILD был запущен с именем существующего файла с образом, файлы, используемые для построения указанного файла с образом будут иметь те имена, в противном случае будут использованы имена по умолчанию, предоставленные BPBUILD. Пример экрана для системы YASBEC без переключения банков может быть следующим:

┌───────────────────────────────────────────────────────────────────────────┐

│ Files (1.1) │

│ │

│ │

│ 1 Command Processor File : ZCPR33 .REL │

│ │

│ 2 Operating System File : ZSDOS .ZRL │

│ │

│ 3 B/P Bios Source File : B/P-YS .REL │

│ │

│ 4 B/P Executable Image : BPSYS .IMG │

│ │

│ Selection : │

└───────────────────────────────────────────────────────────────────────────┘

Выбор любого из этих четырех вариантов позволит вам изменять используемое имя файла, для требуемого сегмента. Для каждой записи существуют расширение файла по умолчанию, и для файлов командного процессора и операционной системы, по умолчанию используется значение .ZRL, если расширение не вводится. Значения по умолчанию для файла BIOS используется .REL, в то время как значение по умолчанию выходного файла с образом .IMG.

#### 6.1.2.2 Экран 2 - Конфигурация BIOS

Выбор конфигурации BIOS из главного меню (Выбор 2) представляет основной экран с размерами и расположением либо существующих сегментов системы (при создании из существующего файла IMG), или значения по умолчанию присваиваемые BPBUILD. Пример экрана может быть следующим:

┌───────────────────────────────────────────────────────────────────────────┐

│ Environment (2.1) │

│ │

│ COMMON (Bank 0) MEMORY BANK 2 MEMORY │

│ ---------------------- ------------------------ │

│ A Common BIOS - D400H E Banked BIOS - 0000H │

│ Size - 83 Size - 0 │

│ B Common BDOS - C600H F Banked BDOS - 0000H │

│ Size - 28 Size - 0 │

│ C Command Proc - BE00H G Command Proc - 0000H │

│ Size - 16 Size - 0 │

│ D User Space - E900H H User Space - 0000H │

│ Size - 6 Size - 0 │

│ │

| Selection : │

└───────────────────────────────────────────────────────────────────────────┘

Хотя на данном этапе предоставляется возможность задавать расположение и размеры для различных системных размеров, мы призываем вас не изменять значения для банка 2 памяти, если вы не очень знакомы с потенциальными последствиями и готовы рискнуть потенциально катастрофическими последствиями. Основная причина включения этого экрана состояла в том, чтобы предоставить возможность настройки базового расположения общей памяти и задавать размер резидентного пространства пользователя. Кроме определения размера общего пространства пользователя (начального адреса расположения по умолчанию в Z3BASE.LIB), остальные значения были включены в основном для тех немногих специализированных пользователей, которым требуются настройка системных расположений.

#### 6.1.2.3 Экран 3 - Настройка окружения

Так как дескриптор окружения является неотъемлемая часть операционной системы из-за определения параметров низкого уровня, таких как выделение памяти, этот экран предназначен для настройки карты памяти создаваемой системы подходящим способом. Например, системе с переключением банков с ZSDOS2, обычно не требуется резидентный командный процессор. Выбрав вариант “D” в этом экране можно указать, что для RCP пространство не выделяется, установив адрес его размещения и размер равными нулю. За счет этого освободившегося пространства пакет IO (Выбор “C”) может быть увеличен до F400H, распространяя его размер за 12 записей. Так как это два самых нижних сегмента в карте памяти, BPBUILD будет использовать его в качестве самого нижнего значения, используемого в окружении, и перемещать сегменты операционной системы (включая резидентное пространство пользователя) в памяти для увеличения на 2k байт области транзитных программ.

Важно отметить, что дескриптор окружения, определенный в этом экране, сохраняется в специальной области в созданном .IMG файле, и загружается в память с помощью LDSYS при активации. Любые изменения после его загрузки, например загружая другой файл ENV как часть сценария начального запуска (STARTUP) могут привести к неправильно функционирующей системе.

┌───────────────────────────────────────────────────────────────────────────┐

│ Environment (3.1) │

│ │

│ A - Environment - FE00H F - Named Dirs - FC00H │

│ Size (# recs)- 2 # of Entries - 14 │

│ B - Flow Ctrl Pkg - FA00H G - External Path - FDF4H │

│ Size (# recs)- 4 # of Entries - 5 │

│ C - I/O Package - EC00H H - Shell Stack - FD00H │

│ Size (# recs)- 12 # of Entries - 4 │

│ D - Res Cmd Proc - F200H Entry Size - 32 │

│ Size (# recs)- 16 I - Msg Buffer - FD80H │

│ E - Command Line - FF00H J - Ext. FCB - FDD0H │

│ Size (bytes) - 208 K - Ext. Stack - FFD0H │

│ │

│ Selection : │

└───────────────────────────────────────────────────────────────────────────┘

Когда все действия по настройке или проверке будут завершены, в любом меню, ввод return без выбора приведет к возврату к предыдущему экрану меню (главному меню из более низких экранов) и запустит процесс сборки, если в главном меню будет введен только один return. Сначала будут считаны все указанные файлы для определения их размеров, и выполнены вычисления адресов внутренней памяти. Вас спросят, хотите ли вы, чтобы BPBUILD использовал оптимальные адреса для максимального размера области транзитных программ (автоматический размер) или использовать значения, определенные в Меню 2.1. Если ввести “N” в этом месте, сборка будет происходить в предположении, что вы хотите использовать значения из меню 2.1. Вторая подсказка спросит, хотите ли вы создать систему с сегментами "стандартных" размеров. Это относится к стандартным размерам CP/M 2.2 16 записей (2K) для командного процессора и 28 записей (3.5K) для базовой дисковой операционной системы (BDOS). Если вы ответите “Y”, и сегменты меньше или равны, чем эти размеры, то система будет построена, с учетом этих размеров системных сегментов. Поскольку многие некорректные, но очень популярные программы предполагают старые размеры сегментов CP/M, обычно должен использоваться этот параметр. Если выбран вариант автоматического назначения размеров, BPBUILD автоматически выполняется до своего завершения и возвращает в приглашение командного процессора.

Чтобы получить максимальную область транзитных программ со "стандартными" размерами сегментов, самый простой способ выполнить BPBUILD с утвердительным ответом (“Y”) на запрос автоматического назначения размеров при выходе, затем снова выполнить BPBUILD с созданным образом, ответив на первый запрос “N” - не использовать автоматическое назначение размеров и “Y” на второй запрос, чтобы скорректировать более низкие размеры сегментов для "стандартных" размещений сегментов. Хотя эта процедура несколько громоздка, она приводит к намного меньшей и более быстрой работе утилиты, чем было возможно обеспечить другими способами, и была компромиссом авторов проекта в процессе разработки.

## 6.2 BPCNFG - утилита конфигурации

Гибкость архитектуры B/P BIOS позволяет настраивать вашу систему и требуемые методы работы. Эта утилита объединяет некоторые наиболее общие и важные функции адаптации в одной утилите. Утилита конфигурации B/P BIOS BPCNFG, обеспечивает простое, управляемое с помощью меню средство адаптации образа жесткого диска, загрузочного сектора дисковода для гибких дисков, файла с перемещаемым образом (IMG) и определенных элементов в выполняющейся системе. Использование BPCNFG уменьшает необходимость компиляции нового образа BIOS для простых изменений и увеличивает скорость, с которой изменения могут быть сделаны.

### 6.2.1 Использование BPCNFG

Синтаксис BPCNFG следует стандартным соглашениям, кратко изложенным в [Разделе 1.2](#_Условные_обозначения), и отвечает на стандартный параметр справки состоящий из последовательности двух символов наклонной черты. Синтаксис, с помощью которого вызывается BPCNFG, определяется типом системы, которую вы хотите сконфигурировать. Синтаксис BPCNFG следующий:

BPCNFG // <-- Распечатать встроенную справку

BPCNFG <-- Запустить интерактивный экранный режим

BPCNFG \* <-- Настройка работающей системы

BPCNFG d[:] <-- Настройка диска “d”

BPCNFG [du:]filename[.typ] <-- Настройка файла с образом

Первая форма синтаксиса очевидна и просто распечатывает короткий файл справки, с описанием цели утилиты и синтаксиса использования программы. В интерактивном режиме работы, запуск BPCNFG без параметров, сначала появится вопрос, хотите ли вы настроить память, диск или образ системы B/P BIOS, и задать внутренние параметры этого режима, а также загрузки требуемых данных при необходимости. Если для файла с образом расширение файла не определено - предполагается расширение .IMG.

BPCNFG может настроить параметры в работающей в настоящий момент операционной системы с некоторыми ограничениями. Наиболее существенным ограничением является то, что разделы жесткого диска не могут быть изменены, поскольку пространство для битовых буферов (ALV и CSV) выделяется во время загрузки системы (загрузки) и не могут быть сброшены, когда система уже установлена. За исключением этого, все остальные параметры могут быть изменены и будут действовать до тех пор, пока не будет загружена другая система во время холодного старта компьютера, или загружен файл с образом при помощи LDSYS.

Чтобы прервать выполнение BPCNFG, не изменяя текущих параметров, нажмите ESC или Ctrl-C во время отображения главного меню. Присвоение параметрам произойдет, если во время отображения главного меню будет введен только возврат каретки.

### 6.2.2 Описание экранов меню

BPCNFG - ориентированная на работу с экраном подпрограмма поддержки, использующая атрибуты терминала, определенные в установленном в настоящее время Termcap. В настоящее время в главном меню определены пять основных экранов с несколькими выборами, приводящими к экранам подменю. Первым является главное меню, из которого для изменения выбирается общая категория. Верхняя строка экрана отражает уровень и режим следующим образом:

┌───────────────────────────────────────────────────────────────────────────┐

│ Main Menu - Configuring Image File [A10:T1.IMG ] │

│ Bios Version 2.0 │

│ │

│ 1 System Options │

│ │

│ 2 Character IO Options │

│ │

│ 3 Floppy Subsystem Options │

│ │

│ 4 Hard Disk Subsystem Options │

│ │

│ 5 Logical Drive Layouts │

│ │

│ │

│ Enter Selection : │

└───────────────────────────────────────────────────────────────────────────┘

Каждый из этих пяти выборов приводит к экрану подменю, отображающему определенные элементы, которые могут быть изменены и текущие значения/настройки. Каждый из выбираемых экранов описывается ниже.

#### 6.2.2.1 Экран 1 - Выбор системных параметров

Системные параметры могут быть установлены из меню экрана 1, которое может выглядеть следующим образом:

┌───────────────────────────────────────────────────────────────────────────┐

│ Menu 1 - System Options │

│ │

│ │

│ 1 System Drive = A: │

│ │

│ 2 Startup Command = "START01 " │

│ │

│ 3 Reload Constant = 23040 (5A00H) │

│ │

│ 4 Processor Speed = 9 MHz │

│ │

│ 5 Memory Waits = 0, IO Waits = 0 │

│ │

│ │

│ Enter Selection : │

└───────────────────────────────────────────────────────────────────────────┘

Логический системный диск, выбираемый первым входом, используется BIOS при "Теплом" старте, для загрузки операционной системы при "Холодном" старте, и выходах при возникновении ошибок. Команда Startup обычно - имя COM файла, который выполняет дополнительную инициализацию системы при первоначальном запуске. Такой файл обычно создается, вводом последовательности инструкций (имен файлов и параметров) в скрипт, используя ALIAS, SALIAS, VALIAS или другие подобные инструменты ZCPR3.

Остальные варианты касаются физического оборудования в компьютере и обычно не должны изменяться, за исключением начальной установки. Константа повторной загрузки (Reload Constant) при выборе 3, может быть "точно настроена" слегка изменяя значение. Она обычно используется для корректировки часов реального времени в компьютере, чтобы позволить ему поддержать правильное время. Скоростью процессора (Выбор 4) должна отражать тактовую частоту процессора, округленную до мегагерц. Например, значение, показанное на демонстрационном экране, - на самом деле 9,216 МГц. При изменении тактовой частоты, предлагается вариант, для автоматической корректировки константы повторной загрузки (Выбор 3), чтобы масштабировать ее с помощью корректного коэффициента, основанного на введенном значении тактовой частоты. Это не всегда необходимо, но обычно желательно в системах HD64180/Z-180, использующих прерывания таймера часов.

Число тактов (состояний) ожидания при доступе к памяти и портам ввода-вывода в выборе 5 являются дополнением к любым тактам ожидания автоматически вставляемым аппаратными средствами. Например, Zilog Z180 вставляет один такт ожидания для всех адресов портов ввода-вывода. Поскольку BPCNFG является утилитой общего назначения, она не может знать этого. Поэтому число установленное в этом выборе является дополнением к тактам ожидания, введенным любыми аппаратными средствами.

Также существует шестое меню, но так как непреднамеренное изменение его значений может привести к аномальным и потенциально разрушительным последствиям, оно скрыто и отображается при выборе. Диалог, при выборе варианта 6 следующий:

Enter Selection : 6

-- This is DANGEROUS...Proceed? (Y/[N]) :

Если явно введено “Y”, на экран выводится скрытое меню, наподобие приведенного ниже для YASBEC с PAL адресацией MEM4, позволяющей использовать 1 мегабайт памяти:

┌───────────────────────────────────────────────────────────────────────────┐

│ Menu 1.1 - System Bank Numbers │

│ │

│ │

│ 1 TPA Bank # = 1 │

│ │

│ 2 System Bank # = 3 │

│ │

│ 3 User Bank # = 4 │

│ │

│ 4 RAM Drive Bank # = 5 │

│ │

│ 5 Maximum Bank # = 31 │

│ │

│ │

│ Enter Selection : │

└───────────────────────────────────────────────────────────────────────────┘

Номера банков должны отразить физические номера 32K банков в системе, увеличиваясь от самого малого, или первого доступного к самому большому. Номера банков начинаются с нуля, и могут колебаться от 0 до 255 (0FFH). Банки пользователя и электронного диска могут быть обнулены, если банк пользователя или электронного диска соответственно не требуются. Нужно отметить, что фактические элементы управления этими двумя функциями находится на других экранах и обнуление номеров банков при активных параметрах может иметь негативные последствия.

#### 6.2.2.2 Экран 2 - Параметры символа ввода-вывода.

Экранное Меню 2 позволяет настраивать символ подсистемы ввода-вывода, для включения присвоений логических устройств к физическим устройствам. Пример экрана для YASBEC мог появиться в следующем виде:

┌───────────────────────────────────────────────────────────────────────────┐

│ Menu 2 - Character IO Options │

│ │

│ 1 IOBYTE Assignment: │

│ Console = COM2 │

│ Auxiliary = COM1 │

│ Printer = PIO │

│ │

│ 2 COM1 - 9600 bps, 8 Data, 1 Stop, No Parity, [In(8)/Out(8)] │

│ │

│ 3 COM2 - 19.2 kbps, 8 Data, 1 Stop, No Parity, [In(7)/Out(8)] │

│ │

│ 4 PIO1 - [Out(7)] │

│ │

│ 5 NULL - [In(8)/Out(8)] │

│ │

│ 6 COM3 - 38.4 kbps, 8 Data, 1 Stop, No Parity, [In(8)/Out(8)] │

│ 7 Swap Devices │

│ │

│ Enter Selection : │

└───────────────────────────────────────────────────────────────────────────┘

Выбор 1 определяет, какие из активных устройств назначаются адресуемым функциям, продиктованным IOBYTE. Это место в памяти используется операционной системой для символьных устройств ввода-вывода, обычно состоящих из консоли, принтера (устройство печати в терминологии CP/M) и вспомогательного ввода-вывода (считыватель и перфоратор в CP/M). Конкретные устройства, присвоенные этим логическим функциям, зависят от битовых характеристик IOBYTE, и позиции в таблице, изображенной в этом меню.

Ранние версии B/P BIOS (Версии 1.0 и ранее) распространяемые как "Стартовые системы" и тестовые версии использовали фиксированное количество устройств и другую структуру данных для доступа к устройству. Производственные системы B/P BIOS с номерами версий от 1.1 до 1.9 будут содержать только четыре драйвера устройств, но использовать новые структуры данных. Однако, начиная с версии 2.0, многие символьные устройства могут быть размещены с первыми четырьмя непосредственно доступными через IOBYTE, и возможностью обмена размещения любых четырех устройств в первой позиции для доступа к ним из прикладных программ. Приведенный выше экран изображает типичный дисплей BIOS версии 2.0. Меньшие системы такие как размещены на загрузочных дорожках будут использовать только четыре фиксированных устройства с обычным номером версии 1.1 и не будут включать выборы 6 или 7.

Выбор 1 предлагает возможность выбрать, какие из первых четырех физических устройств присваивается трем логическим функциям определенных в IOBYTE. Дополнительное меню предложит вам конкретные функции и устройства. Пример взаимодействия при выборе 1:

Set [C]onsole, [A]uxiliary, or [P]rinter : C

Set Com[1], Com[2], [P]io, or [N]ull ?

Выборы 2, 3 и 6 в этом демонстрационном экране позволяют конфигурацию последовательных портов, включенных в эту B/P BIOS. Если вы вносите аппаратные изменения в BIOS, следуете за спецификациями в исходном коде, чтобы обеспечить функционирование этой утилиты согласно описания. Для большинства системных установок, выполненных на сегодняшний день, один или более из этих вариантов не будет оказывать никакого эффекта, но возможность использовать эту опцию уже существует. Этот выбор приводит к серии опций, предоставляемых для проверки или изменения. Текущие параметры настройки перечислены как значения по умолчанию и останутся выбранными, если не будет сделана какая-либо запись перед завершающим возвратом каретки (клавиша Enter). Пример взаимодействия на YASBEC:

Configuring COM1 [In(8)/Out(8)]:

Baud Rate (Max=38.4 kbps) = Selections are:

1-134.5 bps 2-50 bps 3-75 bps 4-150 bps

5-300 bps 6-600 bps 7-1200 bps 8-2400 bps

9-4800 bps 10-9600 bps 11-19.2 kbps 12-38.4 kbps

Select [10] :

Data = 8-bits. Change? (Y/[N]) :

Stop Bits = 1-bits. Change? (Y/[N]) :

Parity = None Change? (Y/[N]) :

XON/XOFF Flow = No Change? (Y/[N]) : \* Defaults selected

RTS/CTS Flow = No Change? (Y/[N]) :

Input is 8-bits. Change? (Y/[N]) :

Output is 8-bits. Change? (Y/[N]) :

Выбор четвертого по умолчанию параллельного порта, обычно интерфейс принтера Centronics. Как в предыдущих выборах, не все опции могут быть активными, но будут присутствовать в последовательности конфигурации для сохранения общности BPCNFG. Текущие параметры настройки (в скобках) будут сохранены, при вводе возврата каретки.

Configuring PIO1 [Out(7)]:

XON/XOFF Flow = No Change? (Y/[N]) :

RTS/CTS Flow = No Change? (Y/[N]) :

Output is 7-bits. Change? (Y/[N]) :

Заключительный выбор (появляющийся только в версии 2.0 и более поздних) позволяет обмен устройствами. В демонстрационном Меню 2 выборы 2-5 являются активными устройствами, с выбором 6 существующем в BIOS, но не доступном. Используя последний выбор, COM3 можно обменять с одним из первых четырех, что сделает его активным устройством при выходе из BPCNFG, который вызовет функцию конфигурации устройства. Взаимодействие может выглядеть следующим образом:

Exchange Device : 3 With Device : 6

После этого выбора активные устройства будут COM1, COM3, PIO1 и NULL.

#### 6.2.2.3 Экран 3 - Параметры гибкого диска

В этом экране возможна обширная адаптация подсистемы гибких дисков. Несмотря на то что многие популярные системы не поддерживают все выборы (в основном из-за логики контроллера), все выборы и соответствующие настройки, необходимо установить в этой точке. Кроме того, в то время как такие элементы как скорость шага будут принимать приращение установленного размера (1 мс для скорости шага), у многих систем есть дискретные приращения, которые не соответствуют допустимым значениям. Например, SMS 9266, используемый в SB-180 MicroMint, только примет приращения в 2 мс с 5 1/4" дисководами гибких дисков, WD 1770, используемый в Ampro Little Board продвигается только в 6, 12, 20 и 30 мс, в то время как 1772, используемый в YASBEC, разрешает шаги в 2, 3, 5, и 6 мс. Установленная скорость в BPCNFG надлежащим образом округляется до наиболее подходящего значения, чтобы позволить настраивать диски обобщенным способом. Настраиваемые параметры для дисководов гибких дисков, которые показаны на YASBEC следующие:

┌───────────────────────────────────────────────────────────────────────────┐

│ Menu 3 - Floppy Disk Options │

│ │

│ │

│ 1 Floppy Drive Characteristics: │

│ Drv0 = 3.5" DS, 80 Trks/Side │

│ Step Rate = 3 mS, Head Load = 4 mS, Unload = 240 mS │

│ Drv1 = 5.25" DS, 40 Trks/Side │

│ Step Rate = 4 mS, Head Load = 24 mS, Unload = 240 mS │

│ Drv2 = 5.25" DS, 80 Trks/Side │

│ Step Rate = 3 mS, Head Load = 24 mS, Unload = 240 mS │

│ Drv3 = 5.25" DS, 80 Trks/Side │

│ Step Rate = 6 mS, Head Load = 24 mS, Unload = 240 mS │

│ │

│ 2 Motor ON Time (Tenths-of-Seconds) : 100 │

│ │

│ 3 Motor Spinup (Tenths-of-Seconds) : 5 │

│ │

│ 4 Times to Try Disk Operations : 4 │

│ │

│ │

│ Enter Selection : │

└───────────────────────────────────────────────────────────────────────────┘

Выбор одного из этих четырех пунктов в этом меню предложит вам необходимую информацию, позволяя вам сохранить текущие параметры настроек в качестве значений по умолчанию. Пример дополнительных подсказок, следующих после выбора одного из вариантов на вышеупомянутом экране:

Configure which unit [0..3] : 2

Size 8"(1), 5.25"(2), 3.5"(3)? [2] :

Single or Double-Sided Drive ? (S/[D]) :

Motor On/Off Control Needed ? ([Y]/N) :

Motor Speed Standard or Hi-Density ([S]/H) :

Tracks-per-Side (35,40,80) [80] :

Step Rate in Milli-Seconds [3] :

Head Load Time in Milli-Seconds [24] :

Head Unload Time in Milli-Seconds [240] :

#### 6.2.2.4 Экран 4 - Параметры жесткого диска.

Подсистема жесткого диска B/P BIOS основывается на стандарте интерфейса малых компьютерных систем (SCSI). Также обеспечена обратная совместимость с более ранним Shugart Associates System Interface (SASI) для использования более старых контроллеров. В течение разработки B/P BIOS несколько типов контроллеров с уникальными функциями прозрачным способом были включены в утилиты. Как компромисс между гибкостью и размером программы, было установлено ограничение на использование трех физических устройств жестких дисков в системе. Это ограничение не влияет на 16 возможных логических дисков, которые может обработать компьютер. Ниже представлен пример меню для единственного SCSI диска Conner CP-3100 размером 100 мегабайт:

┌───────────────────────────────────────────────────────────────────────────┐

│ Menu 4 - Hard Disk Options │

│ │

│ 1 Hard Drive Controller = Conner SCSI │

│ │

│ 2 First Drive : Physical Unit 0, Logical Unit 0 │

│ No. of Cylinders = 776, No. of Heads = 8 │

│ │

│ 3 Second Drive : - inactive - │

│ │

│ 4 Third Drive : - inactive - │

│ │

│ │

│ Enter Selection : │

└───────────────────────────────────────────────────────────────────────────┘

Первый доступный выбор позволяет вам выбирать тип контроллера, установленного или используемого в системе. Определение контроллера применяется для всех трех физических дисков, таким образом, должна применяться некоторая осторожность в смешивании различных типов контроллеров в одной системе. Например, контроллер Shugart 1610-3 несовместим с SCSI Seagate или диском SCSI 2, присоединенным непосредственно к шине SASI поскольку информация об инициализации необходимая для дисков 1610-3, не должна быть отправлена в SCSI из-за отличающихся команд. Типы контроллеров, определенные в начальном выпуске B/P BIOS, появляются в выборе 1:

Select Controller Type as:

(0) Owl

(1) Adaptec ACB-4000a

(2) Xebec 1410a/Shugart 1610-3

(3) Seagate SCSI

(4) Shugart 1610-4/Minimal SCSI

(5) Conner SCSI

(6) Quantum SCSI

(будут добавлены еще в более поздних версиях)

(7) Maxtor SCSI

(8) Syquest SCSI

(9) GIDE (IDE/ATA)

#### 6.2.2.5 Экран 5 - Параметры разделов

Меню 5 позволяет располагать на физических дисках дополнительные логические диски и делить физические устройства на несколько логических разделов.

┌───────────────────────────────────────────────────────────────────────────┐

│ Menu 5 - Logical Drive Layout │

│ │

│ A: = Unit 0, 64 Sctrs/Trk, 4k/Blk, 7984k (998 Trks), 1024 Dirs │

│ B: = Unit 0, 64 Sctrs/Trk, 4k/Blk, 20000k (2500 Trks), 1024 Dirs │

│ C: = Unit 0, 64 Sctrs/Trk, 4k/Blk, 20000k (2500 Trks), 1024 Dirs │

│ D: = Unit 0, 64 Sctrs/Trk, 4k/Blk, 54432k (6804 Trks), 2048 Dirs │

│ E: = Floppy 0 │

│ F: = Floppy 1 │

│ G: = Floppy 2 │

│ H: = -- No Drive -- │

│ I: = -- No Drive -- │

│ J: = -- No Drive -- │

│ K: = -- No Drive -- │

│ L: = -- No Drive -- │

│ M: = RAM │

│ N: = -- No Drive -- │

│ O: = -- No Drive -- │

│ P: = -- No Drive -- │

│ │

│ 1 Swap Drives, 2 Configure Partition 3 Show Drive Allocations │

│ │

│ │

│ Enter Selection : │

└───────────────────────────────────────────────────────────────────────────┘

Выбор 1 позволяет замену двух указанных логических дисков. К примеру экран выше показывает систему, которая будет загружаться с жесткого диска. Если система была настроена в качестве системы загружаемой с гибких дисков можно поменять диск A с диском E с помощью выбора 1:

Enter Selection : 1

Swap drive [A..P] : A with drive [A..P] : E

Выбор 2 разрешает, определение логических разделов на жестком диске, и электронном диске, если он активен. Это запрашивает у вас необходимую информацию, на основании которой можно установить внутренние значения BIOS. При преобразовании из существующей системы, SHOWHD (См. [6.19](#_6.19_SHOWHD_)) будет отображать значения, необходимые для ввода для определенного раздела. Пример взаимодействия:

Configure which Drive [A..P] : D

Allocation Size (1, 2, 4, 8, 16, 32k) [4] :

Number of Dir Entries [2048] :

Starting Track Number [6000] :

# Tracks in Partition [6804] :

Physical Unit Number [0] :

Выбор 3 из меню 5 может использоваться в сочетании с выделением для просмотра существующих назначений для данного устройства жесткого диска. При выборе значения 3 вводится требуемый жесткий диск:

Display Allocations for which Hard Drive [0..2] :

и покажет текущие разделы и распределения. Как пример, четыре раздела на Conner CP-3100 изображенном на экране меню 5 выводятся здесь на отдельном экране:

┌───────────────────────────────────────────────────────────────────────────┐

│ Partition Data Hard Drive Unit : 0 │

│ │

│ Drv Start Trk End Trk │

│ │

│ A 2 999 │

│ B 1000 3499 │

│ C 3500 5999 │

│ D 6000 12803 │

│ [any key to continue] │

└───────────────────────────────────────────────────────────────────────────┘

## 6.3 BPDBUG - Утилита отладки B/P BIOS

Эта утилита предоставляет инструмент низкого уровня по образцу DDT Digital Research, но расширенный, чтобы обеспечить более удобный интерфейс пользователя, умение обращаться с мнемониками Z80 и Z180 при дизассемблировании и осуществлять операции с банками памяти, используя интерфейсы B/P BIOS. В то время как описание главным образом ориентировано на вывод на экран, операционная система позволяет также отправить вывод на определенный принтер путем переключения Ctrl-P, который может быть включен и отключен в BPDBUG.

### 6.3.1 Использование BPDBUG

Синтаксис BPDBUG прост, существует только три варианта:

BPDBUG // <-- Напечатать короткую справку

BPDBUG <-- Выполнить BPDBUG

BPDBUG [fn[.ft]] <-- Выполнить BPDBUG с загрузкой названного файла

При выполнении BPDBUG перемещает большую часть кода в верхнюю память непосредственно под BDOS, переписав командный процессор. Если указана загрузка файла как в третьем способе вызова показанном выше, и расширение файла .HEX, то предполагается файл в шестнадцатеричном формате Intel HEX, и он преобразуется в двоичную форму по указанному адресу.

В дополнение к короткому сообщению справки, доступному с параметром двойной наклонной черты, в любое время доступна встроенная сводка команд из основной подсказки BPDBUG, вводя вопросительный знак. Сводка появляется в виде:

┌───────────────────────────────────────────────────────────────────────────┐

│ B/P Bios DBug V 0.3 │

│ -? │

│ │

│ Available commands: │

│ B{ank} Bank# │

│ D{ump} [From [To]] │

│ E{nter} Addr │

│ F{ill} Start End Byte │

│ G{o} Addr │

│ H{ex sum/diff} Word1 Word2 │

│ I{nput} Port# │

│ L{ist} [From [Thru]] │

│ M{ove} Start End Dest │

│ N{ame} FN[.FT] {for Read/Write} │

│ O{utput} Port# Byte │

│ R{ead file} [Offset] │

│ T{race} {Trace mode On} │

│ U{ntrace} {Trace mode Off} │

│ W{rite file} Number\_of\_128-byte\_blocks │

│ X {set breakpoint} Addr │

│ Z{ero breakpoints} │

│ ? {Show this msg} │

└───────────────────────────────────────────────────────────────────────────┘

Квадратные скобки в сводке указывают дополнительные параметры, в то время как текстовые строки и сокращенные имена указывают параметры и типы.

### 6.3.2 Команды BPDBUG

#### 6.3.2.1 Выбор банка памяти [ B ]

Чтобы выбрать банк памяти в диапазоне 0…255, просто введите команду “B”, за которым следует номер банка в шестнадцатеричном формате. Дополнительные пробелы могут быть помещены между буквой команды и номером банка. Синтаксис этой команды следующий:

B[ ]nn

Номер банка, выбранный этой командой станет текущим банком для команд Display (D), Enter (E) и List (L). Если указанный номер банка превысит самый большой номер банка, физически существующий в системе, то номер банка будет определен по последнему банку, определенному в заголовке B/P BIOS.

#### 6.3.2.2 Дамп (отображение) памяти в Hex и ASCII [ D ]

Эта команда выводит на экран содержание памяти в шестнадцатеричной форме и в форме ASCII. Когда текущий байт на дисплее, управляющий символом (менее 20Н), символ точки печатается в столбце ASCII.

Значения по умолчанию для этой команды 100H и банк TPA в качестве начального адреса, и 256 как число байтов, чтобы вывести на экран, если конечный адрес не указан. Для последующего использования команды Dump начальный адрес будет на единицу больше, чем конечный адрес последнего дампа. Номер банка останется определенным в последний раз, или банк TPA, если никогда не изменялся в сеансе. Используется следующий синтаксис:

D[ ][Начальный\_адрес] [Конечный\_адрес]

Образец вывода, появляющиеся на экране:

-d100 12f

01:0100 ED73 FE03 31FE 0321 5D00 7E23 FE2F 2006 .s..1..!].~#./ .

01:0110 BE11 AD01 2825 2A01 002E 5A7E FEC3 2018 ....(%\*...Z~.. .

01:0120 CDAC 0121 FAFF 197E FE42 200C 237E FE2F ...!...~.B .#~./

#### 6.3.2.3 Ввод значений в памяти [ Е ]

Это команда разрешает вводить значения в память. Точка завершает запись.

E[ ][Начальный\_адрес]

-e100

:41 43 44

:.

#### 6.3.2.4 Заполнение память постоянным значением [ F ]

С помощью этой команды всей области памяти может быть установлено единственное постоянное значение. Должны быть определены все три параметра (Начало, Конец и Значение), и если будет задано меньше параметров команда не будет выполняться. Используется следующий синтаксис:

F[ ] Начало Конец Значение

В качестве примера, следующая команда устанавливает шестнадцать байт от 100H до 10FH в выбранном банке, равные двоичному нулю:

-f100 10f 0

#### 6.3.2.5 Перейти (выполнить) программу в памяти [ G ]

Выполнение может быть запущено с любого произвольного адреса с помощью команды “G”. Если адрес запуска не определен, принимается 100H, так как это - обычный начальный адрес программ, загружаемых в область транзитных программ. Синтаксис команды:

G[ ][Адрес]

#### 6.3.2.6 Шестнадцатеричная сумма и разность [ H ]

Простое шестнадцатеричное сложение и вычитание выполняется с помощью команду “H”. Когда команда выполняется с двумя адресами, их сумма (по модулю 65536) и разность (также по модулю 65536) выводятся на экран в этом порядке. Синтаксис этой команды:

H[ ]Значение1 Значение2

Например если требуется смещение 45H от адреса 0ED3FH, полученный адрес положительный и отрицательный может определяться как:

-hed3f 45 <-- Введено

ED84 ECFA <-- …возвращены сумма и разность

#### 6.3.2.7 Отображение значение из входного порта. [ I ]

Эта команда будет читать требуемый входной порт в диапазоне от 0 до 0FFFFH и отображать результирующий байт. Так как используются 16-разрядный адрес при вычислениях, эта команда будет правильно читать встроенные порты HD64180 и Z180. Синтаксис команды ввода следующий:

I[ ]Номер\_порта

Образец записи и полученный результат с произвольной программой:

-l120 127 <-- Введено

0120 CDAC01 CALL 01AC <-- ..отображено

0123 21FAFF LD HL,FFFA

0126 19 ADD HL,DE

0127 7E LD A,(HL)

Обратите внимание на то, что фактические байты, включенные в дизассемблированные инструкции, также перечислены в отличие от других подобных программ, чтобы предоставить дополнительную информацию для вас. Кроме того, на экран выводится дополнительная пустая строка после всех безусловных переходов и возвратов, чтобы служить визуальным представлением об абсолютном изменении в потоке управления. Используются мнемоники стандартных кодов Zilog Z180.

#### 6.3.2.9 Перемещение содержимого памяти [ M ]

Эта команда позволяет перемещать блоки данных в памяти. осуществляется проверки границ адресов памяти для обеспечения корректной обработки перекрывающихся адреса, так что могут быть выполнены незначительные сдвиги в блоках данных. Синтаксис этой команды выглядит следующим образом:

M[ ]Начало Конец Назначение

#### 6.3.2.10 Задать имя файла для чтения и записи [ N ]

Эта команда используется, чтобы установить имя файла и необязательное расширение для операций чтения или записи. Имя остается активным, пока не изменено, или до выхода из BPDBUG. Используется следующий синтаксис:

N[ ]Имя\_файла[.Расширение]

#### 6.3.2.11 Отправить значение в порт вывода [ О ]

Эта команда формирует дополнение команды ввода описанной выше. Она посылает указанный байт в адресуемый выходной порт. Используется следующий синтаксис:

O[ ]Номер\_порта Значение

Как с всеми параметрами, если указано больше цифр, чем необходимо, только последние два (для байта) или четыре (для адреса) используются в выражении.

#### 6.3.2.12 Чтения файла в память [ R ]

По умолчанию эта команда читает файл, определенный командой “N” в память по адресу 100H (если смещение не определено), или в начальный адрес со смещением +100H. Значение смещения должно быть определено в шестнадцатеричном формате. Синтаксис команды чтения:

R[ ][Смещение]

Когда файл будет загружен, вы будете проинформированы о текущих настройках адреса по умолчанию для начала текущей памяти (значение PC) и байта после последнего, загруженного командой чтения (Next). Дисплей мог бы появиться как:

Next PC

0880 0100

#### 6.3.2.13 Активация режима трассировки [ T ]

Для оказания помощи в отладке программ, включена функция трассировки, которая активируется с помощью этой команды. После обнаружения точки останова (См. команду X ниже) программа переходит в режим трассировки, в котором каждая инструкция обрабатывается, и на экран выводится состояние процессора вместе с дизассемблированным списком инструкций. Ввод одной буквы “T” активирует режим трассировки.

Фрагмент программы, выполненной с активированной трассировкой:

S0Z0H0P0N1C0 A=00 BC=0000 DE=0000 HL=0000 SP=0100

IX=A4AE IY=FFFE 0100 C30B01 JP 010B

S0Z0H0P0N1C0 A=00 BC=0000 DE=0000 HL=0000 SP=0100

IX=A4AE IY=FFFE 010B 2A0500 LD HL,(0005)

S0Z0H0P0N1C0 A=00 BC=0000 DE=0000 HL=52C3 SP=0100

IX=A4AE IY=FFFE 010E CDBD07 CALL 07BD

S0Z0H0P0N1C0 A=00 BC=0000 DE=0000 HL=52C3 SP=00FE

IX=A4AE IY=FFFE 07BD 7C LD A,H

S0Z0H0P0N1C0 A=52 BC=0000 DE=0000 HL=52C3 SP=00FE

IX=A4AE IY=FFFE 07BE B5 OR L

S1Z0H0P0N0C0 A=D3 BC=0000 DE=0000 HL=52C3 SP=00FE

IX=A4AE IY=FFFE 07BF C8 RET Z

#### 6.3.2.14 Деактивация режима трассировки [ U ]

Эта команда выключает режим трассировки, чтобы последующее выполнение происходило на полной скорости без использование ловушек. Ввод одной буквы "U" отключает режим трассировки.

#### 6.3.2.15 Записать файл для хранения [ W ]

Эта команда является дополнением к команде чтения, описанной выше. Она предполагает, что данные для записи начинается с 100H и записывает указанное количество 128-байтных блоков в последний файл, заданный с помощью команды “N”. Синтаксис этой команды следующий:

W[ ]Число\_блоков

#### 6.3.2.16 Установить точку останова [ Х ]

Эта команда используется, для отметки мест в пределах программы, которая будет выполняться под BPDBUG, которая при выполнении, временно остановит выполнение и либо вернется в командную строку BPDBUG или напечатает выходную информацию трассировки. В любой момент времени активными могут быть до двух контрольных точек. Синтаксис команды:

X[ ]Адрес

#### 6.3.2.17 Очистка контрольных точек [ Z ]

Эта команда удаляет все точки останова с помощью команды X, упомянутой выше. Ввод одной буквы “Z” удаляет все контрольные точки.

#### 6.3.2.18 Вывод справки на экран [ ? ]

Ввод единственного вопросительного знака (“?”), в качестве команды выводит на экран встроенную справку, содержащую сводку команд, доступных в BPDBUG.

## 6.4 BPFORMAT - утилита форматирования дискет

BPFORMAT - подпрограмма общего назначения для форматирования гибких дисков в системе B/P BIOS. Она автоматически адаптируется к определенному оборудованию, используемому в вашем компьютере для представления единого интерфейса для широкого диапазона платформ, и включает возможность форматировать диски в форматах, не реализованных на вашем компьютере. Эта возможность позволяет вам форматировать диски для обмена с другими пользователями в их родном дисковом формате, с использованием той же библиотеки форматов, используемой программой EMULATE (см. [6.8](#_6.8_EMULATE_Утилита)).

Эта программа предназначена только для B/P BIOS и не будет работать в системах с другими BIOS. Она одинаково работает в системах с переключением банков памяти и без, и со многими типами доступных микросхем контроллеров физических дисков. В начальной версии поддерживаются следующие типы контроллеров:

765 1692 1770 1771 1772 1790 1791

1792 1793 1795 2790 8473 9266

### 6.4.1 Использование BPFORMAT

#### 6.4.1.1 Встроенные форматы

Самый простой способ отформатировать дискеты состоит в том, чтобы использовать один из форматов, включенных в работающую сейчас B/P BIOS. BPFORMAT может быть вызвана, просто введя название программы и следующие за ним букву диска и двоеточие:

BPFORMAT D:

При необходимости имя каталога может заменить букву диска и будет отформатирован диск, связанный с именем каталога. Например, если каталог с именем WORK: определяет диск C:, область пользователя 10, следующая команда:

BPFORMAT WORK:

будет форматировать диск C:. При вызове любым из выше приведенных способов, BPFORMAT перечислит встроенные форматы, доступные для этого диска из тех, которые включены в файл DPB.LIB (и опционально, DPB2.LIB, см. Разделы [4.1](#_Установка_на_дорожки) и [4.2](#_4.2_Установка_образа_1)). Только 80-дорожечные форматы будет предлагаться для 80-дорожечных дисков и так далее, будут представлены только те форматы, которые точно соответствуют характеристикам дисков. Например, предложения для 5.25" дисковода с 40 дорожками могут быть следующими:

Available formats are:

A - Ampro DSDD B - Ampro SSDD

Select format (^C to exit) :

В качестве меры предосторожности против непреднамеренного форматирования дискет, появляются запросы подтверждения, при выполнении программы, и существует возможность прервать программу форматирования и вернуться в командный процессор. В примере выше ввод Ctrl-C вместо выбора формата прервет операцию форматирования.

#### 6.4.1.2 Библиотеки форматов

Если дискету необходимо отформатировать в формате, который не поддерживается встроенными спецификациями, которые имеются в работающей BIOS, может использоваться библиотека форматов, используемая, EMULATE (см. [6.8](#_6.8_EMULATE_Утилита_1)). Причины для использования библиотеки могут колебаться от необходимости форматировать в режиме используемом на другом типе компьютера, встроенный выбор которых недоступен, до измерения ограничений, при адаптации системы для установки на дорожки начальной загрузки. Какой бы ни была причина, эта гибкость является неотъемлемым свойством B/P BIOS и уточняется указанием параметра “L” при вызове BPFORMAT. Для определении необходимого диска в командной строке, как в любой команде выше, просто добавьте символ параметра в конце (с дополнительной наклонной чертой):

BPFORMAT D: L

Если вы хотите ввести букву диска как часть вызова программы, символ наклонной черты становится обязательным для информирования BPFORMAT, что вы указываете параметр библиотеки вместо накопителя “L:”. Таким образом вызов становится:

BPFORMAT /L

При выполнении с указанием параметра библиотеки, вам будет представлено меню форматов, которые могут быть использованы с физическим диском, определённым в заголовке BIOS. Образец меню выглядит следующим образом:

Available formats are:

A - Actrx SSDD B - Ampro SSDD C - VT180 SSDD D - H-100/4 1D

E - H89/40 1S F - H89/40 1D G - H89/40 1X H - Kaypro 2

I - Osborne 1S J - Osborne 1D K - Ampro DSDD L - H-100 DSDD

M - H89/40 2D N - H89/40 2X O - QC-10 DSDD P - Kaypro 4

Q - MD-3 DSDD R - PMC-101 S - Sanyo 1000 T - TV 802/803

U - XBIOS-3 2D V - XL-M180 T2

Select format (^C to exit) :

Ввода одной из букв соответствующей формату установит все параметры и приступит к операции форматирования. Ввод Ctrl-C на данном этапе вернет вас в командный процессор и предотвратит любое случайное форматирование диска.

Источник на ассемблере для библиотеке форматов присутствует в пакете B/P BIOS как средство добавления форматов, не включенных в дистрибутивный пакет по умолчанию, или экспериментов с новыми форматами.

### 6.4.2 Конфигурация

Существуют две опции адаптирующие BPFORMAT для работы методом, который вы считаете самыми удобными. Первой является опция Quiet, которая минимизирует посторонний вывод на консоль. Она устанавливается логическим флагом, состоящим из байта расположенного со смещением 22 (16H) байт от начала программы. Нулевой байт расположенный по этому адресу означает операцию Verbose, в которой все определенные подсказки и информация о статусе выводятся на экран. Ненулевое значение (обычно 0FFH) указывает, что вывод на консоль должен быть минимизирован, на экран выводится только существенная информация.

Вторая опция служит для выбора имени и расширения файла, который будет использоваться в качестве библиотеки форматов, при указании параметра “L”. По умолчанию значение этой записи - ALIEN.DAT (отформатированное в форме FCB), которое также используется программой EMULATE. Это поле начинается со смещением 23 (17H) байта от начала программы.

### 6.4.3 Сообщения об ошибках BPFORMAT

Must be wheel to FORMAT!!!

В качестве меры безопасности, только пользователи с привилегиями колеса могут форматировать дискеты. Это сообщение об ошибке идентифицирует попытку без соответствующего разрешения.

\*\*\* ERROR ! Not B/P Bios, or Old Version !

В большинстве случаев эта ошибка будет замечена, если будет предпринята попытка, отформатировать диск с BIOS отличным от B/P BIOS. Если некоторые обязательные структуры данных были изменены, или если будет предпринята попытка, выполнить программу под одной из ранних тестовых версий B/P BIOS, то это сообщение будет также будет выведено на экран.

\*\*\* ERROR ! The selected format is not supported by FORMAT!

Эта ошибка будет выведена на экран, если формат из библиотеки форматов несовместимым с указанным диском, например, 5.25" формат выбран для 8" диска.

\*\*\* ERROR ! The detected FDC is not supported by FORMAT!

BIOS сообщила о контроллере гибких дисков (FDC), который не включен в список контроллеров, поддерживаемых BPFORMAT. Чтобы уточнить список поддерживаемых контроллеров, просмотрите внутреннюю справку при помощи параметра двойной наклонной черты.

\*\*\* ERROR ! Disk is Write Protected!

Была предпринята попытка форматирования с защитой от записи на вкладке установленной (для 5,25 "), снятой (для 8") или в положение защиты (3,5 ") указанного диска. Установите диск для чтения/записи, изменив физическую защиту и попытайтесь снова отформатировать диск.

\*\*\* ERROR ! Disk won't recalibrate

Головки дисковода не могут быть возвращены в позицию дорожки 0. Часто эта ошибка происходит из-за сбоя в механизме привода, но также может быть вызвана деформацией дискеты или неисправностью кабеля диска.

Format Error : xx

Обнаружена ошибка во время форматирования. "xx" шестнадцатеричный байт, возвращаемый частью подпрограммы форматирования BIOS, и биты установленные в значение "1" представляют собой код ошибки, который поддается расшифровке на основании данных FDC или списка с кодами ошибок в инструкции по программированию.

+++ Can't Open : fn.ft

BPFORMAT не может открыть файл библиотеки форматов. Чтобы получить доступ к библиотеке форматов, она должна быть либо в библиотекой по умолчанию, или быть доступна программе форматирования установкой общего бита или пути ZSDOS.

No formats available for this drive!

Эта ошибка будет появляться, если отсутствуют поддерживаемые форматы (внутренние, или из библиотеки форматов, в зависимости от способа вызова программы) для указанного диска. Например, если BPFORMAT вызывается для форматирования с использованием внутренних форматов, и если все внутренние форматы для 5,25/3.5" дисков, и в качестве назначения указывается диск 8", то форматы будут не доступны.

## 6.5 BPSWAP - Утилита обмена логических обозначений дисков

Эта утилита позволяет обменивать буквы дисков определяющие два логических диска или раздела в системе. Она выполняет любые операции, необходимые для правильной настройки операционной системы с учетом переопределения приводов и перерегистрации обоих дисков, с помощью DOS функции 37 для принудительного восстановления битовой карты распределения.

BPSWAP - это утилита B/P BIOS и не будет работать в любой другой системе. Она может выполняться в интерактивном режиме, полностью в "экспертном" режиме с параметрами передающимися в командной строке, или их комбинации, где первая буква передается в командной строке, а вторая поступает в ответ на запрос. Если она запущена в интерактивном режиме, ввод Ctrl-C вместо буквы диска прервет выполнение программы и управление вернется в командный процессор. BPSWAP является программой повторно исполняемой в ZCPR с помощью команды "GO".

### 6.5.1 Использование BPSWAP

#### 6.5.1.1 Интерактивные операции

Чтобы выполнить BPSWAP в интерактивном режиме запроса/ответа, просто вызовите программу, введя ее имя:

BPSWAP

Программа проверит, что система работает под управлением B/P BIOS, соберет внутренние данные из операционной среды, а также отобразит подсказку:

First Drive to Swap [A..P] :

В данный момент должна быть введена буква диска (прописная или строчная) в указанном диапазоне от "A" до "P". Все недопустимые символы, за исключением Ctrl-C, который прерывает программу, приведут к повторным подсказкам для ввода первой буквы диска. Когда допустимая буква диска обнаружена, будет затребована вторая аналогичной подсказкой:

Second Drive to Swap [A..P] :

BPSWAP отвечает на ввод в этом месте способом аналогичным первому, неоднократно запрашивая допустимую букву или символ аварийного прекращения работы. Когда допустимая буква получена, каждый логический диск переназначается физическим определениям другого.

#### 6.5.1.2 Работа в командной строке

BPSWAP может принять и проанализировать буквы дисков, переданные ему из командной строки, чтобы включать обмены диска в сценарии запуска или другие команды псевдонимов. Чтобы вызвать программу этим способом, введите название программы с двумя буквами дисков в диапазоне от "A" до "P" с разделителем между каждым полем. Каждые из букв дисков могут сопровождаться дополнительным двоеточием. Символами разделителя являются - табулятор, пробел и запятая. Полный синтаксис команды:

BPSWAP <Drv1>[:] <tab| |,> <Drv2>[:]

Для иллюстрации, приводятся следующие допустимые команды, выполняющие BPSWAP

BPSWAP A: E: <-- Поменять диск E с A

BPSWAP D,H <-- Поменять диск D с H

Если обнаружен недопустимый символ для одной или обеих букв дисков при вызове в режиме командной строки, программа автоматически переходит в интерактивный режим и соответствующая подсказка(ки) будет дана для ввода допустимой буквы диска(ов). Эта функция позволяет использовать гибридный режим работы с уточнением, при котором первая буква передается в командной строке, а вторая вводится в ответ на вторую строку подсказки.

### 6.5.2 Сообщения об ошибках BPSWAP

Единственное сообщение об ошибке, которое может быть распечатано BPSWAP при проверке присутствия B/P BIOS. Любая попытка выполнить эту утилиту с другой BIOS приводит к ошибке:

+++ Not B/P Bios ... aborting +++

после чего программа будет прервана и управление возвращается к командному процессору. Если отображается эта ошибка назначение дисков не происходит.

## 6.6 BPSYSGEN - утилита генерации системы

BPSYSGEN - наша универсальная версия классической программы SYSGEN, используемой, чтобы поместить исполнимый образ системы на загрузочные секторы гибкого или жесткого диска. Она использует информацию, предоставленную BIOS в форме данных DPB/XDPB (см. [5.2.3](#_5.2.3_Блок_параметров_3)), которые определяет характеристики физического и логического диска, чтобы записать информацию о системе из системных дорожек одного диска на другой, или из образа, произведенного MOVxSYS (см. [6.16](#_6.16_MOVxSYS_-_1)) на дорожки начальной загрузки диска.

### 6.6.1 Использование BPSYSGEN

#### 6.6.1.1 Интерактивные операции

Основной интерактивный режим инициируется, просто введя имя программы в командной строке:

BPSYSGEN

Сначала вам будет предложено ввести исходный диск, с которого будет получен загрузочный образ системы, а затем диск назначения для записи образа. Чтобы обеспечить визуальную подсказку о выполнении программы, на экран выводится ряд точек представляющих физический сектор данных. После окончания работы, программа выходит в строку командного процессора.

Двоичный файл, созданный MOVxSYS (см. [6.16](#_6.16_MOVxSYS_-_2)) может быть размещен на системные дорожки жесткого диска или дискеты, указав имя файла в качестве аргумента командной строки:

BPSYSGEN B:ZSDOS64.BIN

при активации этим способом, вам будет предложено ввести букву диска назначения, после того, как BPSYSGEN загрузит файл изображения и проверит является ли он допустимым образом системы. Кроме того, вы можете заменить имя файла буквой диска, сопровождаемой двоеточием, чтобы автоматически загрузить изображение из определенного диска и получить запрос диска назначения.

#### 6.6.1.2 Работа в командной строке

Одна операция может быть полностью определена с помощью аргументов командной строки, тем самым избегая приглашений о задании диска. При вызове этим способом, первый аргумент указывает источник системы (обозначение диска или файла), второй аргумент - спецификацию диска на который предполагается разместить загрузочный образ системы. Синтаксис командной строки следующий:

BPSYSGEN {d: | fn[.ft]} d: <-- d диск в диапазоне [A..P]

### 6.6.2 Сообщения об ошибках BPSYSGEN

\*\*\* Read Error

Неисправимая ошибка чтения или загрузочных дорожек диска, или указанного загрузочного файла.

\*\*\* Bad Source!

Исходный диск не существует или не может быть выбран.

\*\*\* Write Error

Неустранимая ошибка при записи загрузочных дорожек указанного конечного диска.

\*\*\* Bad Destination!

Привод назначения не существует или не может быть выбран.

\*\*\* No System!

Отсутствуют допустимые системные дорожки на диске источнике или назначения, или было обнаружено аномальное условие (больше чем 3 зарезервированных дорожки).

\*\*\* Can't Open Source File!

Указанный файл с образом загрузки не найден или произошла ошибка во время попытки открытия файла.

## 6.7 COPY - универсальная утилита копирования файлов

COPY.COM - программа копирования файлов происходит от инструмента ZCPR3 MCOPY написанного Ричардом Конном {Richard Conn}. Она сочетает множество модификаций от Брюса Моргана {Bruce Morgen} и других в MCOPY48 с дальнейшими улучшениями в духе среды ZSDOS. Поддерживается метки даты файла для полного спектра возможностей временных меток, обеспеченных ZSDOS. Теперь поддерживается определяемый пользователем “Список исключений”, чтобы предотвратить копирование определенных файлов или групп файлов, и были добавлены две опции упрощающие резервное копирование файлов с архивным битом. Также, программа COPY более удобна для пользователя, чем MCOPY и обеспечивает повышенный уровень проверки ошибок и обратную связь с пользователями.

COPY работает только под управлением командной строки или в экспертном режиме. Как и в случае с другими утилитами ZSDOS, COPY отображает короткое сообщение справки при вызове с аргументом в виде двойной косой черты, как описано в [Разделе 1.2](#_Условные_обозначения_1). Сообщение справки также включает список доступных параметров наряду с действием каждого из них, если они заданы в качестве аргумента командной строки.

Хотя COPY готова работать без предварительной процедуры установки, вы можете изменить параметры по умолчанию, чтобы настроить ее на свой стиль работы. Таким образом, вы можете свести к минимуму количество нажатий клавиш, необходимых для выполнения однотипных операций, избегая указания большого числа параметров в командной строке. Чтобы задать условия по умолчанию, обеспечьте доступность файлов COPY.COM, COPY.CFG и ZCNFG.COM в системе и выполните программу ZCNFG, как описано в *Разделе 4.8 Руководства ZSDOS 1.0*.

### 6.7.1 Использование COPY

Основной синтаксис COPY следует оригинальному формату CP/M, перечисляя диск/пользователь назначения, знак равенства, затем исходный диск/пользователь и имя файла. Альтернативный синтаксис, добавленный Брюсом Морганом {Bruce Morgen} в MCOPY48, позволяет определять копирование в форме “Источник-Назначение”, популярной в MS-DOS. В этой альтернативной форме, вы сначала вводите исходный диск/пользователь и имя файла, пробел, и затем диск/пользователь назначения и дополнительное имя файла. Используя символические обозначения, синтаксис в итоге имеет вид:

COPY dir:[fn.ft]=[dir:]fn.ft,... [/]options

или

COPY [dir:]fn.ft dir:,... [/]options

Если имя файла назначения не определено, много уникальных файлов могут быть скопированы в указанный каталог, соединив исходные файлы, разделенные запятыми. Если имя файла назначения определено, то имена файлов источника и назначения и их расширения не должны содержать подстановочных символов. Эта популярная функция “Rename” (переименование) при копировании была очень востребованным дополнением к утилите копирования ZSDOS. Параметры изменяющие действия COPY могут быть добавлены после списка исходных файлов.

Еще один метод передачи файлов был сохранен от исходных корней MCOPY. Если диск/пользователь назначения не представлен в параметрах командной строки, то все файлы, на которые ссылаются, будут скопированы в расположение диска/пользователь по умолчанию, которые содержатся в части заголовка программы COPY. Расположение по умолчанию - диск B, пользователь 0 в программе дистрибутива, но может быть изменено программой конфигурации, упомянутой выше. Если необходимы параметры с этим синтаксисом, разделитель параметров в виде наклонной черты обязателен. Синтаксис для этого метода в итоге следующий:

COPY [dir:]fn.ft,... /options

Различные варианты конфигурации, подробно описанные далее, позволяют настроить COPY для соответствия вашему стилю. Например, индикация статуса каждой операции может быть подавлена с помощью “Quiet” ("Тихого") режима, проверка соответствия скопированных файлов оригиналу (или, по крайней мере, проверка кода той же ошибки) может быть включена или отключена, и т.д. Если метод маркировки даты и времени активен в ZSDOS или ZDDOS, оригинальная информация меток будет передана в файл назначения. Следующие примеры в “Verbose” ("Подробном") режиме работы будет служить для иллюстрации операции копирования файла с текущего привода и области пользователя на тот же диск пользователю 10.

COPY ZXD.COM 10:

COPY Version 1.71 (for ZSDOS)

Copying C2:ZXD .COM to C10:

-> ZXD .COM..Ok (Dated) Verify..Ok

0 Errors

В этом случае файл с тем же именем не существует в месте назначения, но некоторые формы временных меток файла были активны, поэтому информация метки источника успешно был передана в пункт назначения. Выполнение того же действия с другим синтаксисом теперь производит:

COPY 10:=ZXD.COM

COPY Version 1.71 (for ZSDOS)

Copying C2:ZXD .COM to C10:

-> ZXD .COM Replace Same (Y/N)? Y..Ok (Dated) Verify..Ok

0 Errors

Так как программа COPY теперь обнаружила файл назначения с тем же именем, и файл временных меток, а также действовала проверка дублирования (другой вариант флага), COPY сравнила даты последнего изменения для исходного и конечного файлов. Найдя совпадение, выдала запрос “Replace Same” (заменить то же) и получила ответ (Y)es для копирования файла в любом случае. Другими запросами, в зависимости от результатов сравнения дат являются “Replace Older” (заменить старый), что означает, что существует старый файл в месте назначения и “Replace Newer” (Заменить новым), что означает, что вы пытаетесь заменить более новый файл в месте назначения его более старой версией.

Производится аналогичная проверка на наличие ошибки, если обнаруживается дубликат файла с установленным битом атрибута общий. Если общедоступный файл обнаружен на диске назначения, об этом будет напечатано предупреждение. Ответ Yes для замены в этом месте приведет к ошибке только для чтения, если ZSDOS не был настроен, чтобы разрешить запись в общие файлы (смотрите *Раздел 2.8.3) Руководства ZSDOS 1.0*.

Как отмечалось ранее, у программы COPY нет интерактивного режима работы по сути, но параметр “Inspect” (инспектирование) обеспечивает способ выбора файлов для копирования, отчасти интерактивным способом. В этом режиме все файлы, выбранные согласно спецификации файлов в командной строке, выводятся на экран, по одному, и вы можете ввести “Y”, чтобы скопировать файл, “N”, чтобы не копировать файл или “S”, чтобы прекратить копирование остальной части выбранных файлов. Пример, копирования всех файлов из текущего диска и области пользователя в область пользователя 10:

COPY \*.\* 10: /I

COPY Version 1.71 (for ZSDOS)

Copying C2:????????.??? to C10:

Inspect -- Yes, No (def), Skip Rest

BU16 .COM - (Y/N/S)? Y

BU16 .MZC - (Y/N/S)? N

COPY .COM - (Y/N/S)? Y

COPY .Z80 - (Y/N/S)? S

При работе в подробном режиме, статус каждого файла будет напечатан как прогресс копирования.

### 6.7.2 Параметры программы COPY

Несколько параметров в виде символов, доступны для настройки операций копирования. Большинство из этих параметров может быть установлено как условия по умолчанию с помощью утилиты конфигурации ZCNFG Аль-Хоули {Al Hawley}. Кроме того, вы можете ввести любой из них в командной строке для изменения функции одной операции. Символы параметров командной строки, следующие:

**A** Архивный,

**E** Проверка существования файла,

**I** Инспектирование файлов,

**M** Многократное копирование,

**N** Отсутствие замены существующего файла,

**О** Тест существование в R/O файлов на месте назначения

**Q** Тихий режим,

**S** Исключение системных файлов,

**V** Проверкакопирования,

**X** Архивный, только если файл существует.

Из краткого синтаксиса, приведенного выше, вы заметите, что стандартный параметр разделителя, в виде косой черты, является необязательным, если обе спецификации источника и назначения, перечислены в командной строке. Если указана только одна спецификация, при копировании на диск по умолчанию, разделитель является обязательным. Каждый из параметров описывается в следующих параграфах.

#### 6.7.2.1 Параметр архивный

Когда этот параметр активен, либо путем указания в командной строке, или по умолчанию, будут выбраны только файлы, у которых не установлен атрибут Архивный. После того, как выбранные файлы будут скопированы, атрибут "Архивный" будет установлен на исходный файл, чтобы указать, что файл был “Заархивирован”. Если используется в сочетании с настройками по умолчанию диска и области пользователя, параметр “A” обеспечивает простой способ архивирования файлов в одной области пользователя. По умолчанию этот параметр выключен, не обеспечивая контроля выбора атрибута архивный. Добавление параметра “A” в командной строке меняет настройки параметра.

Следует отметить, что этот параметр несовместим с параметром «M» (Многократное копирование). Первая операция копирования установит архивный бит для выбранных файлов, и они не будут появляться в последующих копиях.

#### 6.7.2.2 Параметр проверки существования файла

Этот параметр управляет тестом проверки существования файла с тем же именем на диске назначения. Добавление параметра “E” в качестве аргумента командной строки изменяет установленные настройки. По умолчанию в дистрибутивной версии ZSDOS параметр включен, или осуществляется проверка существующих файлов. Этот параметр не влияет на проверку общих файлов на диске назначения, которая всегда активна.

#### 6.7.2.3 Параметр инспектирования файлов

Как показано выше, параметр “I” предоставляет средство выборочного копирования файлов, без ввода имени каждого файла. В дистрибутиве по умолчанию этот параметр выключен, то есть проверка выбранного списка файлов не происходит. Указание этого параметра в списке аргументов командной строки изменяет существующую настройку.

#### 6.7.2.4 Параметр многократного копирования

Этот параметр может использоваться, чтобы скопировать файл или группу файлов на тот же диск несколько раз, делая несколько копий одной и той же группы файлов на различные диски. Прежде чем начинается создание каждой копии, выводится подсказка и вы можете при появлении подсказки прерваться или заменить диски прежде, чем начать копирование. В дистрибутиве по умолчанию этот параметр выключен, т.е. многократное копирование отсутствует. Добавление параметра “M” к списку параметров командной строки инвертирует настройки этого параметра.

#### 6.7.2.5 Параметр отсутствия замены существующего файла

После добавления в качестве аргумента командной строки, параметр “N” программа не допустит замены файла, который уже существует на диске/области пользователя назначения. Этот параметр нельзя настроить и он всегда предполагает одно и то же начальное состояние, когда вызывается COPY. По умолчанию начальное состояние этого параметра разрешает замену существующих файлов.

#### 6.7.2.6 Тест файла только для чтения

Этот параметр, при добавлении в качестве аргумента, переворачивает настроенный параметр флаг, который проверяет наличие файлов, удовлетворяющих указанному имени и расширению с атрибутом только для чтения. Если этот флаг является активным и доступный только для чтения файл найден, удовлетворяющий этим критериям, файл не будет перезаписан автоматически. Флаг “E” (существование файла) по-прежнему будет диктовать, как обрабатываются другие файлы.

#### 6.7.2.7 Параметр тихого режима

При использовании в системе ZCPR3, этот параметр вызывает реверсирование действия флага ZCPR3 “Quiet”. Если флаг ZCPR3 “Quiet” (тихий) активен, COPY с параметром “Q” работает в “Verbose” (многословном) режиме. Если вы не используете ZCPR3, или среда ZCPR3 определяет флаг “Quiet” как неактивный, то этот параметр отключит подавление ненужных консольных сообщений тихого режима копирования. У этого параметра нет условий по умолчанию, и он действует только для единственного вызова COPY.

#### 6.7.2.8 Параметр исключения системных файлов

Этот параметр определяет, будут или нет файлы с установленным атрибутом системный обрабатываться программой COPY. В дистрибутиве по умолчанию параметр выключен, разрешая добавление системных файлов в список копирования и разрешая копирование таких файлов. Подразумеваемое действие может быть настроено, как описано ниже, и действие по умолчанию может быть отменено путем добавления параметра “S” в списке параметров командной строки.

#### 6.7.2.9 Параметр проверки копирования

Чтобы добавить уверенности, в отсутствии ошибок в ходе операции копирования, может быть активирован параметр “Verify” (Проверка). Когда параметр активен, вычисляется код циклической проверки избыточности (CRC) файла назначения. Этот код затем сравнивается со значением кода вычисленного при чтении исходного файла. Если два значения совпадают, вы можете быть вполне уверены, что конечный файл является удостоверенной копией исходного файла. В дистрибутиве по умолчанию этот параметр имеет значение “True” (Истина), чтобы проверять каждый скопированный файл. Действие этого параметра может быть изменено конфигурацией или инвертировано, добавлением “V” к списку параметров командной строки.

#### 6.7.2.10 Параметр архивный, только если файл существует

Иногда, вы, возможно, захотите обновить часто архивируемые файлы с этой же целью более простым способом, чем указанием имени каждого файла, или использования параметра “inspect”. Параметр “X”, создан именно для этой цели. Если этот параметр добавлен, программа COPY сначала просматривает исходный каталог для файлов, которые не были заархивированы, затем проверяет каталог назначения для каждого файла. Если соответствие найдено, то файл копируется и архивный бит устанавливается на исходный файл. Нет конфигурируемой настройки для этого параметра, действие которого предполагается всегда выключенным перед началом копирования.

## 6.8 EMULATE Утилита эмуляции сторонних дисков

EMULATE блокирует любой или все дисководы для указанных форматов, родных или посторонних, из базы данных форматов. Он также может быть использован для отображения текущих настроек и восстановления дисков для автоматического выбора, если BIOS была скомпилирована с возможностью авто-выбора (см. [4.2](#_4.2_Установка_образа_2)). Информация о формате дискеты содержится в файле под названием ALIEN.DAT, который используется совместно с BPFORMAT (см. [6.4](#_6.4_BPFORMAT_-)). Совместное использование общей базы данных форматов позволяет форматирование, а также чтение и запись дисков более сотни форматов, используемых поставщиками CP/M на протяжении многих лет.

### 6.8.1 Использование EMULATE

Эта утилита может быть использована только с B/P BIOS которая была собрана с активным параметром авто-выбора (AUTOSEL). Это нормальный режим для версий B/P BIOS, хотя в некоторых версиях, размещенных на загрузочных дорожках дискеты может иметь сокращенные дополнения встроенных форматов, чтобы уменьшить размер образа системы (см. [4.3](#_4.3_BIOS_с)). EMULATE может быть выполнена либо в интерактивном режиме запрос/ответ или в "экспертном" режиме командной строки с аргументами, передаваемыми в командной строке. Синтаксис EMULATE следующий:

EMULATE // <-- Распечатать встроенную справку

EMULATE [/]X <-- Список текущих параметров форматов дискет

EMULATE [/]U <-- Возвращение всех дискет в авто-выбор

EMULATE <-- Выполнять в интерактивном режиме

EMULATE d[:] <-- Выберите формат диска d: интерактивно

EMULATE d[:] [nn] <-- Установить диск d: формат запись nn (эксперт)

Чтобы сохранить нумерацию форматов в файле базы данных постоянной, таким образом позволяя экспертный режим конфигурации, все форматы из файла ALIEN.DAT загружаются без проверки допустимости в отношении фактических параметров диска. После того, как выбран формат, необходимые параметры привода (размер диска, число сторон, скорость и количество дорожек) сравниваются с физическими параметрами привода, находящимися в структуре заголовка B/P BIOS (см. [5.2.1](#_5.2.1_Область_конфигурации_1), CONFIG+35). Если выбранный формат может быть размещен на физическом диске, то информация о формате загружается в расширенные поля DPH/DPB для указанного диска и формата, и формат блокируется, чтобы предотвратить повторное назначение при "Теплом" старте.

Следующие форматы в настоящее время включены в файл ALIEN.DAT, исходный код, для которого включен в дистрибутивную версию B/P BIOS ALIEN.LIB:

1 Actrx SSDD 2 Ampro SSDD 3 VT180 SSDD 4 H-100/4 1D

5 H89/40 1S 6 H89/40 1D 7 H89/40 1X 8 Kaypro 2

9 Osborne 1S 10 Osborne 1D 11 Ampro DSDD 12 H-100 DSDD

13 H89/40 2D 14 H89/40 2X 15 QC-10 DSDD 16 Kaypro 4

17 MD-3 DSDD 18 PMC-101 19 Sanyo 1000 20 TV 802/803

21 XBIOS-3 2D 22 XL-M180 T2 23 Ampro SSQD 24 DEC Rainbo

25 Eagle-IIE 26 H89/80 1D 27 H89/80 1X 28 Ampro DSQD

29 Amstrad WP 30 H89/80 2D 31 H89/80 2X 32 XBIOS-4 2Q

33 CCS SSDD 34 IBM 3740 35 Bower 8"1D 36 TTek SSDD

37 Bower 8"2D 38 CCS DSDD 39 TTek DSDD2 40 TTek DSDD1

Текущие набор форматов диска может быть исследован в любое время с помощью параметра “X”, который перечислят 10-и символьные имена форматов, присвоенные каждому дисководу для гибких дисков в системе, или что состояние автоматического выбора. Параметр “U” удаляет все фиксированные форматы, возвращая их к автоматическому выбору.

### 6.8.2 Сообщения об ошибках EMULATE

+++ Can't Open Database File +++

EMULATE не удалось обнаружить файл ALIEN.DAT в текущем каталоге. Решения включают в себя установление общественного атрибут ALIEN.DAT и гарантии того, что путь DOS включает диск, содержащий файл.

+++ Format Not Supported on this drive!

Не требует пояснений. Распространенными причинами этого являются ошибки при выборе 80-дорожечного формата на 40-дорожечный дисковод, или 8" формат для 5.25" привода. Проверьте исходный код ALIEN.DAT, чтобы определить любые необходимые данные по точным требованиям диска для каждого формата.

## 6.9 HDBOOT - утилита загрузки с жесткого диска (заказная)

HDBOOT - это специализированная подпрограмма, которая доступна только для тех компьютеров, в которых реализована возможность загрузки с жестких дисков при холодном старте, таких как компьютеры YASBEC и Ampro Little Board первоначальных версий. HDBOOT это заказная утилита, которая разработана для определенных версий и не будет выполняться в B/P версиях, которые не может распознать. Она изменяет загрузочную запись образа системы для гибких дисках размещённую на диск с помощью BPSYSGEN (см. [6.6](#_6.6_BPSYSGEN_-_1)) для запуска системы, для старта с жесткого диска при включении или перезапуске системы.

### 6.9.1 Использование HDBOOT

HDBOOT чрезвычайно проста в использовании и обращается к структурам данных B/P BIOS целевой системы для получения любых требуемых системных данных, например параметров инициализации для контроллеров типа Shugart/Xebec. При вызове, существующая система проверяется, чтобы убедиться, что она является допустимой версией B/P BIOS. Если допустимая, то вас попросят определить, какое из трех возможных физических устройств SCSI будет доступно, и откуда будет выполнятся автоматическая операция. Типовой экран для успешного выполнения этой утилиты:

┌───────────────────────────────────────────────────────────────────────────┐

│ B/P HDBOOT Utility V1.0 31 Aug 92 │

│ Copyright 1992 by H.F.Bower/C.W.Cotrill │

│ │

│ Configure which unit for Booting [0..2] : 0 │

│ |

│ Target Controller is : Seagate SCSI │

│ ...Reading Boot Record... │

│ ...Writing Boot Record... Ok.. │

│ │

│ A0:BASE>\_ │

└───────────────────────────────────────────────────────────────────────────┘

Следует отметить, что система должна быть размещена на целевом устройстве с помощью BPSYSGEN (см. [6.6](#_6.6_BPSYSGEN_-_2)) перед запуском этой утилиты, или будет выдано сообщение об ошибке и операция прервется.

### 6.9.2 Сообщения об ошибках HDBOOT

\*\*\* No System!

Указанное целевое устройство не содержит действительной системы загрузки. Поместите действительную дорожку загрузки системы на устройство с помощью BPSYSGEN и выполните HDBOOT снова.

\*\*\* Invalid Unit Number \*\*\*

В командной строке указан недопустимый номер устройства,. Это не “0”, “1” или “2”, или устройство не активно.

\*\*\* Invalid Boot Record \*\*\*

Загрузочная запись существующих на указанном устройстве не является допустимой для данного типа компьютера. Нормальной причиной ошибки является отсутствие системы в настоящее время на указанном устройстве или система не является допустимой для данной системы. Обе из них могут быть исправлены путем размещения системы на первый физический раздел устройства с помощью BPSYSGEN (см. [6.6](#_6.6_BPSYSGEN_-_3)).

+++ Image is Not B/P Bios, or Wrong Version +++

Прочтенный образ из загрузочных дорожек указанной системы имеет недопустимую версию B/P BIOS. Двумя наиболее распространенными причинами этого являются: не помещение загрузчика системы на системные дорожки с помощью BPSYSGEN, или изменение структуры фиксированных данных исходного кода BIOS способом, который нарушает стандартное расположение полученной системы, в результате чего она не может быть распознана.

+++ Unit Not Active! Run BPCNFG to Set Drives.

Указанное устройство жесткого диска (0, 1 или 2) не было помечена как активное устройство. Это может быть изменено, сначала выполнив BPCNFG (см. [6.2](#_6.2_BPCNFG_-_11)) на системе памяти исполнителя и повторно вызвав HDBOOT.

+++ Not B/P Bios ... aborting +++

Была предпринята попытка выполнить эту программу в системе, которая не совместима с B/P BIOS. Загрузите систему снабженную B/P BIOS и попробуйте снова.

\*\*\* Read Error

Неисправимая ошибка при попытке чтения целевой системой загрузочной записи. Чаще всего возникает из-за ошибок носителей на первый цилиндр целевого устройства и не может быть устранена. Еще одной причиной может быть неправильное определение физических характеристик контроллера и/или привода.

\*\*\* Write Error

Неисправимая ошибка при попытке записи измененной загрузочной записи на устройство жесткого диска. Если вторая попытка выполнения неудачна, это, вероятно, указывает на неверное определение физической устройства жесткого диска, или неустранимые ошибки носителя на первый цилиндре диска

## 6.10 HDIAG Форматирование жесткого диска/ Утилита диагностики

HDIAG является универсальной B/P служебной программой для выполнения диагностики, форматирования, проверки и изучения параметров жесткого диска, используя любые из определенных типов контроллеров в B/P BIOS, где такие возможности определены. Для дополнительной гибкости существует возможность выбора типа контроллера в начале программы, чтобы вы могли проверить и инициализировать диски с помощью других типов контроллеров, кроме определённого в выполняющемся BIOS. Следующие типы контроллеров обрабатываются в начальном выпуске B/P BIOS:

Adaptec ACB-4000A

Shugart 1610-3 / Xebec 1410A

Seagate SCSI

Shugart 1610-4 (Minimal SCSI)

Conner SCSI

Quantum SCSI

Maxtor SCSI)

(другие добавятся в более поздних выпусках)

Syquest SCSI

GIDE (Generic IDE)

### 6.10.1 Использование HDIAG

Эта утилита работает только в интерактивном режиме, поэтому она вызывается просто указанием своего именем без аргументов (помимо стандартного вызова справки двойной косой чертой). При активации, он считает тип контроллера из структуры заголовка B/P BIOS и спросит вас, тот ли это тип контроллера, который вы хотите использовать. Если вы хотите использовать другой типа контроллера, например диагностировать Seagate SCSI диск от системы, которая имеет контроллер Adaptec для нормального использования, вы можете изменить определение контроллера на оставшуюся сессию HDIAG. Взаимодействие через главный цикл приглашение может выглядеть как:

┌───────────────────────────────────────────────────────────────────────────┐

│ B/P Bios Hard Disk Utility V1.3a, 14 Jun 97 │

│ │

│ Controller = Adaptec Ok ([Y]/N) ? : Y │

│ │

│ Functions: F - Format │

│ V - Verify │

│ D - Run Diagnostics │

│ P - Show Disk Parameters │

│ │

│ Select (^C or ESC to Quit) : │

│ │

└───────────────────────────────────────────────────────────────────────────┘

#### 6.10.1.1 Показать параметры диска [ P ]

Если вы выполняете HDIAG с устройством жесткого диска, которое уже определено в BIOS и было ранее отформатировано, или с одним из самоопределяющихся дисков SCSI, вы можете просмотреть текущие параметры привода с помощью команды “P”. Отображение зависит от типа контроллера и количества и типа доступной информации. Некоторые данные могут быть из определений BIOS и других данных из любого контроллера (например, Adaptec) или электроники диска (SCSI или SCSI-2). Образцы форм информации:

Unit : 0 <-- CONNER CP3100-100mb-3.5 SCSI1 setting

Total Blocks = 204864 (12804 Eq. Tracks)

Sctrs/Track = 33

Sector Size = 512

Interleave = 1

# Cylinders = 776

Num of Heads = 8

Unit : 0 <-- Adaptec ACB-4000, Syquest SQ-312 10 MB

Total Blocks = 22140 (1383 Eq. Tracks)

Sctrs/Track = 18

Sector Size = 512

# Cylinders = 615

Num of Heads = 2

Reduced Wrt. = 615

Precomp. Cyl = 615

Step Rate = 12 uS Buffered

Media type = Removable

Landing Zone = 615

#### 6.10.1.2 Диагностика жесткого диска [ D ]

Некоторые диски оснащены подпрограммами встроенной диагностики, выполняющих проверку блока электроники и носителя. Других системы просто выполняют последовательность включения напряжения питания, которая обычно включает в себя последовательность самотестирования. Обычно, можно полагаться только на функцию повторной инициализации, которая входит в стандартный набор функций HDIAG с этой командой. Образец выходных данных, этой функции:

Select (^C or ESC to Quit) : D

Unit Number [0..2] (^C or ESC to Abort) : 0

Re-Initializing Unit : 0 ..Ok

..Waiting for Ready..

Паузы могут возникнуть при выполнении последовательности, наиболее заметно, после строки статуса, "Re-Initializing Unit" (повторная инициализация устройства) до появления "..Ok". В зависимости от конкретной системы, на этот раз часто, когда фактическая электроника контроллера проверяется, и может включать перемещение головки привода, которая может быть трудоемкой задачей. Очень часто также есть пауза после строки "..Waiting for Ready.." (Ожидание готовности), особенно если головки были перемещены и должны быть вновь установлены на внешний цилиндр диска. Когда диск возвращается в состояние готовности, то главное меню выбора отображается снова и HDIAG готов к другой команде.

#### 6.10.1.3 Проверка носителя диска

Эта функция позволяет оценить состояние отформатированного диска для выявления дефектов различной степени. При использовании сочетания адаптации с коррекцией кода ошибок, где это возможно, и включения или выключения проверки индивидуальных секторов, может быть получен сравнительно обширный, хотя зачастую отнимающий много времени, неразрушающий статус привода.

Select (^C or ESC to Quit) : V

Unit Number [0..2] (^C or ESC to Abort) : 0

Verify Individual Sectors (Y/[N]) : N

Verifying Unit : 0 CONNER CP3100-100mb-3.5

Block 891

...aborted...

#### 6.10.1.4 Формат диска [F]

Целью этой функции является заполнение всей области хранения данных на диске постоянной величиной, а также обновление управляющей информации на диске. Это разрушительно, и все данные на диске будут потеряны. Поэтому несколько проверок включены в программу, чтобы гарантировать, что вы непреднамеренно не активируете эту команду.

Хотя большая часть информации, необходимая для форматирования диска доступна из встроенных данных, которые могут быть считаны с диска или контроллера, или из области данных BIOS, некоторые элементы все еще требуются от пользователя. Поэтому вам будет предложено предоставить любые данные, необходимые для форматирования диска. В более старых системах SASI (1610-3, Xebec, и т.д.) это может означать значительное количество записей. К счастью форматирование дисков требуется не часто, и метод форматирования, используемый в HDIAG является достаточно гибким, чтобы обеспечить широкий спектр устройств для подключения.

### 6.10.2 Сообщения об ошибках HDIAG

Несколько сообщений об ошибках будут представлены для конкретных проблем, возникающих в работе HDIAG. Многие из сообщений характерны для определенных операций, другие изменяют конкретную информацию в зависимости от возможностей выбранного типа контроллера. Наиболее общими являются проблемы команд SCSI/SASI Sense. Новые системы SCSI используют "Extended Sense", который может возвращать больше информации, чем основное значение Sense. При обнаружении Extended Sense, многие сообщения об ошибках отображается значение "Key", а не основной байт "Sense". Консультируйтесь с руководством программирования конкретного контроллера или диск для определенных значений этих байтов. Обычно такое сообщение будет отображаться в следующем виде:

Error! (Comnd = xx) Sense: xx

или

Error! (Comnd = xx) Key = xx

Кроме того, чтобы предоставить дополнительную информацию во время выполнения многих функций, как часть сообщения об ошибке выводится raw (сырой) байт статуса, прочитанный из контроллера при возникновении ошибки:

(status = xxH)

Интерпретация представленного шестнадцатеричного байта, может быть получено из руководство по программированию для конкретного привода или тип контроллера.

+++ Not B/P Bios ... aborting +++

Была предпринята попытка выполнить HDIAG с BIOS отличным от B/P, или сделанные модификации в BIOS, изменили местоположение, или значения необходимые для корректного определения системы.

\*\*\*\* SCSI Block Length Error !

Это сообщение неустранимой ошибки будет отображаться, если из BIOS вернулся слишком маленький блок описания команды, чтобы позволить расширенные команды, необходимые для выполнения запрошенной операции. Это, как правило, является результатом изменений модуля жесткого диска, который изменил необходимые значения.

\*\*\*\* Controller Not Readable !

HDIAG не удалось прочитать параметры из привода или контроллера. Ошибка появляется только в функция “R” (Чтение параметров диска), когда тип контроллера “Owl”.

\*\*\*\* No Diagnostics for : <contyp>

Выбранный контроллер не может выполнить диагностику способом, с помощью которого HDIAG может получить доступ или выполнять необходимые функции. Ошибки будут появляться только в функции “D” (выполнить диагностику).

\*\*\*\* Verify Not Available !

Выбранный конкретные диск/контроллер не был определен надлежащим образом, чтобы позволить проверку диска. Ошибка появляется только в функции “V” (Проверка), когда тип контроллера “Owl”.

+++ 1610-3 Initialization Error...Sense = xxH

Была обнаружена ошибка, после посылки строки инициализации в контроллер Shugart 1610-3 или Xebec 1410A. Убедитесь, что в настройках конфигурации вашего оборудования указан правильный тип контроллера. Ошибка появляется только в функции “V” (Проверка).

## 6.11 INIRAMD Утилита инициализации виртуального диска

INIRAMD - это утилита B/P BIOS, которая инициализирует каталог виртуального диска и при необходимости подготавливает его для использования DateStamper**™** P2DOS, или обоих типов временных меток файлов. Она имеет функцию защиты, предотвращающую самопроизвольную инициализацию уже отформатированного виртуального диска, и может выполняться из командной строки для использования в сценарии запуска.

### 6.11.1 Использование INIRAMD

Эта утилита предназначена для использования с аргументами командной строки, но имеет встроенные значения по умолчанию, которые могут быть сконфигурированы изменением байтов в начале программы с новыми настройками по умолчанию, или путем настройки с помощью утилиты Al Hawley ZCNFG. Чтобы выполнить ее с параметрами по умолчанию, просто введите:

INIRAMD

Полный синтаксис INIRAMD следующий:

INIRAMD [d:][/][Q][D][P]

### 6.11.2 Режим командной строки

Заданием параметров командной строки, при вызове INIRAMD, нескольким внутренним значениям по умолчанию могут быть установлены значения необходимым в настоящее время. Первый ожидаемый параметр при разборе командной строки – буква диска. Он не является обязательным и будет переопределять привод “M:” встроенный в программу по умолчанию. Чтобы указать другой диск по умолчанию, введите:

INIRAMD d:

Отображение лишних сообщений приглашающих к вводу и о статусе могут быть отключены во время выполнения INIRAMD, присвоив встроенному в программу флагу по умолчанию значение "Quiet" (тихий), или введя “Q” в качестве параметра при вызове программы. Инициализация виртуального диска в тихом режиме, используя диск по умолчанию осуществляется путем ввода:

INIRAMD Q

Вы можете также указать, какие типы меток даты/времени, добавить в виртуальный диск при подготовке с помощью параметра “Р” (для меток P2DOS) и/или “D”. Если программа не работает в тихом режиме, INIRAMD уведомит вас, какие методы установки меток были добавлены в оперативную память компьютера после инициализации области каталога пустым значением. Инициализация диска “M:” для обоих типов меток выполняется путем ввода команды:

INIRAMD M: PD

### 6.11.3 Сообщения об ошибках INIRAMD

+++ Not B/P Bios ... aborting +++

Была предпринята попытка выполнить INIRAMD с BIOS отличным от B/P, или сделанные модификации в BIOS, изменили местоположение, или значения необходимые для корректного определения системы.

+++ Already Formatted...Proceed anyway (Y/[N]):

Это предупреждение и запрос будет выдан, если INIRAMD обнаружит фиктивное имя файла, используемого в качестве метки, означающей что виртуальный диск уже отформатирован. Оно наиболее часто появляется в системах, содержащих резервную батарею ОЗУ и INIRAMD вызывается либо непосредственно, либо с помощью псевдонима скрипта Startup. Чтобы свести к минимуму появление этого сообщения, если вы хотите, использовать инициализацию виртуального диска, в скрипте Startup включите следующее строчки:

IF ~EX M:-RAM.000 <-- Предположим, что RAM диск M:

INIRAMD M: <-- Предполагая, что RAM диск является М:

FI

+++ Drive d: does NOT Exist

По умолчанию или явно указанный при активации INIRAMD диск не был определен в системе. Одной из возможных причин является, что виртуальный диск была обменен на букву неопределенного диска. Проверьте диск назначения с помощью BPCNFG, опция 5, если вы хотите проверить назначения дисков.

+++ Drive d: is Not a RAM Drive!

По умолчанию или явно указанный при активации INIRAMD диск является реальным, а не виртуальным диском. Как и в случае с предыдущей ошибки, одна из причин заключается в том, что виртуальный диск была обменен на диск другого типа. Используйте BPCNFG, опция 5, для проверки назначения дисков.

## 6.12 INITDIR Утилита инициализации меток P2Dos-типа

INITDIR готовит диски для временных меток файлов P2DOS-типа. Она делает это путем замены каждой четвертой записи в каталоге диска на запись времени и даты, которая начинается со специального символа (шестнадцатеричное 21H). Существующие записи каталога в четвертой позиции, затем сдвигаются к первой записи в следующем логическом секторе, и инициализированные сектора каталога записываются обратно на диск.

**ПРЕДУПРЕЖДЕНИЕ**

Программа INITDIR **не должна выполняться для дисков, содержащих действующие DateStamper метки файлов**, поскольку она перестраивает данные каталога. Чтобы установить метки DateStamper и P2DOS на один диск, используйте пустой диск, или же не содержащего временные метки любого типа и запустите PUTDS и INITDIR для диска перед его использованием. При других действиях любые существующие данные временных меток будут недействительными.

### 6.12.1 Интерактивный режим INITDIR

INITDIR может работать и в интерактивном режиме, просто введя его имя в командной строке, или в “экспертном” режиме, указав букву диска в качестве параметра командной строки. В интерактивном режиме, вам будет предложено указать букву привода, которое укажет на диск для инициализации меток P2DOS.

Если DateStamper файл “!!!TIME&.DAT” обнаружен на диске, INITDIR выдает предупреждение и спрашивает, хотите ли вы продолжить или нет (см. [6.12.2](#_6.12.2_Сообщения_об) ниже). Во избежание потери информации времени и даты от DateStamper, эта процедура должна выполняться только на свеже-отформатированные или пустые диски. Кратко синтаксис INITDIR выглядит следующим образом:

INITDIR // <-- Печатать сообщение краткой справки

INITDIR <-- Выполните в интерактивном режиме

INITDIR d[:] <-- Инициализировать диск D:

### 6.12.2 Сообщения об ошибках INITDIR

Directory already initialized

Выбранный диск уже подготовлен для P2DOS меток.

Illegal drive name

Введенный символ не был в диапазоне от "A" до "P".

Not enough directory space on disk

Каталог на выбранном диске более чем три четверти полный, поэтому не достаточно места, для поддержки P2DOS временных меток файла.

Directory write error

Ошибка произошла при записи инициализированного каталога. Это, вероятно, приведет к потере данных файла.

--> DateStamper !!!TIME&.DAT File Found <--

Proceed anyway (Y/[N]) :

Специальный DateStamper файл “!!!TIME&.DAT” существует на диске. Если другие файлы будут также на диске, то большая часть информации о времени и дате DateStamper будет потеряна. На недавно отформатированных или пустых дисках не существуют данных с временными метками файла DateStamper, таким образом, безопасно ответить "Y" и инициализировать диск.

## 6.13 INSTAL12 Утилита поддержки загрузочных дорожек

INSTAL12 - установка CPR, ZSDOS, B/P BIOS из файлов образов MOVxSYS типа стандартного размера (2K CCP, 3.5K DOS, ~4.375K BIOS).

INSTAL12 это новейшая модификация ZSDOS утилиты INSTALOS, распространяемой с ZSDOS 1.0, которая автоматически накладывает файл на образ системы вашего компьютера, такой как программы MOVCPM.COM (CP/M) или MOVZSYS.COM (ZRDOS) или файл системы в абсолютном формате (например, CPM64.COM) с ZSDOS или ZDDOS, для получения нового файла, содержащего **ZSDOS**/**ZDDOS** вместо вашей исходной базовой дисковой операционной системы. INSTAL12 также позволяет вам устанавливать значения по умолчанию различным параметрам ZSDOS во время процесса установки (эти параметры могут также быть изменены позже программой ZSCONFIG).

Программа INSTAL12 разработана, чтобы сделать процесс установки максимально простым. С INSTAL12 вы можете загрузить файлы со всех дисков и пользовательских областей от “A0:” до “P31:”. Обнаружение ошибок обширно, и [Раздел 6.13.2](#_6.13.2_Сообщения_об) этого руководства подробно объясняет все сообщения об ошибках INSTAL12. Наконец, вы можете безопасно прервать выполнение INSTAL12 почти в любом месте нажатием Ctrl-C.

Перед использованием INSTAL12, убедитесь, что следующие файлы из вашего дистрибутивного диска B/P BIOS и/или ZSDOS присутствуют:

* Скомпилированный B/P BIOS в формате Microsoft .REL (MOVCPM = YES)
* ZSDOS.ZRL (если заменяется операционная система)
* ZCPR33.REL, ZCPR34.ZRL или другой процессор команд (если заменяется CPR)
* INSTAL12.COM

Следующие файлы из вашего дистрибутивного диска B/P BIOS, CP/M или системного диска ZRDOS должны также быть доступны:

* MOVxSYS.COM (B/P BIOS), MOVCPM.COM (CP/M), MOVZSYS.COM (ZRDOS), или файл образа системы (для таких систем, как Oneac ON!)

### 6.13.1 Использование INSTAL12

Для запуска INSTAL12, большинству пользователей нужно просто ввести

INSTAL12

в командной строке. Это говорит INSTAL12, что вы устанавливаете сегмент системы на перемещаемый файл образа системы, например MOVxSYS.COM, MOVCPM.COM или MOVZSYS.COM. Если вам нужно установить сегмент поверх файла системы в абсолютном формате, например, CPM59.COM, ZSYSTEM.MDL, или на файл Oneac!, вы должны ввести

INSTAL12 /A

для запуска INSTAL12 в абсолютном режиме. INSTAL12 теперь отображает свой баннер открытия и просит ввести имя файла:

System Image file to patch (Default=MOVCPM.COM) :

в перемещаемые режиме или

Absolute System Model (Default=SYSTEM.MDL) :

в абсолютном режиме.

Вам не нужно вводить всю информацию. INSTAL12 заполнит недостающие элементы диск, пользователь или имя файла по умолчанию. Если вы просто нажмете Return, INSTAL12 по умолчанию ищет в текущем каталоге образ системы или файл системы в абсолютной формате (MOVxSYS.COM, MOVCPM.COM или SYSTEM.MDL). Ниже приведены некоторые примеры ответов:

System Image file to patch (Default=MOVCPM.COM) : B3:

(Выбор MOVCPM.COM на диске "B" в пользовательской области 3)

System Image file to patch (Default=MOVCPM.COM) : 10:MOVYSYS

(Выбор MOVYSYS.COM на текущем диске, пользователь 10)

System Image file to patch (Default=MOVCPM.COM) : C:MOV18SYS.OLD

(Выбор MOV18SYS.OLD на диске "C", в текущей области пользователя)

Как только INSTAL12 находит требуемый файл, он проверяет ваш образ операционной системы. Если CCP, BDOS или участки BIOS образа системы или файл системы в абсолютном режиме являются недопустимыми, INSTAL12 печатает сообщение об ошибке и завершает работу в этом месте. Это может произойти, если был загружен образ абсолютной системы, но INSTAL12 был вызван без параметра “/A”. Если оба способа вызова INSTAL12 неудачны, сначала убедитесь, что ваш образ системы, или программа генерации работает должным образом. Если вы уверены, что у вас есть рабочий файл MOVxSYS, MOVCPM, MOVZSYS, или абсолютной модели, который INSTAL12 не может проверить, вам будет нужно, обратитесь к дистрибьютору, который будет инициировать действия по устранению проблемы.

Если все значения в файле операционной системы соответствуют ожидаемым параметрам, отображается краткое описание этих значений. Если указан файл образа системы (например, MOVxSYS.COM), дисплей должен быть похож на:

Адреса в образе системы (как показывает DDT):

CCP : 0980H Map @ 3610H

BDOS: 1180H Map @ 3710H

BIOS: 1F80H Map @ 38D0H

Показанные адреса вероятно будут отличаться от этих, но если оба столбца отображают значения, отличные от 0000H, INSTAL12 правильно наложит три части образа системного сегмента с указанными файлами.

Если вы определили систему в абсолютном формате, дисплей будет подобен:

Адреса в образе системы (как показывает DDT):

CCP : BC00H

BDOS: C400H

BIOS: D200H

адреса будут вероятно отличаются от примера приведенного выше, который предназначен для системы 54K.

Если не появляется сообщение об ошибке, INSTAL12 считает ваш файл правильным. Далее появляется меню выбора:

1 - Replace CCP (Замена CCP)

2 - Replace DOS (Замена DOS)

3 - Replace BIOS (Замена BIOS)

4 - Save and Exit (Сохранить и выйти)

Enter Selection (^C Quits) : \_ (Введите выбор (^C выход))

Чтобы установить новое изображение B/P BIOS их вашего ассемблирование, выберите опцию “3”. Вас попросят ввести имя файла:

Name of BIOS file (Default=CBIOS.REL) : \_

Тип файла по умолчанию - REL, и файл должен быть в формате Microsoft REL. Когда вводится завершающий возврат каретки, будет использовано имя которое вы ввели или имя по умолчанию. Если найдено, то будет оцениваться размер доступного места в файле изображения. Часто при добавлении B/P BIOS на более старую программы генерации системы, истовая карта части программы будут перемещена. Вы будете уведомлены сообщением, означающий расстояние в байтах, на которое карта была перемещена. Это просто диагностический инструмент, и вы не должны быть встревожены сообщением. После сообщения “...overlaying BIOS...” (…наложение BIOS...), INSTAL12 вернется в главное меню для следующей команды.

Замена процессора команд с ZCPR33.REL или ZCPR34.ZRL идентична замене BIOS, за исключением того, что перемещение битового массива не возможно. Указанный файл или будет соответствовать и наложен на оригинал, или он будет слишком большим, и программа выйдет с сообщением об ошибке по этой причине.

Для установки ZSDOS введите “2”. Вам будет предложено ввести имя файла дисковой операционной системы:

Name of DOS file (Default=ZSDOS.ZRL) : \_

расширение имени файла по умолчанию на данный момент является ZRL, но операционные системы в формате Microsoft REL, такие как версии распределения ZRDOS также допустимы. Как указано выше, Вы можете указать полную или частичную спецификацию файла, и INSTAL12 заполнит любые недостающие элементы по умолчанию диск, пользователя или имя файла.

После того, как файл дисковой операционной системы найден отображается следующий запрос:

ZSDOS.ZRL Size OK...overlaying BDOS..

Examine/Change ZSDOS parameters ([Y]/N)? : \_

На этом этапе, INSTAL12 позволяет изменять настройки запуска всех опций ZSDOS. Если это ваша первая установка ZSDOS, мы рекомендуем вам нажать “N” для “Нет”, чтобы пропустить этот шаг, и пропустить следующий абзац.

Если ввести любой символ кроме “N” или “n”, подразумевается вариант по умолчанию в квадратных скобках ([Y] для “Да”), и INSTAL12 отобразит текущие значения **ZSDOS** по умолчанию:

|  |  |
| --- | --- |
| 1 - Public Files | : YES |
| 2 - Pub/Path Write Enable | : NO |
| 3 - Read-Only Vector | : YES |
| 4 - Fast Fixed Disk Log | : YES |
| 5 - Disk Change Warning | : NO |
| 6 - Path w/o System Attr | : YES |
| 7 - DOS Search Path | : Disabled |
| 8 - Wheel Byte Protect | : Disabled..Assumed ON |
| T - Time Routine (Clock) | : Disabled |
| A - Stamp Last Access Time | : Disabled |
| C - Stamp Create Time | : Disabled |
| M - Stamp Modify Time | : Disabled |
| G - Get Date/Time Stamp | : Disabled |
| S - Set Date/Time Stamp | : Disabled |
| Entry to Change ("X" to EXIT) | : \_ |

Эти варианты представлены в том же порядке, как и в ZSCONFIG и подробно описаны в *Разделе 4.10 руководства ZSDOS 1.0*.

После того, как вы пропустите шаг настройки или выйдете из него, нажав “X”, в зависимости от того, устанавливается ли файл изображения или абсолютного формата появляется одно из следующих сообщений:

Name to save new system (Default=MOVZSDOS.COM) : \_

или

Name to save new system (Default=ZSSYS.MDL) : \_

Снова вы можете задать полное или частично определенное имя файла и INSTAL12 заполнит любые отсутствующие элементы по умолчанию: диск, пользователя или имя файла. Если файл с таким же именем существует, INSTAL12 предложит ввести новое имя. Когда INSTAL12 получит допустимое имя, он создает новый системный файл и завершает работу, одним из следующих сообщений:

..Saving MOVZSDOS.COM - перемещаемый

или

..Saving ZSSYS.MDL - абсолютный

### 6.13.2 Сообщения об ошибках INSTAL12

Иногда INSTAL12 может выдавать сообщения об ошибках. Большинство ошибок возникает в результате, если указанные вами файлы не соответствуют ожиданиям INSTAL12. Часто выходом является запуск INSTAL12 снова, выбрав перемещаемый режим вместо абсолютного режима или наоборот. Многие ошибки INSTAL12 возникает из-за поврежденных файлов. Если INSTAL12 выдает ошибки как в абсолютном, так и перемещаемом режимах, Попробуйте, скопировав исходный файл от мастеров, или перекомпилируйте исходную программу и повторно выполните INSTAL12.

Если все указанное выше не поможет, системные файлы могут содержать информацию, которую INSTAL12 не может распознать. Вы можете попытаться выполнить альтернативные установки с NZCOM или JetLDR для CPR и DOS сегментов, но может потребоваться связаться с экспертами с Ladera Z-Node для помощи с проблемами, связанными с BIOS.

Ниже приводится краткое описание всех сообщений об ошибках INSTAL12, их значений, и некоторые возможные способы устранения.

\*\*\* SORRY! ZSDOS will only run on Z80 type computers!

ZSDOS и ее утилиты будут работать только на процессорах выполняющих инструкции Z80 таких как Z80, NSC-800, Z180 или HD64180. Не существует способа исправления этого условия, кроме как запустить ее на другой системе.

\*\*\* Unable to open [filename.typ]

INSTAL12 не удается найти или открыть системный файл, указанный вами. Во-первых убедитесь, что файл находится в расположении по умолчанию или на указанном диске/области пользователя. Если вы определили файл правильно, но эта ошибка сохраняется, получите новую копию своего системного файла и попробуйте еще раз.

\*\*\* Can't find CCP/BDOS/BIOS at standard locations !!!

Операционная система, содержавшаяся в вашем системном файле, не является стандартной системой CP/M. Она содержит CCP, длина которого не точно 2 килобайта, BDOS длина, которой не точно 3.5 килобайта, или обеих. Если это сообщение появляется, сначала убедитесь, что ваш системный файл не был поврежден. Если вы по-прежнему получаете это сообщение, свяжитесь с авторами на Ladera Z-Node или вашим дистрибьютором.

++ Image Vector does not match Calculations ++

INSTAL12 нашел внутреннюю ошибку в файле изображения при установке файла MOVCPM-типа. Если вы не использовали опцию /A при выполнении INSTAL12, вы возможно пытаетесь выполнить относительную установку с абсолютным файлом. Попытайтесь выполнить INSTAL12 снова с помощью команды INSTAL12 /A.

\*\*\* Cannot find legal Relocation Bit Map

INSTAL12 не смог определить расположение допустимого образца битового массива перемещения в файле MOVCPM-типа при установке в перемещаемом режиме. Нестандартные файлы перемещаемых образов являются общей причиной этой ошибки. Чтобы обойти проблему нужно, во-первых, создать абсолютную формат с помощью MOVCPM, затем использовать INSTAL12 в абсолютном режиме (/A) с файлом в абсолютном формате.

---Can't find [filename.typ].. reenter (Y/[N]) :

Указанный файл замены (CCP, BIOS или BDOS) не найден. Убедитесь, что диск, пользователь и имя файла указаны правильно.

\*\*\* Error in .REL sizing [filename.typ]

Err Code : nn

Произошла ошибка INSTAL12 во время назначения размеров REL или ZRL файла. Файлы с расширением REL или ZRL должны быть в перемещаемом формате Microsoft. Именованные сегменты Common, кроме \_CCP\_, \_BDOS\_, и \_BIOS\_ не допустимы, сегменты кода и сегменты данных (если такие имеются) не должны накладываться.

\*\*\* file too large to fit...

Размер перемещаемых CCP или BDOS превышает доступное пространство в файле образа (2048 байт для CCP, 3584 байт для BDOS). Эта ошибка может возникнуть если перемещаемый файл не в нужном формате Microsoft REL, или если используется настроенный файл. Эта ошибка никогда не должно возникать с дистрибутивом ZSDOS. Файл ZRL, которого имеет длину ровно 3584 байт (3,5 k).

\*\*\* Error opening : [filename.typ]

INSTAL12 не смог открыть указанный перемещаемый файл. Убедитесь, что Вы выбрали допустимый файл REL.

\*\*\* Error reading : [filename.typ]

INSTAL12 обнаружил ошибку при чтении указанного перемещаемого файла. Попытайтесь перекопировать файл.

\*\*\* Error in .REL file : nn

В перемещаемом входном файла была обнаружена ошибка при попытке заменить CCP, BDOS или часть BIOS вашей операционной системы. nn это шестнадцатеричный код, который может помочь в обнаружении причины ошибки. Если вам нужна помощь в разрешении ошибки такого рода с кодом в сообщении об ошибке, обратитесь к дистрибьютору.

--- That file already exists. Overwrite it (Y/[N])?

Файл, в который вы сказали INSTAL12 производить запись уже, существует. Если вы введете здесь “Y”, то INSTAL12 сотрет предыдущую копию и создаст новый файл с этим именем. Введите “N”, чтобы выбрать новое имя.

\*\*\* No Directory Space for [filename.typ]

На выбранном диске в каталоге недостаточно места для выходного файла. Отправьте выходной файл на другой диск, поставив в начале имени файла описатель диска, или замените диск в выходном дисководе.

\*\*\* Error writing file. Try again with another disk (Y/[N])? :

Это сообщение обычно возникает из-за нехватки места на диске, указанном для вывода. Смените диск и введите “Y”, чтобы попробовать снова.

## 6.14 IOPINIT Утилита инициализации пакета IO

IOPINIT инициализирует буфер IOP, определенный в дескрипторе окружения в формате стандартного пустого IOP и исправляет его в таблице переходов BIOS. Он выполняет ту же основную функцию, что и старый ZCPR3 способ загрузки файла SYS.IOP, но был добавлен в качестве самостоятельной процедуры, чтобы сделать, по сути, всю установку пакета. При этом, дополнительное пространство было освобожден в коде ядра В/Р BIOS, позволяя добавить другие процедуры, которые не могут быть удалены для внешних программ.

### 6.14.1 Использование IOPINIT

Эта программа должна быть включена практически в самом начале сценария STARTUP для любой системы, в которой определяется IOP. **Примечание:** Эта процедура должна быть запущена перед любыми программами, которые изменяют переход теплого старта в расположении 0!

При вызове IOPINIT не ожидает никаких параметров со всеми значениями, определенными из исполняющейся среды системы. Процедура отвечает на обычный запрос справки двойной косой черты как и все процедуры поддержки.

### 6.14.2 Сообщения об ошибках IOPINIT

--- No IOP Buffer defined in Environment ---

Не требует пояснений. Если буфер IOP был сознательно удален во время конфигурации или компиляции, не будет причинено никакого вреда путем выполнения IOPINIT.

--- No Z-System Environment Defined ---

Это сообщение никогда не должно появляться, поскольку допустимый дескриптор среды требуется системам оснащенным В/Р BIOS. Если это происходит, то возможной причиной является неправильное значение(я) в критических точках, в которых дескриптор используется для проверки среды.

\*\*\* IOP Already Installed! \*\*\*

Не требует пояснений. Никакого вреда системе, это сообщение просто для информации.

## 6.15 LDSYS Загрузчик файла образа системы

LDSYS является основной утилитой для активации файла образа системы, подготовленного BPBUILD (см. [6.1](#_6.1_BPBUILD_-_5)). Она сначала проверяет в настоящее время работающую систему, а затем загружает файл образа, помещает составные части, предназначенные для них места в памяти компьютера и выполняет BIOS холодный старт недавно загруженной системы. Файлы изображений могут быть либо с переключением банков памяти или без переключения и не должны помещаться в зарегистрированном каталоге, так как LDSYS может получить доступ к файлам по пути системы, или из спецификаций пути Z3-стиля.

### 6.15.1 Использование LDSYS

Эта утилита предоставляет единственный способ установки системы B/P системы с переключением банков памяти и простой способ проверки систем без переключения банков перед заключительным преобразованием в загружаемые системы, которые будут помещены на системные дорожки, используя INSTAL12, MOVxSYS и BPSYSGEN. LDSYS ожидает, только единственный параметр передаваемый в командной строке, являющийся названием образа файла для загрузки. Если расширение файла явно не указано, предполагается .IMG. Местоположение искомого файла может быть явно указано обычным способом ZCPR3 с любыми префиксами DU: или DIR:. Полный синтаксис LDSYS следующий:

LDSYS [du|dir:]name[.typ]

При загрузке, отображаются два кратких экрана, первый из LDSYS и второй из процедуры холодного старта загруженной системы после передачи управления от LDSYS недавно загруженной системе. Пример отображения из системы с переключением банков памяти во время разработки при установке на MicroMint SB-180:

┌───────────────────────────────────────────────────────────────────────────┐

│ B/P Bios System Loader Vers 1.0 31 Aug 92 │

│ Copyright (C) 1991 by H.F.Bower & C.W>Cotrill │

│ │

│ CCP starts at : CC80 (0F80H Bytes) │

│ DOS starts at : DC00 (0F00H Bytes) │

│ Banked Dos at : 1080 (0500H Bytes) │

│ BIOS starts at : EB00 (0880H Bytes) │

│ Banked Bios at : 1580 (1269H Bytes) │

│ │

│ ...installing Banked System │

│ │

│ SB180 B/P 60.25k Bios Ver 0.6 26 Jan 92 (Banked) with: │

│ │

│ ZCPR3+ Env │

│ ZSDOS Clock │

│ Hard Disk Support │

│ Warm Boot from RAM │

│ RAM Disk (M:) │

│ Full Error Messages │

│ \_ │

└───────────────────────────────────────────────────────────────────────────┘

Все сообщения в приведенном выше образце экрана до строки "...installing Banked System" напечатаны LDSYS. Все последующие строки в вышеупомянутом экране отображаются из недавно загруженного BIOS. Во время изменения или модификации новой системы, это подразделение в области отображения может быть ключом к разгадке любых возникающих трудностей. Положение курсора у основания типового экрана - пункт, в котором новый BIOS, уже работающий, пытается загрузить Startup файл, определенный в заголовке B/P. Если ничего не найдено, то дополнительная инициализация не будет выполнена, и вы увидите только приглашение для диска A, пользователя 0.

### 6.15.2 Сообщения об ошибках LDSYS

\*\*\* No file specified ! \*\*\*

LDSYS был вызван без указания файла для загрузки. Вы можете повторно вызвать его непосредственно из ZCPR командой "GO":

GO filename

или вызвать его в нормальном режиме, указав файл образа для загрузки.

--- Ambiguous File: fn[.ft]

По крайней мере один вопросительный знак (или звездочка расширения) был обнаружен в указанном файле, для загрузки. Повторно выполните с однозначным именем файла.

--- Error Opening: fn[.ft]

Не найден указанный файл образа. Распространенными причинами могут являются несоответствие в имени файла, отсутствие файла с расширением по умолчанию “IMG” или невозможность найти файл по пути DOS или в общедоступном каталоге.

## 6.16 MOVxSYS - Утилита загрузочных дорожек системы

Эта процедура представляет собой программу для создания загрузочного образа системы для загрузочных дорожек дискеты или жесткого диска. Она настраивается для каждого типа аппаратной системы, используя B/P BIOS. Общее имя MOVxSYS переводится в определенное имя, отражающее ID стандартного имени, примерами которых являются:

MOVYSYS YASBEC

MOVAMSYS Ampro Little Board

MOV18SYS MicroMint SB-180

### 6.16.1 Использование MOVxSYS

Эта программа построена по образцу оригинальной MOVSYS.COM распространяемой с большинством систем CP/M 2.2 Digital Research, но активно обновляться, чтобы отражать Z-System стандартную справку, запись базового (начального) адреса или размер системы в килобайтах, и необходимые проверки на соответствие стандартам B/P.

Два основных параметра могут передаваться в эту программу как параметры в командной строке. Первая определяет размер системы либо эквивалентное количество килобайт в эквивалентной системы CP/M 2.2, или базовый (начальный) адрес процессора команд. MOVxSYS анализирует первый элемент, чтобы определить, является ли введенное значение шестнадцатеричным числом оканчивающимся буквой “H” и предполагает, что значение указывает начальный адрес, если это так. Допустимые адреса должны быть больше, чем 8000H, чтобы гарантировать, что резидентная часть операционной системы будет в общей области памяти. Если параметр не является шестнадцатеричным адресом, то предполагается, что это количество килобайт в диапазоне от 39 до 63. Эти размеры основаны на "стандартных" элементах CP/M, составляющих 2 Кбайт для процессора команд, 3.5 Кбайт памяти для базовой дисковой операционной системы, и как минимум 1,5 Кбайт для резидентной части BIOS. Выполняется несколько проверок в процессе выполнения MOVxSYS и начальной части BIOS (холодного старта) для определения неправильного местоположение и некорректные размеры области данных.

Второй параметр, который может быть определен в командной строке, является дополнительной звездочкой ("\*") или именем и расширением файла после спецификации размера. Если параметр присутствует, то будет принято одно из двух действий. Звездочка дает программе команду перемещать образ системы к указанному размеру, и просто сохранять его в памяти после выхода, не сохраняя образ на диск. Любые другие символы определят имя файла, под которым образ должен быть записан в диск. Если второй параметр не задан, то образ будет записан под именем файла SYSnnK.BIN, где “nn” число определяющего размер системы в килобайтах, описанной выше, округленное в меньшую сторону до самого близкого четного числа.

Размещение образа системы на загрузочные дорожки диска осуществляется BPSYSGEN (см. [6.6](#_6.6_BPSYSGEN_-_4)), которое может быть сделано сразу же после MOVxSYS (вызывается с аргументом звездочка), BPSYSGEN с помощью выбора пункта «Image in Memory», или указав выходной файла из MOVxSYS и вызвав BPSYSGEN с именем результирующего файла.

Если вы перенастраиваете BIOS для вашей системы с целью изменения системы загрузки, вы должны скомпилировать B/P BIOS в режиме без переключения банков памяти с параметром BANKED приравненным к NO, и установив приравнивание MOVCPM к YES в файле DEF-xx.LIB (см. [4.2](#_4.2_Установка_образа_3)). Пересмотренный BIOS затем может быть добавлен к MOVxSYS.COM с INSTAL12 (см. [6.13](#_6.13_INSTAL12_Утилита_1)), для получения модифицированной загрузочной системы, отражающей ваши потребности. Обратитесь к [Главе 3](#_Адаптация_B/P_BIOS) для изучения более полного описания процесса установки.

### 6.16.2 Сообщения об ошибках MOVxSYS

\*\*\*\* Start < 8000H !!!

Значение размера или начальный адрес CPR, указанный в результате, в качестве начального адреса, менее 8000H. Поскольку нижние 32K памяти (0..7FFFH) могут быть переключены, CPR должен быть в верхней половине памяти. Запустите программу снова с скорректированной спецификацией размера.

\*\*\*\* Create Error !

Программа не может создать указанный двоичный выходной файл. Возможные причины полный каталог, дискеты с защитой от записи, или плохой носитель или диск.

\*\*\*\* Write Error...Exiting

Произошла ошибка при записи заданного двоичного выходного файла. Возможные причины являются ошибка носителя или диска, или диск с недостаточным местом для хранения файла.

\*\*\*\* Close Error !

Произошла ошибка при попытке закрыть указанный двоичный выходной файл. Это обычно происходит из-за ошибки носителя или диска.

\*\*\*\* Size must be in 39..63 !!

MOVxSYS был вызван с недопустимой спецификацией размера (число K). Еще раз выполните программу с исправленной спецификацией размера.

\*\*\*\* Bad Syntax !!

Программа запуталась и не смогла решить, что вы хотели сделать. Просмотрите встроенную справку, введя:

MOVxSYS //

и соблюдайте перечисленный синтаксис.

## 6.17 PARK - Утилита парковки головок жесткого диска

PARK - простая B/P утилита, которая перемещает головки жесткого диска на всех дисках в назначенную область посадки (также называемую зоной доставки или парковки), если она определена для типа контроллера/диска в вашей системе. Когда все диски припаркованы, утилита выполняет инструкцию HALT, которая требует, аппаратный сброс или последовательное выключение/включение питания для ее преодоления. Чтобы избежать возможность повреждения жесткого диска при отключении питания дисков, в то время как головки расположены над зоной хранения данных на жестких дисках, утилита PARK всегда должна быть выполнена до выключения компьютера. Эта рекомендация особенно важна для дисков, которые не имеют функции автоматической парковки, или где эти аппаратные возможности потерпели неудачу.

### 6.17.1 Использование PARK

Эта утилита просто вызывается без параметров, и последовательно сканирует все три возможных устройства жестких дисков, выполняя команду SCSI "Stop Unit" на каждом. Когда все три устройства обработаны, процессор отключает прерывания и выполняет инструкцию HALT, чтобы препятствовать повторной активации устройств последующими инструкциями. Обычно, только повторное включение питания или нажатие кнопки сброса на компьютере позволят возобновить работу. Ставшая привычкой выполнение HALT, прежде чем включить компьютер может привести к увеличению срока службы жестких дисков, и должна стать обязательной.

Эта утилита - специализированная версия SPINUP (см. [6.20](#_6.20_SPINUP_Утилита)), которая разрешает отдельным устройствам быть выключенными и включенными во время нормального функционирования.

### 6.17.2 Сообщения об ошибках PARK

+++ Not B/P Bios ... aborting +++

Была предпринята попытка выполнить PARK с BIOS отличным от B/P, или сделанные модификации в BIOS, изменили местоположение, или значения необходимые для корректного определения системы.

+++ Can't Park this type of Controller! +++

PARK выполнялся с типом контроллера или диска в BIOS, которого не реализует функцию SCSI “Stop Unit”.

\*\*\*\* SCSI Block Length Error !

BIOS не поддерживает дополнительные команды, необходимые для парковки головок с помощью функции SCSI “Stop Unit”. Она вероятней всего появляется из-за изменений BIOS во время редактирования/компиляции, которые изменили размер команды дескриптора блока или функцию жесткого диска, которая возвращает значения.

## 6.18 SETCLOK утилита установки часов реального времени

Эта утилита обеспечивает средство установки часов B/P BIOS из физических часов, содержащихся в библиотеке ZSDOS CLOCKS.DAT. Она предоставляет интерфейс подобный утилите ZSDOS 1 TESTCLOK, из которой она была получена.

### 6.18.1 Использование SETCLOK

SETCLOK вызывается, вводом своего имени с необязательным номером драйвера часов. Для первоначального тестирования, или тестирования разных часов (всегда опасно процедура) просто введите имя утилиты:

SETCLOK

Вам будет предложено извлечь часы из библиотеки. Если вы используете пользовательские часы, ответьте “N” Если вы хотите использовать часы из подготовленной библиотеки ZSDOS, введите “Y”, которая является значением по умолчанию, если вводится возврат каретки. Вас также могут спросить расположение (диск/пользователь) файла часов или библиотеки. Эта последовательности запросов может выглядеть как:

Extract Clock from Library ([Y]/N) : \_

Location of CLOCKS.DAT [B0:] : \_

Диск B, пользователь 0 показанные в приведенном выше примере будут, вероятно, отличаться от текущего диска и пользователя в вашей системе, всегда отображаемых по умолчанию. Если файл находится на зарегистрированном в данный момент диске и является общим, он также будет найден без указания уникальной области пользователя. Если значение по умолчанию отражает расположение файл CLOCKS.DAT, или расположение, доступно через функции общего доступа, просто введите возврат каретки, в противном случае введите расположение CLOCKS.DAT, сопровождаемое двоеточием и возвратом каретки. Появится список состоящий из более чем 40 доступных часов. Чтобы выбрать один из этих драйверов, введите число, соответствующее часам, и программа сделает все остальное. Будут отображаться различные сообщения во время загрузки, подключения и выполнения драйвер часов. Если все пойдет хорошо, то последнее сообщение будет последовательно сообщать дату и времени читаемые из часов, означает что часы B/P BIOS были установлены правильно.

Кроме того, драйвер часов может быть выбран для автоматического исполнения в соответствии со спецификацией в командной строке. Чтобы использовать этот способ, файл CLOCKS.DAT должен находиться на зарегистрированном в настоящее время диске и области пользователя, или на текущем диске с установленным битом атрибута общий. Синтаксис этого способа установки часов B/P BIOS следующий:

SETCLOK nn

где “nn” - это число, соответствующее одному из часы в файле CLOCKS.DAT. Этот способ может быть использован для установки часов BIOS с помощью алиаса скрипта, например, STARTUP.COM который обычно используется, при первоначальной загрузке компьютера.

### 6.18.2 Сообщения об ошибках SETCLOK

Некоторые ошибки в утилите SETCLOK создаются программой верхнего уровня. Эти ошибки следующие:

+++ This is only for Z80 type computers!!!

Эта подпрограмма будет работать только с Z80 или совместимыми типами процессоров, так как это утилита B/P BIOS, который тоже ограничен этими типами.

-- Error in locating Clock file

Эта процедура не может найти библиотеку CLOCKS.DAT. Убедитесь, что библиотека существует в зарегистрированном в настоящее время каталоге, или может быть найдена с помощью функции общего доступа, или по пути DOS.

Некоторые ошибки появляются в процессе извлечения и проверки выбранных драйверов, из библиотеки CLOCKS.DAT. Такими ошибками являются:

-- Error Opening : clockname

Не удается открыть выбранный драйвер часов в библиотеке. Она появляется чаще всего из-за повреждения файла CLOCKS.DAT. Восстановить его с дисков резервной копии ZSDOS и повторите попытку.

-- Error Reading : fn.ft

Произошла ошибка при чтении код часов из библиотеки. Также появляется из-за повреждения файла CLOCKS.DAT.

-- Error in : clockname

Произошла ошибка в логической перемещаемой структуре выбранного драйвера часов.

-- Error initializing DAT file

Произошла ошибка при инициализации библиотеки CLOCKS.DAT.

-- Memory overflow in DAT file

Произошла ошибка выделения памяти в подпрограмме часов, которая привела к превышению выделенной памяти. Это не должно происходить с любым драйверов часов из библиотеки, но может наблюдаться, если не соблюдаются рекомендации при разработке пользовательских часов.

Еще некоторые ошибки касаются чтения и подключения выбранной подпрограммы часов, из библиотеки часов, или загружаемой как автономный драйвер. Этими ошибками являются:

+++ Can't find : CLOCKS.DAT

SETCLOK не смог найти файл часов по ссылке. Эта ошибка будет появляться, если делается попытка использовать автономный драйвер часов, который не удалось найти на текущем диске/пользователя, с помощью атрибута общий, или по пути DOS.

+++ Error on file open

Произошла ошибка при попытке открытия файла автономных часов. Убедиться, что файл был правильно скомпилирован и попробуйте снова.

+++ Error sizing : fn.ft

Выбранный файл часов содержит информацию ошибочного или недопустимого размера. Если она выводится при загрузке файла CLOCKS.DAT, перепишите файл с диска с резервной копии ZSDOS и повторите попытку. Если она появляется при загрузке автономных часов, это чаще всего из-за неправильной спецификации CSEG/DSEG/именованных общих областей в шаблоне часов. Убедитесь, что спецификации часов соблюдаются, установите драйвер и повторите попытку.

+++ Link Error : nn in file : fn.ft

Произошла ошибка при линковке перемещаемого кода драйвера часов. В поле “nn” сообщается индикатор точной причины ошибки. Обратитесь к авторам, если не можете устранить ошибку.

Последние две ошибки являются индикатором, возникновения ошибок после загрузки, сборки и проверки драйвера часов. Если одна из них появляется, это происходит чаще всего из-за неправильного выбора драйвера часов, проблем с оборудованием, управляющим выбранными часами или изменениями в коде B/P BIOS, которые изменили указанный интерфейс.

-- Clock Not Ticking --

Выбранный драйвер часов не смог обнаружить активные часы. Это - чаще всего результат выбора неправильного драйвера или задание некорректных значений, при запросе конкретных адресов или значений, при активации.

-- Error Setting B/P Bios Clock !!

BIOS сообщил об ошибке при попытке установить часы B/P BIOS. Она чаще всего вызвана ошибками при изменении модуля CLK.Z80.

## 6.19 SHOWHD утилита просмотра раздела

SHOWHD - утилита, которая предоставлена пакетом B/P BIOS как средство в преобразовании существующих систем к B/P BIOS, без потери данных, особенно на жестких дисках. Она предназначена не только для B/P BIOS и должна корректно выполниться в любой системе, совместимой с CP/M 2.2. Ее цель состоит в том, чтобы вывести на экран текущие настройки параметров раздела жесткого диска так, чтобы Вы могли сконфигурировать B/P BIOS или в исходном коде или в виде образа (с помощью BPCNFG), чтобы отразить те же данные разбиения.

### 6.19.1 Использование SHOWHD

Эта процедура является основной утилитой, обычно редко используемой, поэтому излишества не были добавлены. Она не имеет параметров и работает только в интерактивном режиме. Чтобы выполнить ее, достаточно ввести:

SHOWHD

Вам будет предложено ввести букву диска. После ввода, вам будет представлен список логических параметров диска. Пример исполнения:

┌───────────────────────────────────────────────────────────────────────────┐

│ Show Hard Drive Partition Data - 2 Nov 91 │

│ │

│ Enter Drive Letter [A..P] : C │

│ │

│ Drive: C │

│ DPH Info BPCNFG Info │

│ │

│ │

│ Sectors/Track = 64 (same) │

│ Blk Shift Fctr = 5 4k/Block │

│ Block Mask = 31 │

│ Extent Mask = 1 │

│ Disk Blocks-1 = 4999 20000k Total (2500 Tracks) │

│ Max Dirs - 1 = 1023 1024 Dir Entries │

│ Alloc bytes = FFH, 00H │

│ Check Size = 0 │

│ Track Offset = 3500 (same) │

│ \_ │

└───────────────────────────────────────────────────────────────────────────┘

### 6.19.2 Сообщение об ошибке SHOWHD

Только одно сообщение может отображаться в утилите:

+++ Invalid Drive : d

Выбранная буква диска не является допустимым диском в BIOS.

## 6.20 SPINUP Утилита управления двигателем жесткого диска

SPINUP -это универсальная B/P утилита для прямого управления головками и моторами новых SCSI-дисков. Она перемещает головки на указанном устройстве жесткого диска на место доставки или зону парковки и может выключить мотор в устройстве диска если вызвана для остановки устройства, и функция существует в приводе. Если вызвана, для запуска устройства, мотор диска включается (если возможно) для указанного устройства диска и головки перемещаются к цилиндру 0. Эта подпрограмма может быть использована в качестве функция экономии электроэнергии, когда работа может быть продолжена в течение длительного периода времени с виртуальным или флоппи-диском без необходимости доступа к устройству жесткого диска. Попытки доступа к устройству, который “замедлил вращение” (spun down) с помощью SPINUP приведет к ошибке. Эта подпрограмма - по существу универсальная версия PARK (см. [6.17](#_6.17_PARK_-)), и предоставлена в исходном коде, для демонстрации методов взаимодействия с жесткими дисками из прикладных программ.

### 6.20.1 Использование SPINUP

Эта утилита была написана для использования прежде всего в системах с батарейным питанием, где требуется сохранение электроэнергии. Обычно только более новые диски SCSI отвечают на команды Stop/Start Unit, управляя двигателями диска и перемещая головки в определенную “Область парковки”. SPINUP – управляется их командной строки и ожидает в качестве параметра, действительный номер устройства для физического устройства жесткого диска (см. [5.2.1](#_5.2.1_Область_конфигурации_2), CONFIG+61). Допустимые номера устройств “0”, “1” и “2”.

Остановка устройства обозначается, предшествующим номеру устройства знаком минус:

SPINUP -0

что заставляет устройство 0 запарковать головки и отключить питание мотора, если это возможно. Запуск устройства указывается просто указывая номер устройства в качестве параметра:

SPINUP 0

При запуске или остановке устройства жесткого диска, SPINUP контролирует состояние устройства после выдачи команды и сообщает о состоянии диска, когда получены результат выполнения. Результаты возвращены в виде сообщения: устройство остановлено, запущено, или произошла ошибка (см. [6.20.2](#_6.20.2_Сообщения_об) ниже).

Перед использованием SPINUP, для остановки жесткого диска, вам необходимо убедиться, что доступ к любому логическому диску на этом устройстве не произойдет, пока устройство остановлено, либо обменом логических дисков с помощью BPSWAP (см. [6.5](#_6.5_BPSWAP_-)) или изменяя командный процессор и пути поиска DOS в ZSDOS утилитой ZPATH. Например если ваша система включает в себя раздел на этом жестком диске как диск A:, вы зарегистрировались на виртуальном диске M: и устройство остановлено, то ZCPR3 и ZSDOS могут попытаться искать файл на диске A:, что приведет к ошибке чтения. В этом примере вы можете поменять диск A: с M:, убедившись что он не M: либо путь, или установить оба пути, чтобы исключить диск A:

### 6.20.2 Сообщения об ошибках SPINUP

+++ Not B/P Bios ... aborting +++

Была предпринята попытка выполнить SPINUP с BIOS отличным от B/P, или сделанные модификации в BIOS, изменили местоположение, или значения необходимые для корректного определения системы.

+++ Can't handle this Controller Type! +++

Тип контроллера в BIOS не может справиться с командами SCSI “Stop/Start Unit”.

\*\*\*\* SCSI Block Length Error !

BIOS не поддерживает расширенные команды, необходимые для включения и выключать привода с помощью функции устройств SCSI “Stop Unit” и “Start Unit”. Она вероятней всего появляется из-за изменений BIOS во время редактирования/компиляции, которые изменили размер команды дескриптора блока или функцию жесткого диска, которая возвращает значения.

\*\*\*\* Invalid Unit # !

Указанный номер устройства не является “0”, “1” или “2”. Только три физических устройства жесткого диска распознаются B/P BIOS.

## 6.21 TDD Утилита поддержки умных часов

TDD это адаптированная версия ZSDOS утилита TD. В системах ZSDOS2, он используется для отображения, установки или обновления часов B/P BIOS. Последняя возможность существует только для Dallas SmartWatch (DS-1216E) или JDR No-Slot-Clock в Ampro Little Board, SB180 или YASBEC.

### 6.20.1 Использование TDD

TDD получает системное время и дату с DOS функцией 98 и выводит информацию на консоль. В системе должен быть установлен драйвер часов, чтобы использовать эту утилиту. Если драйвер часов поддерживает функцию установки, TDD может установить дату и время, используя DOS функцию 99. При настройке часов TDD позволит вам работать в интерактивной режиме или в командной строке.

По TDD отвечает на стандартный запрос о помощи описанный в Разделе 1.7. Вы можете получить краткую справку, введя:

TDD //

Вы можете получить текущую дату и время из системных часов, просто введя имя программы:

TDD

Если вы используете часы на основе Dallas 1216E, вы можете установить часы B/P BIOS путем прямого ввода:

TDD U

Непрерывное отображение, обновляемое каждую секунду, до нажатия любой клавиши, может быть получено вводи команды:

TDD C

Системные часы могут быть установлены в интерактивном режиме, введя название программы, сопровождаемое параметром “S”:

Вам будет предложено ввести дату. В запросе будет отображаться формат, в котором дата будет принята (американский или европейский) либо как:

Enter today’s date (MM/DD/YY): - американский

или

Enter today’s date (DD.MM.YY): - европейский

Поля даты (месяц, день и год) могут быть или одной или двумя цифрами в каждой позиции. Недопустимые записи, такие как недопустимый день в течение введенного месяца приведет к повторному отображению подсказки для нового ввода.

Когда допустимая дата будет введена, вам будет предложено ввести текущее время. Приглашение будет варьироваться в зависимости от того, используете ли вы часы реального времени, или относительный счетчик заменяющий часы. Две соответствующие подсказки:

Enter the time (HH:MM:SS): - Часы реального времени

Enter the relative time (+XXXX): - Относительный счетчик

Предполагается, что время находится в 24-часовом формате, если используются часы реального времени и секунды могут быть опущены. Если используются относительные часы числу которое вы хотите присвоить относительному счетчику, должен предшествовать префикс ‘+’. Допускаются значения от +0 до +9999.

После завершения записи времени с помощью символа возврата каретки, вам будет предложено нажать любую клавишу. В этот момент, следующий нажатие клавиши (кроме Shift и Control) будет устанавливать часы. Эта процедура позволяет точно синхронизировать время с одним из многих источников точного времени.

Вы можете установить часы из командной строки, введя имя программы с дополнительным указанием даты и времени. Дата должна быть в правильном формате (США или европейском) для настроенного TDD. Если обнаружится ошибка, TDD перейдет в интерактивный режим и запросит дату и время, как описано выше.

### 6.20.2 Настройка TDD

TDD может быть настроен в настоящее время в любом формате: США месяц, день, год, как: Sep 18, 1988, или европейских и военный стиль как: 18 Sep 1988. Кроме того функция установки принимает формат США MM/DD/YY или Европы DD.MM.YY. По умолчанию могут быть установлены с помощью утилиты Al Hawley ZCNFG с использованием данных, содержащихся в файл TDD.CFG на дистрибутивном диске.

### 6.20.3 Сообщения об ошибках TDD

Сообщения об ошибках TDD просты и в основном не требует пояснений. Для ясности, однако, они приведены ниже.

SORRY! ZSDOS or ZDDOS is required to run this program!

Вы пытались запустить утилиту с чужой DOS. Используйте ZSDOS или ZDDOS. Эта ошибка прерывает работу программы и возвращает управление в командный процессор.

\*\*\* NO Clock Driver installed!!!

Вы пробовали читать часы, которых не существует. Установите часы с помощью SETUPZST и повторите попытку. Эта ошибка прерывает работу программы и возвращает управление в командный процессор.

\*\*\* Clock does NOT Support SET!!!

Драйвер часов на вашем компьютере не позволит вам установить время с помощью TDD. Эта ошибка прерывает работу программы и возвращает управление в командный процессор.

\*\*\* Must have B/P Bios to use!!!

Эта утилита работает только под B/P BIOS. Если вы используете B/P BIOS и это сообщение появляется, это скорее всего из-за редактирования (компилирования) исходного кода BIOS, который изменил критические значения в структурах данных.

\*\*\* Hardware Type Not Supported ! \*\*\*

Так как TDD неразрывно связан с указанными аппаратными платформами, он будет работать только если компьютера является одним из типов, для которых была выполнена корректного кода. Эта ошибка не должна появляться, если выдается команда “U” или “S”.

++ Insufficient Memory! ++

Начальный адрес доступной памяти был уменьшен ниже необходимого для части TDD, который перемещен в верхнюю память. Наиболее распространенная причина этого - добавление резидентных системных расширений (RSX), которые заставляют сообщать верхний адрес TPA ниже 8100H.

\*\*\* Error in Data Input

Недопустимый символ или число было введено при попытке установить дату и время. Эта ошибка может привести к переходу в интерактивный режим, и выдавать запрос для повторного ввода правильных значений даты/времени.

\*\*\* Must be wheel to set clock!

Попытка была предпринята, чтобы установить часы без доступа колеса. Используйте ZSCONFIG (ZSDOS 1) или ZSCFG2 (ZSDOS2), чтобы установить допустимый байт колеса или отключить его (см. [6.22](#_6.22__ZSCFG2) для ZSCFG2 или руководство ZSDOS 1 для ZSCONFIG). Эта ошибка прерывает работу программы и возвращает в командный процессор.

\*\*\* Must have Z180 Processor!!!

Эта ошибка может появиться, если BIOS идентифицирует тег ID для компьютера с Z180/HD64180, но происходит сбой проверки для данного процессора. Это, вероятно, из-за неправильного портирования BIOS или манипулирования структурой заголовка.

\*\*\* Must have Z180 to Set No-Slot-Clock!!!

Ошибка аналогичная предыдущей, но будет появляться при попытке настроить часы al altered system.

\*\*\*\* Can't find No-Slot-Clock!

JDR No-Slot-Clock/Dallas DS-1216E SmartWatch не могут быть проверены в системе. Это сообщение будет выведено, если TDD будет выполняться без установленных часов, или часы перестали работать по некоторым причинам.

## 6.22 ZSCFG2 Утилита конфигурации ZSDOS2

ZSCFG2 – программа настройки различных параметров операционной системы ZSDOS2. Она включена в это руководство из-за тесного взаимодействия всех частей операционной системы, особенно BIOS с переключением банков памяти. Программа ZSCFG2 работает или в интерактивном (новичок), или в управляемом из командной строки (экспертном) режиме для максимальной гибкости и простоты использования. Если ваш компьютер использует ZCPR3, автоматически обнаруживается среда Z3, и ZSCFG2 будет использовать атрибуты экрана, такие как инверсное изображение и адресуемый курсор для улучшения отображения.

Как во всех наших подпрограммах поддержки, краткое сообщение онлайн-справки доступно, вводя имя, сопровождаемое двумя наклонными чертами:

ZSCFG2 //

Этот инструмент конфигурирования автоматически адаптирует себя в соответствии с системой ZSDOS2, и все сообщения, от справки до интерактивных подсказок точно отразят параметры и состояние для рабочей конфигурации ZSDOS2.

### 6.22.1 Интерактивный режим ZSCFG2

Чтобы запустить ZSCFG2 в интерактивном режиме, просто введите название программы:

ZSCFG2

Если обнаружена допустимая среда ZCPR3, экран очистится, и вы будете видеть, что экран содержит необходимые адреса среды и табличное отображение текущих параметров настройки работающей системы ZSDOS2. При наличии используется инверсное изображение, чтобы улучшить отображение. Если вы используете компьютер, на котором не установлен ZCPR3 или который не поддерживает прямую адресацию курсора, информация (за исключением ZCPR3) просто построчно выводится на экран, прокручивая экран. Единственное различие в содержании между двумя выводами заключаются в том, что данные о среде ZCPR3 не будут выведены на экран. Пример экрана ZSDOS2:

┌───────────────────────────────────────────────────────────────────────────┐

│ ...Configuring ZSDOS Ver 2.0 Z3 Environment at : FE00H │

│ ZCPR Path Address : FDF4H │

│ Wheel Byte at : FDFFH │

│ │

│ 1 - Public Files : YES │

│ 2 - Pub/Path Write Enable : NO │

│ 3 - Read-Only Vector : YES │

│ 4 - Fast Fixed Disk Log : YES │

│ 5 - Disk Change Warning : NO │

│ 6 - Path w/o System Attr : YES │

│ 7 - DOS Search Path : Enabled - Internal │

│ 8 - Wheel Byte Protect : Enabled Addr = FDFFH │

│ T - Time Routine (Clock) : F168H │

│ A - Stamp Last Access Time : Disabled │

│ C - Stamp Create Time : EEB2H │

│ M - Stamp Modify Time : EEBCH │

│ │

│ Entry to Change ("X" to EXIT) : \_ │

└───────────────────────────────────────────────────────────────────────────┘

Тип операционной системы и номер версии отображаются в верхней части экрана, затем следуют вся необходимая информация о среде ZCPR3. Если среда ZCPR не найдена, появится сообщение об этом. В таком случае некоторые параметры будет ограничены, как описано ниже в подробном описании.

Интерактивная работа состоит просто из ввода цифры или буквы слева от каждой строки для выбора функции. Если вы выбираете цифры от единицы до шести, значение параметра изменится с OFF на ON или наоборот, и меню и состояния снова будут выведены на экран. Если вы выберете какой-либо из оставшихся элементов, то у вас запросят более подробную информацию. [Раздел 6.22.2](#_6.22.2_Использование_командной) ниже, содержит подробное описание всех параметров.

### 6.22.2 Использование командной строки (режим эксперта) ZSCFG2

Вход командной строки (или режим эксперта) дает возможность динамически задавать параметры ZSDOS сценарии запуска, псевдоним ZCPR файлов, отправки файлов или непосредственно из консоли. Это позволяет параметры операционной системы для конкретных задач, и восстановлен после завершения. Например, Submit или команда псевдонима могли бы выполнить следующую последовательность:

Disable/Enable ZSDOS features and set addresses

...Process application programs

Restore ZSDOS features and addresses and return

Адаптация ZSDOS в этой последовательности будет через вызов ZSCFG с параметрами, переданными в командной строке в скрипте. В этом режиме, вы не должны постоянно присутствовать у компьютера, чтобы изменить параметров DOS.

Параметры передаются ZSCFG как группы символов, разделенные одним или несколькими символами табуляции, пробела или запятыми. Каждая группа символов начинается с идентификатора, который идентифицирует параметр, который будет изменен. В случае элементов, связанных со временем и датой, используется последовательность из двух символов: знак “+” идентифицирует команду как часы или связанную с меткой времени функцию, и следующий символ говорит какой из этих шести параметров будет изменяться. Для ZSDOS идентификаторы команд в режим эксперта следующие:

**P** - Поддержка общих файлов

**W** - Разрешение записи общий/путь

**R** - Поддержка диска только для чтения

**F** - Быстрая перерегистрация жесткого диска

**!** - Предупреждение при смене диска

**S** - Путь без системных атрибутов

**>** - Путь поиска ZSDOS

**\*** - Защита от записи с помощью байта колеса

**C** - Адрес подпрограммы часов

**+A** - Адрес установки временной метки времени доступа

**+C** - Адрес установки временной метки создания

**+M** - Адрес установки временной метки изменения

Параметры, которые являются просто переключателями On/Off, включаются вводом. только одного символа и отключаются следующим за символом знака "минус" (“-”). Например, следующая команда включит быструю перерегистрацию жесткого диска и отключит предупреждения о смене дисков:

**ZSCFG2 F,!-**

Некоторые параметры требуют дополнительных параметров, которые обрабатываются вторичной строкой в интерактивном режиме. Поскольку в экспертном режиме запрос не выдается, дополнительные параметры передаются путем их добавления к их идентификаторам команд. Например чтобы задать адрес ZCPR байта колеса защиты от записи, активировать внутренние пути поиска и установить адрес часов в стандартный вектор часов B/P BIOS введите команду:

**ZSCFG2 \*Z,>I,CB**

Помните, что пробелы или другие разделители (табуляция, запятые, и т.д.) не разрешены между идентификаторами команды и их параметрами. Обычно произойдет недопустимая ошибка, если вы забыли это. Все адреса вводятся как шестнадцатеричные (основание 16), числа с дополнительными начальными нулями. Используемый алгоритм, интерпретации вводимого числа, сохраняет только заключительные четыре шестнадцатеричных цифры, поэтому если вы вводите последовательность “0036C921045”, она интерпретируется как 1045H.

В следующем разделе описаны различные параметры для каждого варианта.

### 6.22.3 Описания параметров ZSCFG2

Два инструмента, которые позволяют настраивать систему ZSDOS2 к вашим определенным потребностям, INSTAL12 и ZSCFG2, имеют одинаковый интерактивный дисплей. Этот раздел, следовательно, применим к установке, а также настройке “на лету” с помощью ZSCFG2. Каждый параметр также включает конкретные аргументы для ввода параметра в командной строке. Параметры описываются в порядке их появления в интерактивные меню INSTAL12 и ZSCFG2.

#### 6.22.3.1 Общее файлы

Интерактивная подсказка : **1** - Public Files

(переключатель)

Символ командной строки: **P**

Включить - **P**

Отключить - **D-**

Этот параметр управляет распознаванием бита атрибута общий Plu\*Perfect (бит 7 второй буквы в имени файла). Если параметр установлен в YES или активирован, любой файл, с установленным этим битом, будет доступен из любой области пользователя на диске. Это означает, что поиск файла определит его местоположение при первой попытке, независимо от выбранной области пользователя (см. 2.9.4, Общий доступ). Если параметр установлен в NO, файл будет личным и может быть найден только в соответствующей области пользователя. Настройка по умолчанию для этого параметра - YES, чтобы распознать общие файлы.

#### 6.22.3.2 Разрешение записи общий/путь

Интерактивная подсказка : **2** - Pub/Path Write Enable

(переключатель)

Символ командной строки: **W**

Включить - **W**

Отключить - **W-**

Если параметр установлен в YES или активирован в режиме командной строки, ZSDOS2 будет разрешить операции записи общих файлов, и в случае ZSDOS2, файлы, расположенные вдоль пути. Если параметр установлено в NO или отключен, попытки записи в файл приведут к ошибке "Только для чтения". Настройка по умолчанию для этого параметра - NO.

#### 6.22.3.3 Поддержка вектора только для чтения

Интерактивная подсказка : **3** - Read-Only Vector

(переключатель)

Символ командной строки: **R**

Включить - **R**

Отключить - **R-**

Если этот параметр имеет значение YES или включен, обычно установленный вызовом Функции 28 вектор защиты от записи ZSDOS2 не будет очищен при теплом старте как в CP/M и ZRDOS. Если установлен в NO или выключен, вектор защиты от записи будет функционировать как в CP/M и ZRDOS.

#### 6.22.3.4 Быстрая перерегистрация жесткого диска

Интерактивная подсказка : **4** - Fast Fixed Disk Log

(переключатель)

Символ командной строки: **F**

Включить - **F**

Отключить - **F-**

Если этот параметр имеет значение YES или включен, битовый массив распределения для несъемного диска (тот, в котором буфер WACD - ноль), не будет восстановлен после начальной регистрации диска в системе. Это приводит к более быстрой работе систем с жесткими или виртуальными дисками. Если установлен в NO или выключен, то битовый массив распределения будет восстановлен каждый раз, когда жесткий диск в первый раз выбирается после теплого старта. Настройка по умолчанию для этого параметра - YES.

#### 6.22.3.5 Предупреждение при смене диска

Интерактивная подсказка : **5** - Disk Change Warning

(переключатель)

Символ командной строки: **!**

Включить - **!**

Отключить - **!-**

Когда этот параметр имеет значение YES или включен, будет выведено предупреждение всякий раз, когда ZSDOS2 обнаруживает, что съемный диск в дисководе (обычно дискета) был изменен. Если нажать любую клавишу, кроме Ctrl-C, ZSDOS2 автоматически регистрирует новый диск и продолжает работу. Если параметр установлен в NO, или выключен, предупреждение не выводится, диски будут автоматически регистрироваться, и выполнение операции продолжится. Настройка по умолчанию для этого параметра - NO.

#### 6.22.3.6. Путь без системных атрибутов

Интерактивная подсказка : **6** - Path w/o System Attr

(переключатель)

Символ командной строки: **S**

Включить - **S**

Отключить - **S-**

Если параметр установлен в YES или включен, общие файлы на дисках по пути будут найдены без установленного атрибута системный (см. 2.9.2, Путь доступа к каталогу). Если этот параметр установлен NO или отключен, то общие файлы на диске по пути, не будут найдены, если бит атрибута системный (бит 7 второго символа в расширении файла), установлен (см. 2.9.3, Путь доступа к файлу). Настройка по умолчанию для этого параметра - NO или отключен, требуя установки бита общий для доступа к файлам.

#### 6.22.3.7 Путь поиска DOS

Интерактивная подсказка : **7** - DOS Search Path

Опции - (**D**)isable, (Отключить)

(**S**)et addr, (Установить адрес)

(**I**)nternal (Внутренний)

(**Z**)CPR3 (Только если запущена ZCPR3)

Символ командной строки: **>**

Включить - **>addr**, **>I**

**>Z** (Только если запущена ZCPR3)

Отключить - **>-**

При выборе этого параметра в интерактивном режиме, вам будет предложен выбор одного из трех вариантов. В отличии от настройки ZSDOS1 требуется описатель среды ZCPR3 стиля, поэтому всегда будет отображаться следующее сообщение:

DOS Path [(D)isable, (S)et, (I)nternal, (Z)CPR3] :

Запуск ZSCFG2 в режиме командной строки позволяет вам выбирать те же параметры из командной строки, которые кратко описаны выше. Для отключения - (**D**)isable, выбора внешнего - (**I**)nternal или пути (**Z**)CPR3 дополнительные символы не требуются. Если вы выбираете вариант установить - (**S**)et, предлагает вам ввести шестнадцатеричный адрес:

Enter PATH Address :

Если вы отключаете (**D**)isable параметр путь DOS, ZSDOS2 функционирует как CP/M 2.2 и ZRDOS. Запросы на доступ к файлам только с текущего диска и пользователя, изменяется только Public возможностью, если она активна. Это приводит к знакомым требованиям утилит установки, таких как компиляторы, текстовые процессоры и системы управления базами данных, чтобы сказать им, где искать их оверлейные программы.

Правильное использование пути DOS преодолевает ограничения в поиске оверлейных программ и других файлов, просто установив путь DOS к области диска и пользователя, где хранятся соответствующие оверлеи и другие файлы. Путь может задаваться тремя способами.

Первый способ заключается в том, чтобы назначить фиксированный адрес с помощью варианта (**S**)et в интерактивном режиме, или добавить адрес к символьной команде в режиме командной строки. Вы будете ответственны за обеспечение того, чтобы любой путь в этом адресе соответствовал надлежащим определениям пути ZCPR3.

Второй способ установить путь DOS состоит в том, чтобы использовать внутренний путь ZSDOS с тремя элементами при выборе варианта (**I**)nternal в интерактивном режиме или добавлении “I” после символа команды в режиме командной строки. В дистрибутиве, внутренний путь ZSDOS состоит из записи “A0:” который направляет путь поиска в область пользователя 0 на диске A. Альтернативный способ активировать внутренний путь, с помощью утилиты ZPATH.COM из ZSDOS1. ZPATH позволит вам изменить путь по умолчанию и определить до трех элементов поиска диск/пользователь.

Последний способ установки пути DOS доступен только, если вы работаете в системе ZCPR3. Выбрав параметр **(Z)CPR3** в интерактивном режиме, или указания после символа команды “Z” в режиме командной строки. Скорее всего, этот режим пути, будет мало использоваться, но он доступен для систем, которые нуждаются в более, чем в трех элементах пути. Основным недостатком использования пути ZCPR3 заключается в том, что запросы из командной строки (например, А0>) могут привести к N-поискам, где N-число элементов пути. Это происходит потому, что ZCPR3 будет выбрать первый элемент пути, и ZSDOS будет последовательно вести поиск вдоль всего пути, если командный файл не найден, возвращаться в ZCPR3 с ошибкой “файл не найден”. Затем ZCPR3 выбирает второй элемент и ZSDOS снова будет искать командный файл вдоль всего пути для. Такая ситуация не возникает после запуска в прикладных программах, так как командный процессор ZCPR3 больше не является активным.

Значение по умолчанию для DOS пути – “Внутренний”.

#### 6.22.3.8 Защита от записи с помощью байта колеса

Интерактивная подсказка : 8 - Wheel Byte Protect

Опции - (**D**)isable, (Отключить)

(**S**)et addr, (Установить адрес)

(**Z**)CPR3 (Только если запущена ZCPR3)

Символ командной строки: \*

Включить - \*addr

\*Z

Отключить - \*-

Когда вы выберите этот параметр в интерактивном режиме ZSCFG, вам будет представлена дополнительная строка, содержащая доступные варианты:

Wheel [(D)isable, (S)et, (Z)CPR3] :

Выбор параметра (**D**)isable, вводя “D”, или отключение байта колеса последовательностью “\*-” в режиме командной строки, заставляет ZSDOS предполагать, что байт колеса всегда включен, давая всем пользователям полные полномочия в управлении файлом (запись, переименование и стирание). Ввод “S” для параметра (**S**)et в интерактивном режиме позволяет вам вводить шестнадцатеричный адрес своего байта колеса. Ваша обязанность гарантировать, что байт защищен по мере необходимости от неумышленного изменения. Установка адреса байта колеса из командной строки просто требует добавления шестнадцатеричного адреса после управляющего символа колеса. Ввод в интерактивном режиме, “Z” или строки параметров “\*Z” в режиме командной строки устанавливает адрес байта колеса в B/P ZCPR3 совместимой среде. По умолчанию этот параметр OFF или отключен предполагая, что пользователь имеет полные привилегии.

#### 6.22.3.9 Подпрограмма времени (драйвер часов)

Интерактивная подсказка : T - Time Routine (Clock)

Опции - (**D**)isable, (Отключить)

(**S**)et addr, (Установить адрес)

Символ командной строки: C

Включить - C addr

Отключить - C-

Этот параметр позволяет пользователю ввести адрес процедуры драйвера часов, соответствующей стандартам ZSDOS или отключить существующую процедуру часов. Ввод “T” в интерактивном режиме вызывает появление следующего сообщения:

Time (Clock) Routine [(D)isable, (S)et, (B)ios+4EH] :

Ввод “D” в интерактивном режиме или последовательность “C-” в командной строке отключит любые существующие часы. Это приводит к ошибкам при возврате из DOS вызовов функций, 104 и 105, и отключает функцию записи меток даты/времени файла. Если в данный момент вы ответите “S” в интерактивном режиме, вам будет дополнительно предложено ввести шестнадцатеричный адрес драйвера часов. То же действие установки адреса драйвера часов достигается в режиме командной строки, вводом символа команды “C”, а затем шестнадцатеричного адреса начинающегося с цифры. При выборе “B”, или ввода “CB” в качестве аргумента командной строки, будет установлен DOS адрес драйвера часов расположенный от начала B/P BIOS со смещением 4EH, который соответствует записи в таблице переходов для B/P ZSDOS-совместимого драйвера часов. Не вводите неизвестные значения, так как результаты могут быть непредсказуемы!

#### 6.22.3.10 Метки времени последнего доступа

Интерактивная подсказка : **A** - Stamp Last Access Time

Опции - (**D**)isable, (Отключить)

(**E**)nable, (Включить)

Символ командной строки: **+A**

Включить - **+A**

Отключить - **+A-**

Этот параметр доступен только с временными метками типа DateStamper. Для P2DOS функция не определена и игнорируется в ZSDOS2. Как указано в Разделе 3.4.4.2, если у вас нет определенной потребности в сохранении записи времени последнего доступа к файлу, мы рекомендуем, отключить этот параметр для уменьшения ненужных накладных расходов. Чтобы выбрать параметр времени последнего доступа, введите “A” в интерактивном меню. Это приведет к отображению следующего сообщения:

Stamp Last Access Time Routine [(D)isable, (E)nable] :

если в этот момент вы введете “D” в интерактивном режиме или отключите функцию последовательностью “+A-” в режиме командной строки, то время не будет занесено в поле последнего доступа в файл DateStamper. Этот параметр снова может быть включен выбором опции “Е” в интерактивном режиме из этой второй строки или последовательностью “+A” в режиме командной строки.

#### 6.22.3.11 Метки времени создания

Интерактивная подсказка : **C** - Stamp Create Time

Символ командной строки: **+C**

Включить - **+C addr**

Отключить - **+C-**

Ввод “С” в главном меню интерактивного режима позволит вам включить или отключить функцию создания меток времени. Затем появляется второй запрос:

Stamp Create Time Routine [(D)isable, (E)nable] :

Чтобы отключить создание меток времени, введите “D” во втором приглашении, или последовательность “+С-” в режиме командной строки. Чтобы включить создание меток времени, введите “Е” во второй строке в интерактивном режиме или введите в качестве параметра последовательность “+С” в режиме командной строки.

#### 6.22.3.12 Метки времени изменения

Интерактивная подсказка : **C** - Stamp Modify Time

Опции - (**D**)isable, (Отключить)

(**E**)nable, (Включить)

Символ командной строки: +M

Включить - +M

Отключить - +M-

Время последнего изменения файла является, вероятно, самым ценным из времен, предлагаемых в системе ZSDOS. Таким образом, у вас, вероятно, никогда не появится необходимость изменять этот параметр. Однако, в случае возникновении необходимости, введите “M” в главном интерактивном меню. Затем вам будет предложено:

Stamp Modify Time Routine [(D)isable, (E)nable] :

Если в данный момент ввести “D” в интерактивном режиме или отключить функцию с помощью последовательности “+M-” в режиме командной строки, время не будет сохраняться в поле "изменения" при любой активности меток времени.

### 6.22.4 Сообщения об ошибках ZSCFG2

Только два сообщения об ошибках существуют в ZSCFG2. По большей части любая ошибка, которую Вы видите, будет иметь дело с неправильными параметрами или ошибками ввода. Эти два сообщения об ошибках:

-- Invalid --

Неверный адрес или символ был введен в качестве параметра.

\*\*\* ERROR: DOS is not ZSDOS2!

Была предпринята попытка запустить ZSCFG2 в операционной системе, которая не является ZSDOS2. Эта программа не может работать под любой другой операционной системой.

## 6.23 ZXD – Утилита вывода файлов для ZSDOS2

Версия 1.66 ZXD - модификация более ранней версии, выпущенной с нашим пакетом ZSDOS 1. Это - программа ZSDOS расширенного отображения листинга каталога, полученная из утилиты ZCPR3 XD III написанной Ричардом Конном {Richard Conn} и теперь измененная для возврата правильной информации о размере дисков в операционной системе ZSDOS2 с переключением банков памяти. Были добавлены многие дополнительные возможности, не последним из которых является возможность отображения временных меток для каждого файла в различных форматах. ZXD может отображать метки даты и времени файлов типа DateStamper, P2DOS, и метки DosDisk Plu\*Perfect Systems. В системах ZCPR3, используется байт колеса, чтобы отключить некоторые функции в целях безопасности в системах удаленного доступа.

### 6.23.1 Использование ZXD

ZXD запускается, вводом его имени в командной строке и может сопровождаться необязательными спецификациями диска и пользователя, чтобы получить доступ к каталогу на другом диске или области пользователя. Она может также сопровождаться различными параметрами, которые изменяют формат и/или содержание вывода. Вы можете получить краткое сообщение справки введя:

ZXD //

ZXD принимает эту информацию в естественной форме широко используемой с серии ZCPR замен процессора команд. Используя соглашения, описанные в Разделе 1.2, синтаксис ZXD следующий:

ZXD [dir:][afn] [/][options]

Если ZXD вызывается без параметров, выводятся только те файлы, которые удовлетворяют встроенным условиям по умолчанию. Обычно эти значения по умолчанию выбирают только несистемные файлы на текущем диске и области пользователя. Значения по умолчанию могут быть изменены необязательными параметрами, описанными ниже. Если необязательные параметры желаемы без диска, пользователя или спецификаций файла, то параметрам должен предшествовать префикс наклонной черты черта, косая черта является необязательной, если вводятся любые перенаправления или спецификации файла.

### 6.23.2 Параметры ZXD

Необязательные параметры, состоящие из одного или двух символов, позволяют получить информацию, выбранную из файлов на диске, или адаптировать в соответствии с вашими конкретными потребностями. Каждый из этих параметров также используется как встроенное значение по умолчанию. После принятия решения, какие параметры вы используете чаще всего в командной строке, мы рекомендуем настроить ZXD, чтобы использовать эти параметры в качестве значений по умолчанию. В результате уменьшится количество нажатий клавиш и, следовательно, будет быстрее работа, при сканировании каталогов.

Символы параметров описаны в следующих разделах в алфавитном порядке.

#### 6.23.2.1 Выбор файлов по атрибутам

Чтобы избежать загромождения отображения каталогов с именами нежелательных файлов ZXD имеет параметр, который управляет выбором этих файлов с установленным битом системного атрибута. Параметр “A” управляет этой функцией. Ему требуется второй символ “S”, “N”, или “A”. Управление, предлагаемое этими символами:

S Включать только файлы с атрибутом "Системный".

N Включать только файлы без атрибута "Системный" (это условие по умолчанию).

A Включить все файлы

Поскольку перечисление всех несистемных файлов является условием по умолчанию, вы, вероятно, не используете параметр “N” очень часто. Параметр “A”, с другой стороны, предлагает простой способ просмотра всех файлов в текущем каталоге, в том числе системных файлов, которые обычно невидимы из-за бита атрибута.

В системе ZCPR3, где не был предоставлен доступ колеса (байт колеса выключен), этот параметр используется только для несистемных файлов и параметр с символом “A” не допускается.

#### 6.23.2.2 Формат отображения даты

ZXD может отображать даты в формате США MM/DD/YY или в европейском формате DD.MM.YY. Вы можете изменить формат по умолчанию с помощью параметра “D”. Вот пример двух типов отображения даты:

В формате США:

**ZXD Ver 1.66 3 Apr 1993 15:43:17**

**Filename.Typ Size Modified Filename.Typ Size Modified**

**-------- --- ---- -------- -------- --- ---- --------**

**INITDIR .COM 4k 07:01-09/17/88 ZPATH .COM 4k 07:50-09/17/88**

**ZXD .COM 8k 08:01-09/17/88**

**C2: -- 3 Files Using 16K (324K Free)**

В европейском формате:

**ZXD Ver 1.66 3 Apr 1993 15:43:11**

**Filename.Typ Size Modified Filename.Typ Size Modified**

**-------- --- ---- -------- -------- --- ---- --------**

**INITDIR .COM 4k 07:01-17.09.88 ZPATH .COM 4k 07:50-17.09.88**

**ZXD .COM 8k 08:01-17.09.88**

**C2: -- 3 Files Using 16K (324K Free)**

#### 6.23.2.3 Отключить отображение даты (без даты)

Хотя отображение информации о дате и времени в ZXD является режимом по умолчанию, оно может быть отключено с помощью параметра “N” для отображения большего числа имен файлов на экране.

#### 6.23.2.4 Параметр управления выводом

Параметр “O” управляет выводом ZXD на принтер или экран, и требует второго символа, который добавляет дополнительное управление к выходным форматам. Вторые распознаваемые символы:

F Посылать символ перевода страницы в конце списка.

H Переключение отображения горизонтальной/вертикальной отсортированного списка.

#### 6.23.2.5 Выход на принтер

Параметр “P” контролирует вывод на принтер. Когда этот параметр задан, отсортированный листинг каталога отправляется экран консоли и принтер. Этот параметр отключен и не доступен в системе ZCPR3, где колесо доступ не был предоставлено (байт колеса выключен).

#### 6.23.2.6 Сортировка по имени или расширению

Условие сортировки по умолчанию для ZXD - первая сортировка по имени файла, затем по расширению файла в пределах совпадающих имен. Параметр “S” изменяет эту последовательность на противоположную.

#### 6.23.2.7 Первичные временные метки

ZXD имеет алгоритм, который будет пытаться найти один из нескольких типов меток даты/времени для каждого файла. Условия по умолчанию требует от ZXD, сначала пытаться найти метки типа DateStamper. Если это не удается, происходит поиск меток DosDisk на дисках MS/PC-DOS, и наконец, проверяется метки типа P2DOS. Параметр “T” указывает исключить проверку DateStamper, тем самым ускоряя время поиска, если метки типа DateStamper никогда не использовались.

#### 6.23.2.8 Все области пользователя

Дистрибутивная версия ZXD будет искать файлы только в области одного пользователя, либо текущего или явно указанного. Параметр “U” предписывает искать файлы во всех пользовательских областей на диске. Сочетание параметров “U” с “AA” выведет список всех файлов на диске во всех пользовательских областях, как системные, так и несистемные. Эта опция отключена и не доступна в системе ZCPR3, где колесо доступ не предоставлено (байт колеса выключен).

#### 6.23.2.9 Широкоформатный дисплей

ZXD отображает только метки даты/времени “Last Modified” (последнего изменения). Это может быть отменено путем добавления параметра “W” в командной строке, который генерирует широкий вывод из всех доступных меток. Только тип DateStamper имеет поля для всех трех категорий меток. P2DOS содержит только метки создания и изменения, в то время как одна метка MS/PC-DOS, доступна DosDisk наиболее соответствуя “Modified”. Вывод созданный с этим параметром:

Filename.Typ Size Created Last Access Modified

-------- --- ---- ------- ---- ------ --------

BU16 .COm 8k 17:26-06/12/88 08:42-08/21/88 17:26-06/12/88

COPY .COM 8k 15:06-09/17/88 15:06-09/17/88

ZPATH .COM 4k 07:50-09/17/88 15:02-09/17/88 07:50-09/17/88

ZXD .COM 8k 08:00-09/17/88 08:01-09/17/88

### 6.23.3 Настройка ZXD

Утилита конфигурации ZCNFG.COM используется для изменения параметров по умолчанию ZXD. Настройки параметров и текстовые подсказки ZXD содержатся в файле ZXD.CFG, который должен быть доступен ZCNFG для любого изменения конфигурации. Все настраиваемые параметры - простые ON/OFF переключатели или инвертируемые параметры и соответствуют параметрам, описанным в [Разделе 6.23.2](#_6.23.2_Параметры_ZXD) выше. Посмотрите *Раздел 4.8 Руководства пользователя ZSDOS 1.0* или документацию ZCNFG на Z-Node Ladera для получения дополнительной информации об использовании ZCNFG.

# 7 ZSDOS версии 2

В настоящее время ZSDOS версии 2 находится в стадии развития. Версия, предоставленная в этом пакете, предварительная и не должна считаться законченным продуктом. Убедитесь, что вы делаете резервные копии всех файлов, которые вы не хотите потерять, и сообщаете нам как можно больше информации о любых проблемах, с которыми вы сталкиваетесь.

Помимо вызова “Возврат версии ZSDOS” (Функция 48), который возвращает 20H, означает ZSDOS2, были добавлены три новые функции операционной системы:

|  |  |  |  |
| --- | --- | --- | --- |
| Функция 46 | | Возврат свободного места на диске | |
| Вход: | C = 46 (номер функции)  E = Номер диска (A=0..P=15) | Вывод: | A=0 если Ok, <>0 если Ошибка  Свободное место на диске в килобайтах помещается начиная с DMA+0 (младший бит) по DMA+3 (старший бит). |

Эта функция возвращает свободное место на диске в системе с переключением банков, в которых буферы ALV недоступны прикладным программам непосредственно. Она должна использоваться для надежного определения свободного места, так как у программ нет возможности выяснить в каком системном банке (если используются более одного) содержится битовая карта распределения. Для наиболее вероятного размера системы, будут использоваться только два или три нижних байта, но выделяются четыре байта для размещения системы максимального размера.

|  |  |  |  |
| --- | --- | --- | --- |
| Функция 49 | | Возврат адреса идентификатора среды | |
| Вход: | C = 49 (номер функции) | Вывод: | HL= Адрес идентификатора среды |

Эта функция возвращается, адрес ZCPR 3.4 "ввода" идентификатора среды, необходимый в системах B/P BIOS. Вместо того, чтобы полагаться на процессор команд, вставляющий адрес ENV в прикладные программы после выполнения, может использоваться эта функция, чтобы надежно получить адрес ENV в любое время.

|  |  |  |  |
| --- | --- | --- | --- |
| Функция 152 | | Разбор имени файла | |
| Вход:  DMA | C= 152 (номер функции)  DE= Указатель FCB назначения  --> Начало разбираемой строки | Вывод: | A= Число "?" в fn.ft  DE= Указывает на разделитель  FCB+15 = 0, если разбор успешен,  0FFh если произошла ошибка |

Эта функция может использоваться для замены библиотечных подпрограмм Z3LIB более надежным образом и поэтому производить небольшие программы приложений. Она полностью соответствует спецификации разбора ZCPR 3.4.

# 8 ZCPR Версия 4

Z40.ZRL - объединение ZCPR34 и многих функций обычно используемых RCP, модифицированных в связи с переключением банков памяти для большей части процессора команд, как это возможно. Когда Z40 используется в системе с переключением банков, вам, возможно, не понадобится большая часть, или весь резидентный процессор команд в вашей системе. Выпуск Z40 для ZSDOS2 и не будет работать без него, поскольку анализатор командной строки и вычисление свободного места на диске были удалены в пользу сервисов ZSDOS2. Кроме того, строка приглашения к вводу команд выводит на экран время и будет функционировать правильно, только если включены часы ZSDOS2. Комментарии о работе этих новых компонентов системы будут приветствоваться.

Более полная документация предоставляется в файлах Z40.HLP, включённых в дискеты дистрибутива, и список активных функций доступен с помощью команды “H” в командной строке. Для чтения файлов интерактивной справки, используйте программу HELP.COM доступную для скачивания из любого Z-Node.

# Глоссарий

**Прикладные программы**. В отличие от утилит (см), прикладные программы или приложения - крупные программы, такие как текстовые процессоры, которые функционируют в интерактивном режиме с пользователем.

**BDOS**. Базовая дисковая операционная система. Машинно-независимая, но, как правило, зависимая от процессора программа, которая контролирует взаимодействие между прикладными программами и машинно-зависимыми устройствами, такими как принтеры, дисководы, часы и т.д. Она также устанавливает понятие файлов на периферийных устройствах хранения информации и контролирует открытие, чтение, запись, и закрытие таких конструкций.

**BGii**. Система переключения задач работы с окнами для пользователей CP/M с жесткими дисками или псевдодисками BackGrounder ii от Plu\*Perfect Systems.

**BIOS**. Базовая система ввода-вывода. Машинно-зависимые подпрограммы, которые выполняют фактическое управление периферийным устройством, такое как отправка и получение символов к консоли, чтению и записи в дисководы, и т.д.

**Бит**. Двоичная цифра. Элемент, у которого может быть только состояние включен или выключен.

**Boot**. Термин, используемый для запуска компьютера. Как правило, применяется начиная с "холодного" старта, или выключенного, и включает в себя загрузку операционной системы, и действия по ее настройке.

**Байт**. Объединение восьми битов.

**CPR.** Замена процессора команд. Заменяет CCP (см. ниже). Пример: ZCPR

**CCP**. Процессор консольных команд (Console Command Processor). Часть операционной системы, которая интерпретирует команды пользователя и или выполняет их непосредственно или загружает прикладные программы с диска для выполнения. CCP может быть перезаписан приложениями и перезагружен функцией BIOS "Теплый" старт.

**Контрольная сумма**. Значение, которое арифметически суммирует содержание серии ячеек памяти, и используется для быстрой проверки текущего содержания на ошибки.

**Драйвер часов**. Программное связь между ZSDOS, без переключения банков и часами вашей системе. Драйвер часы позволяет ZSDOS и ее утилитам чтение часов, которые обычно присущи B/P BIOS.

**Командный скрипт**. Иногда их называют просто скрипты, командные скрипты позволяют создать единую команду, которая выдает другие команды для выполнения уникального набора действий. файлы submit CP/M один вид командного сценария, знакомый всем пользователям CP/M. ZCPR также предлагает более сложные типы сценариев, таких как псевдонимы (aliases) и командные файлы (command files) (например, ALIAS.CMD).

**DateStamper**. Пакет программного обеспечения, разработанный Plu\*Perfect Systems, для маркировки даты и времени файлов. ZDDOS содержит DateStamper внутренне, в то время как ZSDOS использует внешний модуль, чтобы реализовать DateStamper. DateStamper уникален среди программ реализующих временные метки файла для микрокомпьютеров по двум причинам: во-первых, он поддерживает все метки файла в файле; во-вторых, он поддерживает метки для создания, доступ и изменения время/дату для каждого файла.

**DDT**. Динамический инструмент отладки. Утилита распространяется с CP/M 2.2, которая может отображать, разбирать или изменять файлы диска или области памяти, используя коды операций или шестнадцатеричные значения.

**DOS**. Дисковая операционная система. Часто используется термин для BDOS, но, как правило, относится к совокупности CCP, BDOS и BIOS.

**DosDisk**. Пакет программного обеспечения разработанный Plu\*Perfect Systems, которая позволяет пользователям CP/M и совместимых компьютеров, читать и писать файлы напрямую с или на стандартную 5,25" DS DD c 40-дорожкамм дискету в формате MS-DOS. Это стандартный "360K" формат диска используемый в IBM-PC-совместимых компьютеров.

**FCB**. Блок управления файлом. Стандартная структура памяти, используемая CP/M и совместимыми операционными системами для осуществления дисковых операций с файлами.

**Атрибуты файлов.** Зарезервированные биты, которые хранятся вместе с именами файлов в каталогах дисков, определяющие доступ к файлам. См Приложение 5.

**Hexadecimal**. (Шестнадцатеричный) Система счисления с основанием 16, состоящая из цифр 0-9 и букв A-F. Часто используется для представления байта в виде двух цифр (от 00 до FF). Использование шестнадцатеричных чисел, как правило, сопровождается суффиксом “H”, например “01H”.

**IOBYTE**. Байт Ввода/Вывода. Резервный байт с адресом 3, который используется некоторыми CP/M BIOS, чтобы перенаправить ввод и вывод между устройствами, такими как терминалы и принтеры.

**K**. Обычно относится к Килобайту или 1024 (2^10-ой степени) байт.

**P2D**. Метки даты P2DOS. Альтернативная форма временных меток файла использована в P2DOS HAJ разработана Ten Brugge. Метки P2D совместимы с метками времени и даты CP/M Plus. Несколько программ ZSDOS, предоставленных ZSDOS, поддерживают этот формат. ZDDOS не поддерживает метки P2D.

**RAM**. (ОЗУ) Оперативная память компьютера. В противоположность постоянной памяти (ПЗУ) может быть считана и записана.

**RSX**. Резидентное системное расширение. Программный модуль, соответствующий стандарту, разработанному Plu\*Perfect Systems для расширения функциональности CP/M 2.2 совместимых операционных систем. Модуль должен быть загружен в верх области резидентных программ ниже процессора консольных команд (CCP).

**System Prompt**. Знакомая строка A>, которая появляется вскоре после того, как выпущены CP/M компьютеры.

**TPA**. Область транзитных программ. Это адресуемое пространство памяти от самого низкого доступного адреса до самого высокого доступного адреса. Обычно она начинается с 100H до базового адреса BDOS (предполагая, что CCP перезаписан), или базового адреса самого нижнего модуля RSX.

**Utility Programs**. (Программные утилиты) В отличие от прикладных программ (см), программные утилиты или утилиты - более короткие программы, такие как программы просмотра каталога, которые принимают единственную команду от пользователя.

**Байт колеса.** (Wheel Byte) Получил свое название от разговорной фразы "Big Wheel,". Байт колеса управляет безопасностью в ZCPR и ZRDOS или ZSDOS. Когда байт устанавливается в ненулевое значение, то пользователь имеет "колесико" и может выполнять команды недоступны другим пользователям.

**Word**. (Слово) В компьютерном контексте, фиксированное число байтов. Для восьми разрядных микрокомпьютеров Слово обычно - два байта или 16 битов.

**Z-System**. Операционная система, которая полностью заменяет CP/M, подставляя ZCPR вместо командного процессора Digital Research и ZRDOS или ZSDOS вместо дисковой операционной системы Digital Research. ZCPR и ZRDOS или ZSDOS дополняют друг друга несколькими способами для повышения производительности.

**ZCPR**. Z80 замена командного процессора. Первоначально разработана усилиями группы "Специальной группы для Микрокомпьютеров" (SIG/M), но в дальнейшем усовершенствована Ричардом Конном {Richard Conn} до версии 3.0 ZCPR и Джей Сейджем {Jay Sage} до версий 3.3 и 3.4.

**ZRL**. Форма файла перемещаемого образа, использующая определенные именованные общие (Common) области. Для ZSDOS файлы этого типа - совместимы с Microsoft .REL файлами, использующими только перемещаемый общий (Common) сегмент "\_BIOS \_".

**SRT** - Step Rate Time (Время шага)

**HUT** - Head Unload Time (Время отвода головки после операции чтения/записи)

**HLT** - Head Load Time (Время между подводом головки к поверхности и началом чтения)

# Библиография B/P BIOS

Данное руководство не должно рассматриваться как полный справочник для Z80 ассемблера, CP/M и ZCPR операционных систем. На эти темы написано много книг, мы приводим здесь только те, которые мы нашли наиболее полезным.

**Информация о CP/M**

1. Waite, Mitchell and Robert Lafore. **Soul of CP/M: How to Use the Hidden Power of Your CP/M System**. Indianapolis: Howard W. Sams & Co., 1983.

Хорошо написанное основное введение в принципы и дизайн CP/M и использование 8080 ассемблеров. Включено много примеров программирования и несколько полезных приложений. Проиллюстрированный, приложения, индекс, 391 стр.

2. Johnson-Laird, Andy. **The Programmer's CP/M Handbook**. Berkeley: Osborne/McGraw-Hill, 1983.

Очень подробное, полное описание структуры CP/M для программистов. Инструкции по использованию накладок, проектирование и отладка новых CP/M системы. Включен полный пример BIOS. Иллюстрированный источник, приложения, указатель, 501 стр.

3. Digital Research. **CP/M Operating System Manual**. Pacific Grove, California: Digital Research, 1982.

Исходная документация Digital Research, разработчика CP/M. Краткое и авторитетное, и написанное техническим языком, это руководство может, главным образом, использоваться в качестве справочника для тех, кто уже понимает CP/M. Включает описание исходных утилит CP/M и скелетного BIOS в качестве примера. Проиллюстрировано, с приложениями, есть индекс, 250 стр.

**Язык ассемблера Z80**

4. Mostek Corporation. **Programming Manual for Z80 Microcomputer**. Publication No. MK78515, MOSTEK Corp., 1977.

"Библия" программирования ассемблера Z80 с подробными описаниями и примерами каждого кода операции.

5. Hitachi America, Ltd. **HD64180 8-Bit High Integration CMOS Microprocessor Data Book**. Publication #U77, Hitachi America, Ltd., 1985. The reference book for the 64180.

Справочник для 64180. Никаких примеры программирования, но очень подробные списки всех инструкций.

6. Zaks, Rodnay. **Programming the Z80**. Berkeley: SYBEX, Inc., 1982.

Очень подробная информация о каждом аспекте микропроцессора Z80 и ассемблера Z80. Раскрыты основные концепции программирования, аппаратная организация Z80, программирование, адресация и методы ввода-вывода. Включены примеры программирования, и более чем 200 страниц информации посвящены самой системе команд Z80. Проиллюстрированный, приложения, индекс, 624 стр.

**Информация о ZCPR, Z-System и вспомогательные материалы**

7. Morgen, Bruce and Jay Sage. **The Z-System User's Gui**de.

Введение в Z-систему для менее технического пользователя.

8. Sage, Jay. **The ZCPR 3.3 User's Guide**.

Руководство по ZCPR версии 3.3, в том числе многих примеров того, как особенности Z-System можно заставить работать. Выделены расширенные возможности обработки команд и безопасности.

9. Conn, Richard. **ZCPR3: The Manual**. New York: New York Zoetrope, 1985.

"Библия" ZCPR3, записанного исходным автором ZCPR. Много информации теперь устарело, но все еще очень полезно. Проиллюстрированный, индекс, 351 стр.

10. Conn, Richard. **ZCPR3: The Libraries**.

Руководство по библиотекам SYSLIB, Z3LIB и VLIB, которые позволяют продвинутым ZCPR3 программистам на ассемблере создавать свои собственные программы для Z-System.

11. Mitchell, Bridger. **Backgrounder ii**. Idyllwild, CA, 1986.

(Доступно у Plu\*Perfect Systems и др.). Руководство для Backgrounder ii, включает много усовершенствованных методов и описание RSX стандарта, разработанного Plu\*Perfect.

12. Bower, Harold; Cotrill, Cameron and Wilson, Carson. **ZSDOS 1.0 User's Guide**. 1988.

Подробная инструкции пользователя для версии ZSDOS без переключения банков памяти, вместе с подробными ссылками. Она необходима, если вы не перешли на ZSDOS2 с переключением банков.

13. Bower, Harold; Cotrill, Cameron and Wilson, Carson. **Programmer's Manual for ZSDOS 1.1**. 1990.

Описание областей данных, интерфейсов и программирования.

# Приложение 1 Справочник функций BIOS

| **№** | **Имя функции** | **Входные параметры** | **Возвращаемые значения** | **Используются** |
| --- | --- | --- | --- | --- |
| 0 | **CBOOT** Холодный старт | Нет | Нет | Все регистры |
| 1 | **WBOOT** Теплый старт | Нет | Нет | Все регистры |
| 2 | **CONST** Опрос состояния консоли | Нет | A=FFH – если готов  00H – если не готов | AF |
| 3 | **CONIN** Ввод символа с консоли | Нет | A=Символ с консоли  (с маской ICFG) | AF |
| 4 | **CONOUT** Вывод символа на консоль | С=Символ на консоль (с маской ICFG) | Нет | AF |
| 5 | **LIST** Вывод символа на печать | С=Список символов (с маской ICFG) | Нет | AF |
| 6 | **AUXOUT** Вывод символа на AUX: | С=Символ для вывода (с маской ICFG) | Нет | AF |
| 7 | **AUXIN** Ввод символа с AUX: | Нет | A=Прочтенный символ (с маской ICFG) | AF |
| 8 | **HOME** Установка головки на 0 дорожку | Нет | <Код состояния> | Все регистры |
| 9 | **SELDSK** Выбор логического диска | C=Номер диска (0…15) | HL=Адрес DPH HL=000H если диск недоступен | Все регистры |
| 10 | **SETTRK** Установка номера дорожки | BC=Номер дорожки | Нет | Нет |
| 11 | **SETSEC** Установка номера сектора | BC=Номер сектора | Нет | Нет |
| 12 | **SETDMA** Установка адреса буфера обмена данных при дисковых операциях | BC=Адрес DMA | Нет | Нет |
| 13 | **READ** Чтение выбранного сектора | Нет | <Код состояния> | Все регистры |
| 14 | **WRITE** Запись выбранного сектора | C=00H Запись данных  01H Запись каталога  02H Запись в новый  блок | <Код состояния> | Все регистры |
| 15 | **LISTST** Опрос состояния устройства печати | Нет | A=FFH – если готов  00H – если не готов | AF |
| 16 | **SECTRN** Преобразование номера сектора | BC=Номер логического  сектора | HL=Номер физического  сектора DE=Адрес таблицы  трансляции | Все регистры |
| **<<< Конец векторов CP/M 2.2 >>>** | | | | |
| 17 | **CONOST** Опрос состояния вывода консоли | Нет | A=0FFH если CON: готова  00H если CON: занята | AF |
| 18 | **AUXIST** Опрос состояния вспомогательного ввода | Нет | A=0FFH если AUXin готов  00H если нет символов | AF |
| 19 | **AUXOST** Опрос состояния вспомогательного вывода | Нет | A=0FFH если AUXout готов  00H если AUXout занят | AF |
| 20 | **DEVTBL** Адрес таблицы устройств IO | Нет | HL--> Таблица символов IO | HL |
| 21 | **DEVINI** Инициализация устройств IO | Нет | Нет | Все регистры |
| 22 | **DRVTBL** Получить адрес таблицы дисков | Нет | HL--> Таблица DPH если Ok (0 записей для No Drv) | HL |
| 23 | < Зарезервировано для **MULTIO** (Чтение/Запись нескольких секторов)> | | | |
| 24 | **FLUSH** Сброс буферов на диски | Нет | <Код состояния> | Все регистры |
| 25 | **MOVE** Перемещение блоков памяти | HL=Адрес источника DE=Адрес назначения BC=Размер области | Нет | Все регистры |
| 26 | **TIME** Получить или установить текущую дату и время | C=0 (Чтение),  1 (Установка) DE--> 6-байт времени | A=1 - Ok, A=0 – ошибка E=Буфер из 6 байт D=1/10th сект. (Чтение) | Все регистры |
| 27 | **SELMEM** Выбрать банк памяти | A=Номер банка | Нет | Нет |
| 28 | S**ETBNK** Выбрать банк памяти для DMA операций | A=Номер банка | Нет | Нет |
| 29 | **XMOVE** Предварительная загрузка банков для перемещения | C= № банка источника  B=№ банка назначения | Нет | Нет |
| **<<< Конец векторов CP/M 3 >>>** | | | | |
| 30 | **RETBIO** Вернуть версию и указатель на область данных BIOS | Нет | A=Номер версии BIOS BC--> Табл. переходов BIOS DE--> Обл. конф. данных  HL--> Табл. конф. устройств | A, BC, DE, HL |
| 31 | **DIRDIO** Выполнить низкоуровневую функцию непосредственно на дискете или SCSI устройствах | <--**Подробности см. ниже для устройства прямого ввода-вывода**--> | | |
| B=Тип драйвера C=Номер функции | <См. таблицу> | <См. таблицу> |
| 32 | **STFARC** Задать номер банка для последующего выполнения функции 33 | A=Номер банка | Нет | Нет |
| 33 | **FRJ**P Выполнить подпрограмму из банка памяти заданного функцией 32 | HL=Адрес выполнения | ?? | ?? |
| 34 | **FRCLR** Возврат из подпрограмм переключения банков памяти при ошибке | HL= Возвращаем. адрес | ?? | Нет |
| 35 | **FRGETB** Загрузить байт по адресу из указанного банка памяти | HL-->Получ. байт C= Номер банка | A=Байт в (C:HL) | AF |
| 36 | **FRGETW** Загрузить слово по адресу из указанного банка памяти | HL-->Получ. слово C= Номер банка | DE=Слово в(C:HL) | AF, DE |
| 37 | **FRPUTB** Сохранить байт по адресу в указанный банк памяти | HL-->Байт назначения C= Номер банка A=Посылаемый байт | Нет (A сохранено в (C:HL)) | F |
| 38 | **FRPUTW** Сохранить слово по адресу в указанный банк памяти | HL-->Слово назнач. C=Номер банка DE=Посылаемое слово | Нет (DE сохранено в (C:HL)) | F |
| 39 | **RETMEM** Вернуть номер текущего банка памяти | Нет | A=Номер текущего банка | AF |
| **Подфункции гибкого диска (Функция 31)** | | | | |
| 0 | **STMODE** Задать режим чтения/записи контроллера гибких дисков | A=0 (DD) A=0FFH (DS) C=0 (№ подфункции) B=1 (гибкий диск) | Нет | AF |
| 1 | **STSIZE** Установить размер, скорость вращения и флаг включения мотора для диска | A=0 (Норм. скорость)  0FFH (Выс.скорость) D=0 (Мотор включен)  0FFH (Упр. мотором) E=0 (Hard), 1 (8"),  2 (5.25"), 3 (3.5") C=1 (№ подфункции) B=1 (Гибкий диск) | Нет | AF |
| 2 | **STHDRV** Установить головку и номер диска для дисковых операций | A=№ диска (биты 0,1)  № головки (бит 2) C=2 (№ подфункции) B=1 (Гибкий диск) | Нет | AF |
| 3 | **STSECT** Задать номер физического сектора, размер и номер последнего сектора на дорожке | A=Номер физ. сектора D=1 Сектор 128 байт,  . . .  3 Сектор 1024 байт) E=Номер посл. сектора C=3 (№ подфункции) B=1 (Гибкий диск) | Нет | AF |
| 4 | **SPEC** Установить скорость шага и время загрузки и выгрузки головки | A=Скорость шага в мс  (B7=1 для 8" Drv) D=Выгр. головки в мс E=Загр. головки в мс C=4 (№ подфункции) B=1 (Гибкий диск) | <Код состояния> | AF |
| 5 | **RECAL** Повторная калибровка привода (перемещает голову к дорожке 0) | C=5 (№ подфункции) B=1 (Гибкий диск) | <Код состояния> | AF |
| 6 | **SEEK** Установить дорожку для дисковых операций и перейти к ней | A=Номер дорожки D=0FFH Проверка E=0 (Один шаг)  <> 0 (Два шага) C=6 (№ подфункции)  B=1 (Гибкий диск) | <Код состояния> | AF |
| 7 | **SREAD** Чтение сектора дискеты | HL--> Буфер чтения C=7 (№ подфункции) B=1 (Гибкий диск) | <Код состояния> | AF, HL |
| 8 | **SWRITE** Запись на дискету | HL--> Буфер записи C=8 (№ подфункции) B=1 (Гибкий диск) | <Код состояния> | AF, HL |
| 9 | **READID** Считать первую корректную информацию ID на дорожке | C=9 (№ подфункции) B=1 (Гибкий диск) | <Код состояния> | AF |
| 10 | **RETDST** Вернуть состояние привода | C=10 (№ подфункции) B=1 (Гибкий диск) | A=Байт статуса BC=Тип контроллера HL--> Байт статуса | AF, BC, HL |
| 11 | **FMTTRK** Выполнить полное форматирование дорожки на одной стороне дискеты | HL--> Формат данных D=Сект. на дорожке E= Число байт межсек-  торного промежу-  тка GAP3 C=11 (№ подфункции) B=1 (Гибкий диск) | <Код состояния> | Все регистры |
| **Подфункции жесткого диска (Функция 31)** | | | | |
| 0 | **HDVALS** Установить адрес области данных пользователя для прямого SCSI IO | DE--> Буфер 512 байт C=0 (№ подфункции) B=2 (Жесткий диск) | A=Число байт в CDB | AF, HL |
| 1 | **HDSLCT** Установить бит физического устройства и поместить номер логического устройства в блок команд SCSI | A=Байт устройства C=0 (№ подфункции) B=2 (Жесткий диск) | A=Биты физических дисков | AF |
| 2 | **DOSCSI** Прямой драйвер SCSI | DE=Команда управл.  диском A=Нет запис. данных C=0 (№ подфункции) B=2 (Жесткий диск) | H=Байт сообщения L=Байт статуса A=Маскированный байт  сообщения | Все регистры |

**Код состояния**: A = 0, бит нуля установлен (Z), если операция выполнена успешно

A <> 0 бит нуля очищен (NZ), если возникли ошибки

# Приложение 2 Справочник функций ZSDOS2

| **№** | **Имя функции** | **Входные параметры** | **Возвращаемые значения** |
| --- | --- | --- | --- |
| 0 | Теплый старт | Нет | Нет |
| 1 | Ввод символа с консоли | Нет | A=Символ |
| 2 | Вывод символа на консоль | E = Символ | A=00H |
| 3 | Ввод символа с логического устройства RDR | Нет | A=Символ |
| 4 | Вывод символа на логическое устройство PUN | E = Символ | A=00H |
| 5 | Вывод символа на логическое устройство LST | E = Символ | A=00H |
| 6 | Прямой ввод/вывод символа с консоли | E = 0FFH (Ввод) | A=Входной символ |
| Е = 0FEH (Ввод) | A=Статус консоли |
| Е = 0FDH (Ввод) | A=Входной символ |
| E=00H..0FCH (Вывод) | A=00H |
| 7 | Получить байт ввода/вывода | Нет | A=Байт I/O (0003H) |
| 8 | Установить байт ввода/вывода | E= Байт I/O | A=00H |
| 9 | Вывод строки | DE=Адрес строки | A=00H |
| 10 | Чтение консольного буфера | DE=Адрес консольного буфера | A=00H |
| 11 | Опрос статуса консоли | Нет | A=00H нет символа  A=01H есть символ |
| 12 | Запрос номера версии | Нет | A=Номер версии (22H) |
| 13 | Сброс дисковой системы | Нет | A=00H No $\*.\* on A  A =FFH $\*.\* on A |
| 14 | Выбор диска | E= Номер диска | A=00H No $\*.\* File  A =FFH $\*.\* File |
| 15 | Открытие файла | DE=Адрес FCB | A=Код каталога |
| 16 | Закрытие файла | DE=Адрес FCB | A=Код каталога |
| 17 | Поиск первого | DE=Адрес FCB | A=Код каталога |
| 18 | Поиск следующего | DE=Адрес FCB | A=Код каталога |
| 19 | Удаление файла | DE=Адрес FCB | A=Код ошибки |
| 20 | Последовательное чтение | DE=Адрес FCB | A=Код чтения/записи |
| 21 | Последовательная запись | DE=Адрес FCB | A=Код чтения/записи |
| 22 | Создание файла | DE=Адрес FCB | A=Код каталога |
| 23 | Переименование файла | DE=Адрес FCB | A=Код ошибки |
| 24 | Получить вектор установленных дисков | Нет | HL=Вектор дисков |
| 25 | Получить номер текущего диска | Нет | A=Текущий диск |
| 26 | Установить адрес буфера обмена | DE=Адрес буфера обмена | A=00H |
| 27 | Получить адрес вектора распределения | Нет | HL=Адрес вектора распределения |
| 28 | Записать ключ защиты диска | Нет | A=00H |
| 29 | Получить вектор R/O дисков | Нет | HL=Вектор R/O |
| 30 | Установить атрибуты файла | DE=Адрес FCB | A=Код ошибки |
| 31 | Получить адрес параметров диска | Нет | HL=Адрес DPB |
| 32 | Установка/получение кода пользователя | E=FFH (получить)  E=Код пользователя (уст) | A= Код пользователя  A=00H |
| 33 | Прямое чтение | DE=Адрес FCB | A=Код чтения/записи |
| 34 | Прямая запись | DE=Адрес FCB | A=Код чтения/записи |
| 35 | Вычислить размер файла | DE=Адрес FCB | A=Код ошибки |
| 36 | Установить номер записи прямого доступа | DE=Адрес FCB | A=00H |
| 37 | Сброс дисков | DE=Маска | A=00H |
| 38 | Не реализована |  |  |
| 39 | Получить вектор несъемного диска | Нет | HL=Вектор диска |
| 40 | Прямая запись на обнуленный блок | DE=Адрес FCB | A=Код чтения/записи |
| 41-44 | Не реализованы |  |  |
| 45 | Установить режим ошибок | E=FFH (Получить)  E=FEH (Get Err/Disp)  E=01H (Уст. ZSDOS)  E=00H (Уст. CP/M) | A=00H  A=00H  A=00H  A=00H |
| 46 | Вернуть свободное место | E=Номер диска | Размер в DMA+0..DMA+3 |
| 47 | Получить адрес DMA | Нет | HL=Текущий адрес DMA |
| 48 | Получить тип DOS | Нет | H=Тип DOS: “S”-ZSDOS  “D”-ZDDOS  L=Номер версии BCD |
| 49 | Вернуть адрес ENV | Нет | HL=Адрес дескриптора ENV |
| 50-97 | Не реализованы |  |  |
| 98 | Получить время | DE=Адрес для разм. Time | A=Код Time/Date |
| 99 | Установить время | DE=Адрес Time | A=Код Time/Date |
| 100 | Получить флаги | Нет | HL=Флаги |
| 101 | Установить флаги | DE=Флаги | Нет |
| 102 | Получить временную метку | DE=Адрес FCB | A=Код Time/Date  Метка в буфере DMA |
| 103 | Установить временную метку | DE=Адрес FCB  Метка в буфере DMA | A=Код Time/Date |
| 104-151 | Не реализованы |  |  |
| 152 | Разбор File Spec | DE=Адрес FCB,  строка в буфере DMA | A=No. если "?" в Fn.Ft  DE=Адрес символа разделителя  FCB+15=0 если разбор Ok,  0FFH если ошибка(и) |
| 153-255 | Не реализованы |  |  |

**Сводка кодов возврата ZSDOS2**

|  |  |
| --- | --- |
| Код каталога | A=00H, 01H, 02H, 03H, если нет ошибки  A= FFH, если ошибка |
| Код ошибки | A=00H, если нет ошибок  A=FFH, если ошибка |
| Код Time/Date | A=01H, если нет ошибок  A=FFH, если ошибка |
| Код чтения/записи | A=00H, если нет ошибок  A=01H Чтение => Конец файла  Запись => Каталог переполнен  A=02H Диск переполнен  A=03H Ошибка при закрытии при чтении/записи в  произвольную запись  A=04H Прочитана пустая запись во время прямого  чтения записи  A=05H Переполнение каталога во время записи  произвольной записи  A=06H Слишком большая запись во время чтения/записи  произвольной записи |
| Расширенные коды ошибок в режиме возврата ошибок: | A=FFH Расширенный флаг ошибки  Н=01H Ошибка ввода/вывода диска (Плохой сектор)  Н=02H диск только для чтения  Н=03H Запись в защищенный файл  Н=04H Недопустимый диск (Выбор) |

# Приложение 3 Спецификации и форматы временных меток

Универсальные метки и форматы времени, используемые в ZSDOS основаны на упакованных BCD цифрах. Было решено, что это самый легкий формат для обработки прикладными программами Z80 и совместимый с большинством часов реального времени. Формат меток и функции часов идентичны форматам DateStamper Plu\*Perfect для этих функций.

Некоторые форматы временных меток файлов (например CP/M Plus типа) не хранят все данные, находящиеся в универсальном формате на диске. Временные меток типа CP/M Plus, не поддерживают время последнего доступа. Подпрограммы интерфейса ZSDOS заполняют нереализованные поля меток нулями, когда используется функция получить временную метку, и игнорируют содержимое неиспользуемых полей, когда используется функция поместить временную метку.

В зависимости от выбранного типа используемых временных меток, формат временных меток на диске может отличаться от универсального формата. Эти различия будут эффективно скрыты от пользователей ZSDOS и процедур временных меток, пока используются функции ZSDOS для получения или манипулирования метками.

**Формат времени** (6 байт упакованных BCD):

|  |  |
| --- | --- |
| TIME+0 | последние 2 цифры года (Префикс 19 предполагается для 78 до 99, еще 20 предполагается) |
| TIME+1 | месяц [1..12] |
| TIME+2 | день [1..31] |
| TIME+3 | часы [0..23] |
| TIME+4 | минуты [0..59] |
| TIME+5 | секунды [0..59] |

**Формат файла меток** (15 байт упакованы BCD):

|  |  |
| --- | --- |
| DMA+0 | Поле создания (первые 5 байт формата времени) |
| DMA+5 | Поле доступа (первые 5 байт формат времени) |
| DMA+10 | Поле изменения (первые 5 байт формата времени) |

# Индекс

!!!TIME&.DAT, 106 Disk Change, 133

Disk Parameter Block (see DPB)

Absolute System Model, 108 Disk Parameter Header (see DPH)

ALIAS, 69 Dos, 64, 146

ALIEN.DAT, 85, 96 DosDisk, 141, 146

ALV, 47, 55, 57, 143 DPB, 55, 83

Application Programs, 145 DPH, 29, 33, 52

Archive, 91, 93

Assembly, 7, 10 EMULATE, 84, 96

AutoSize, 67 Environment Descriptor, 58

Auxiliary, 28

FCB, 146

Bank number, 47, 70 File Attributes, 146

Banked Bios, 9, 11 Files, 10

Base Page, 7 Format, 83

Baud Rate, 72 Function 0 (Bios), 25

BDOS, 109, 145 Function 1 (Bios), 26

BGii, 145 Function 2 (Bios), 26

Bios, 64, 109, 145 Function 3 (Bios), 27

Bit Map, 145 Function 4 (Bios), 27

Boot, 145 Function 5 (Bios), 27

Boot tracks, 14, 17, 18, 71, 145 Function 6 (Bios), 28

BPBUILD, 64 Function 7 (Bios), 28

BPCNFG, 16, 19, 68, 99, 123 Function 8 (Bios), 28

BPDBUG, 77 Function 9 (Bios), 29

BPFORMAT, 83, 96 Function 10 (Bios), 29

BPSWAP, 87 Function 11 (Bios), 29

BPSYSGEN, 16, 18, 89, 98, 115 Function 12 (Bios), 30

Breakpoint, 81, 82 Function 13 (Bios), 30

Byte, 145 Function 14 (Bios), 30

Function 15 (Bios), 30

CCP, 109, 145 Function 16 (Bios), 31

Character Device, 51, 71 Function 17 (Bios), 31

Checksum, 145 Function 18 (Bios), 31

Clock, 34, 70 Function 19 (Bios), 32

Clock driver, 120, 145 Function 20 (Bios), 32

CLOCKS.DAT, 120 Function 21 (Bios), 32

Cold Boot, 68 Function 22 (Bios), 33

Command Script, 145 Function 23 (Bios), 33

COMMON bases, 64 Function 24 (Bios), 33

Common Memory, 8, 11 Function 25 (Bios), 33

Configuration, 68 Function 26 (Bios), 34

Controller types, 83 Function 27 (Bios), 34

COPY, 91 Function 28 (Bios), 34

CP/M, 2 Function 29 (Bios), 35

CPR, 64 Function 30 (Bios), 35

CSV, 47 Function 31 (Bios), 36, 42, 43

Floppy SubFunction 0, 36

Data Structures, 46 Floppy SubFunction 1, 36

DateStamper, 34, 141, 146 Floppy SubFunction 2, 37

DDT, 146 Floppy SubFunction 3, 37

Disassembly, 80 Floppy SubFunction 4, 38

Floppy SubFunction 5, 38 Memory Bank, 34, 78

Floppy SubFunction 6, 39 Memory map, 8

Floppy SubFunction 7, 39 Modules, 10

Floppy SubFunction 8, 40 MOVCPM, 11, 14, 18, 108

Floppy SubFunction 9, 40 MOVxSYS, 18, 115, 117

Floppy SubFunction 10, 41 MOVZSYS, 18, 108

Floppy SubFunction 11, 41

Hard SubFunction 0, 42 Named COMMONs, 64

Hard SubFunction 1, 42 No-Slot-Clock, 126

Hard SubFunction 2, 43 Non-Banked system, 9, 17

Function 32 (Bios), 43 NZBLITZ, 3

Function 33 (Bios), 44 NZCOM, 3, 26

Function 34 (Bios), 44

Function 35 (Bios), 44 Options, 11

Function 36 (Bios), 44

Function 37 (Bios), 45 P2DOS, 141, 146

Function 38 (Bios), 45 PARK, 119, 124

Function 39 (Bios), 45 Parse, 143

Path, 132, 133

Function 46 (Dos), 143 Programming, 25

Function 49 (Dos), 143 PUBlic, 92, 97, 120, 132

Function 99 (Dos), 126

Function 152 (Dos), 143 RAM, 146

RCP, 21, 144

Hard Drives, 98 Read-Only, 132

HDBOOT, 19, 98 REL, 64, 110

HDIAG, 100 RSX, 146

HELP, 144

Hexadecimal, 146 SASI, 42, 50, 74

SCSI, 42, 50, 74

Identifiers, 46 SCSI-2, 74

Image, 19 Sense, 102

Image (IMG) file, 65, 115 SETCLOK, 120

INIRAMD, 104 SHOWHD, 75, 123

INITDIR, 106 SmartWatch, 126

INSTAL12, 18, 108, 115 SPINUP, 119, 124

Installation, 17 Stack Switcher, 44

INSTALOS, 108 Stamp Create, 137

IOBYTE, 4, 26, 27, 28, 31, Stamp Last Access, 136

51, 71, 72, 146 Stamp Modify, 137

IOP, 114 STARTUP, 16, 21, 69, 105, 114,

IOPINIT, 114 120

Step rate, 73

Jump table, 25 Syntax, 2

System Attribute, 133

K, 146 Системный банк, 9, 11

System Prompt, 146

Landing zone, 119, 124

LDSYS, 19, 68, 115 Tailoring, 7, 10, 46

TCAP, 60

MCOPY48, 91 TDD, 126

Memory, 7 TERMCAP, 60

Timer, 49

TPA, 147

Trace, 81

User Bank, 9

Utility Programs, 147

Version, 10, 11, 12

VLIB, 60

Wait states, 49, 70

Warm Boot, 11

Wheel, 85, 135, 139, 147

Word, 147

XBIOS, 3

XD III, 139

XDPB, 55

XDPH, 49, 52

Z-System, 7, 147

Z40, 144

ZCPR, 58, 144, 147

ZRDOS, 2

ZRL, 64, 110, 147

ZSCFG2, 129

ZSCONFIG, 111

ZSDOS, 2, 110

ZSDOS2, 58, 126, 143

ZSYSTEM.MDL, 108

ZXD, 139

# Примечания B/P BIOS

Версия ZSDOS2 (Z-System DOS с переключением банков) и Z4x замена командного процессора с переключением банков разрабатывались на протяжении нескольких лет. Руководство может ссылаться на конкретные версии, или обобщенные имена. По состоянию на весну 2001 года выпуск под Универсальной общественной лицензией GNU, предоставляет две версии ZSDOS2. ZS203.ZRL, которая содержит код для хеширования каталогов и ZS227G который не имеет. Аналогично, последняя версия замены процессора Z4x является Z41.ZRL, которая включает небольшое количество изменений. Новая утилита CONFZ4.COM доступна для этой цели.

Z41.ZRL - последняя версия командного процессора с коммутацией банков памяти на основе ZCPR34. Первое, что вы, вероятно, заметите, что при "теплом" старте после завершения операций, которые перезаписывают командный процессор, система не загружает командный процессор с диска. Копия части процессора, которая находится в области TPA, сохранена в системном банке и просто восстанавливается с помощью функции межбанковского перемещения блоков. Это намного быстрее, чем поиск и чтение с диска!

Z41 объединяет многие из обычно используемых функций RCP и это означает, что вы можете полностью исключить использование резидентного процессора команд (RCP) и сэкономить около 2 кбайт. Встроенная справка с привилегиями колеса (эквивалент “Супер пользователя” в Z-System) выводит следующий список:

CPR

CLS cp DATE DIR ECHO

era feed get go H

jump list NOTE p poke

port prot reg ren res

save sp spop TIME type

VER WHL

Из 27 резидентных команд доступны, только те, что перечислены в заглавными буквами, когда байт колесо выключен, например, при работе в режиме удаленного доступа в системе BBS. Даже среди функций, доступных всем пользователям, есть определенные отключённые функции, например функции настройки даты и времени, которые зарезервированы для тех, кто имеет привилегии колесо. Для тех, кто не знаком с Z-System, эти встроенные функции следующие:

CLS - Очистить экран, используя текущее определение TermCap

cp - Скопировать файл из одного местоположения в другое

DATE - Отобразить дату

DIR - Отображение файлов в текущем или указанном каталоге

ECHO - Распечатать остаток строки на консоль

era - Стереть файл или файлы

feed - Отправьте байт перевода формата бумаги (извлечение)

get - Загрузите файл из текущего каталога до указанного адреса памяти

go - Выполнить код в 100H. Используется, для повторного выполнения команд.

H - Справка. Выводит список доступных команд в CP, RCP и FCP

jump - Выполнение кода по указанному адресу

list - Отправить указанный файл на устройство LST: (как правило, принтер)

NOTE - Просто игнорировать остаток строки

p - Отображение содержимого памяти (быстрый просмотр)

poke - Ввод шестнадцатеричного байт в память по указанному адресу

port - Чтение или запись байта в указанном порту ввода-вывода

prot - Установить/очистить атрибуты системный и только чтение для указанный файлов

reg - Установить, увеличить, уменьшить, очистить 10 регистров ZCPR3

ren - Переименовать файл

res - Сброса дисковой подсистемы и перерегистрация всех дисков

save - Записать память начиная с 100h на заданную длину в файл

sp - Вернуть свободное места на указанном диске

spop - Вытолкнуть значение из верхней части стека оболочки ZCPR3

TIME - Отображение времени (установка если колесо)

type - Отображение файла на консоли

VER - Отображение номеров версий Z41, ZSDOS2 и B/P BIOS

WHL - Переключить байт колеса ON (w/пароль), OFF или отобразить состояние

Этот командный процессор опирается на ZSDOS2 и B/P BIOS с переключением банков памяти и не будет работать без них. Для сведения к минимуму дублирования кода, анализатор командной строки в ZSDOS2, а также другие новые функции используются вместо дискретного кода, используемого в других Замен обработчика команд. В качестве дополнительных улучшений, строка приглашения может показывать время ZSDOS2 и отображение даты может быть настроено для отображения либо в "MM DD,YY" (стиль США) или "DD MM YY" (европейских) форматов. Эти две особенности постоянно "зашитыми" в оригинальном дистрибутиве Z40 несколько лет назад, но с тех пор были сделаны настраиваемыми с помощью утилиты CONFZ4.COM в Z41.

Фактические требования к памяти TPA для Z41 и ZSDOS2 меньше, чем «стандартные» размеры CP/M 2 и 3.5 килобайт соответственно. Для максимально возможного пространства TPA они могут определяться автоматически (опция в утилите BPBUILD). К сожалению, существуют еще некоторые устаревшие программы, которые неправильно вычитают размер "стандартного" BDOS из начального адреса таблицы переходов BIOS в результате неправильного обращения к памяти. Чтобы справиться с этой проблемой, есть опция определить местоположение запуска Z41 и частей ZSDOS2 TPA в "стандартных" расположениях ниже BIOS с мертвым пространством в верхней части заполненным неинформативными символами до следующего элемента системы расположенного выше. Это оказалась адекватным методом решения случайных проблем, если возникает такая необходимость.

Некоторые утилиты, описанные в данном руководстве, также повзрослели за эти годы, и добавились некоторые новые. Некоторые наиболее значимые из них:

**BPCNFG.COM** - был добавлен новый пункт меню в основной экран, позволяющий загружать настройки из файла сценария конфигурации (.CNF). Формат описан в справке.

**HASHINI.COM** - эта утилита готовит новый каталог для доступа к файлам с помощью ZSDOS2 хэш-алгоритм реализованный в ZS203.ZRL. Смотрите встроенную помощь для более подробной информации об использовании.

**SIZERAM.COM** - Сканирует и выдает отчет об использовании памяти в системе с переключением банков. Эта утилита использует функции межбанковских перемещений данных системы BP BIOS с переключением банков памяти и не работает в системе без переключения банков.

**TDD.COM** - Эта утилита обеспечивает большее количество часов, чем DS-1216E, описанная в этом руководстве. Она управляет часами во всех поддерживаемых версиях BP BIOS.

**TURBO.COM** - при использовании процессора Z8S180, эта утилита позволяет вам подключать или отключать схему внутреннего деления на два, в результате удваивая или уменьшая вдвое скорость процессора.

**BPDBUG** - DDT подобный отладчик с поддержкой BP BIOS с переключением банков памяти. Хотя примитивный, он позволяет ковыряться в среде с переключением или без переключения банков памяти.

1. XDPH - расширенный заголовок параметров диска [↑](#footnote-ref-1)