**NULU.DOC**

**Документация для NULU.COM**

**01.11.85**

**Версия 1.5**

На NULU.COM и NULU.DOC авторское право 1984 и 1985 принадлежит Мартину Мюррей {Martin Murray}, почтовый ящик 35972, Даллас, Техас 75235. Оба могут использоваться свободно для некоммерческих целей, но ни один не может продаваться, включаться в пакет для продажи или использоваться в качестве стимула купить, любым человеком, организацией или корпорацией без предварительной договоренности с правообладателем, Мартином Мюррей. Кроме того, Мартин Мюррей не несет никакой ответственности за убытки, возникшие в результате использования или невозможности использовать эту программу. Программа NULU.COM не может распространяться без файла документации NULU.DOC, и при этом сообщения об авторском праве не могут быть удалены ни из одного файла, не может быть предотвращено отображение их на экране.

# Благодарности

Эта программа была бы невозможна без изящной работы Гэри Новосельски {Gary P. Novosielski} и тех, кто работал над структурой библиотеки LU.

Кроме того, я хотел бы поблагодарить людей, которые поддерживали разработку NULU в различных формах от предложений до финансовых взносов. Все комментарии (хорошие, плохие и катастрофические) приветствуются. Особая благодарность Кейту У. Антклиффу {Keith W. Antcliff} из Мида, штат Вашингтон за разработку им NULUTERM.ASM, наложения заплатки, и Рону Фаулеру {Ron Fowler} из Nightowl Software за его любезное разрешение включить MLOAD23.COM в выпуски библиотеки NULU.

Почести за смелость переходят к Деннису Рекла {Dennis Recla}, Норму Грегори {Norm Gregory} и Джиму Свитцу {Jim Switz} за их мужество "рискнуть всем", и высказанные пожелания и указанные ошибки при тестировании NULU.

# Введение

NULU является полной заменой LU и LSWEEP. При размере 15k, она включает в себя все функции обоих программ, и еще некоторые. Ее функции:

* Два режима работы, каждый с полным меню
* Поддержка проверки и CRC в стиле LU
* Просмотр и печать элементов и дисковых файлов, распакованных при необходимости
* Извлечение файлов-элементов с возможностью распаковки при необходимости
* Более крупные библиотеки, до 800 элементов или более в зависимости от доступной памяти
* Автоматическая сортировка элементов, все элементы всегда отсортированы
* Более быстрая перестройка
* Расширенная обработка ошибок
* Обработка диска с подкачкой
* Автоматическая самопроверка
* Доступ к удаленным элементам
* Доступ к библиотекам только для чтения
* Обработка библиотеки с использованием подстановочных символов
* Дополнительный патч только для чтения
* Поддержка CP/M Plus
* Общая совместимость с синтаксисом LU и NSWEEP
* Многочисленные пользовательские патчи

# Различия между NULU 1.5 и NULU 1.1

NULU 1.5 включает несколько существенных изменений, не в последнюю очередь которые вызваны исправлениями ошибок. В версии 1.1 было несколько ошибок, включая случайное неправильное извлечение больших файлов и затирание байтов в именах элементов записи каталога. Все известные ошибки исправлены в новом выпуске. (На самом деле "ошибка" извлечения происходила из-за одной из ошибок в CP/M приводящей к необычному эффекту при комбинации прямого и последовательного доступа к файлу. Если вы посмотрите внимательно, то увидите, что BDOS повторно выделяет используемые блоки диска, в порядке убывания, не больше не меньше!)

При попытке внести изменения в файл библиотеки, который логически установлен только для чтения, NULU отобразит сообщение:

**Library is READ-ONLY. Modify anyway? (y/n)**

и будет ждать от вас, ввода yes (да) или no (нет). Если вы ответите "no", то сделанная вами попытка обработки будет отменена. Если вы ответите "yes", вы сможете внести изменения и когда закроете библиотеку, файл снова будет только для чтения.

Имеются и другие, более видимые изменения. Дисковый интерфейс был расширен с тем, чтобы некоторые команды, которые раньше применялись только к файлам-элементам библиотек, могли теперь также работать с дисковыми файлами. Эти команды включают в себя: -D, -P, -V и новую команду -S, которая выполняет поиск ограниченной строки. Например, чтобы удалить элемент библиотеки команда прежняя:

**-d filespec**

но вы можете также удалить дисковые файлы:

**-d du:filespec**

где комбинация du: (диск/пользователь) сообщает NULU, что вместо элементов библиотеки вы подразумеваете дисковые файлы. (Обратите внимание, что команда -Z была переназначена). Этот синтаксис, также, применяется к другим упомянутым выше командам.

Кроме того, я приложил усилие, чтобы привести NULU 1.5 к большему соответствию с синтаксисом и работой LU, особенно LU310.COM. Поэтому, в дополнение к другим изменениям, пользователь обнаружит, что команда -M для команды NULU - Меню была изменена на -H для отображения Справки.

# NULU

Насколько мне известно, NULU будет работать на любой системе CP/M 2.x или выше, однако она была протестирована мной только на компьютерах Osborne 1 и Osborne Executive. Она написана на ассемблере Z80, совместимом с 8080.

Полное обсуждение сущности и преимуществ файлов библиотек выходит за рамки моей выносливости, однако краткое изложение уместно.

Файл библиотеки - один файл CP/M, который содержит 1 или несколько меньших файлов в качестве его элементов. Каждый файл записан в библиотеку и зарегистрирован в каталоге библиотеки. После этого к нему можно получить доступ для просмотра, печати или выполнения, или он может быть извлечен из библиотеки, чтобы снова принять состояние автономного файла.

Причин для этого много. Например, в CP/M каждый файл занимает минимальный объем дискового пространства. Этот объем может быть в диапазоне от 1k до 16k, в зависимости от системы. Даже если файл содержит только 1 байт, он все равно займет минимально выделяемый объем. Остальная часть пространства будет потрачена впустую. Помещение этих файлов в библиотеку минимизирует возможные потери до 127 байт.

Также, каждый файл занимает запись каталога. В большинстве систем с гибкими дисками максимальное количество записей файлов на диске равняется 64. Включение файла в библиотеку освобождает ранее занимаемое им место в каталоге для другого файла.

Кроме того, использование файлов библиотеки может упростить процесс систематизации, включив несколько файлов аналогичных типов или объектов в один файл.

Тем не менее, есть и опасности, связанные с использованием библиотек. Главной из них является то, что если файл библиотеки потерян или поврежден, все файлы-элементы также теряются. Решение этой проблемы можно суммировать в одном слове: BACKUP. Программа NULU, вероятно, не является совершенной, и все мы знаем, что дисковые системы тоже, поэтому сделайте резервную копию важных библиотек, пожалуйста. Я не хочу слышать печальные истории.

Почти так же важно, как резервное копирование библиотеки, является здравый смысл при принятии решения, что поместить в них. Давайте посмотрим правде в глаза, если файл находится в библиотеке, к нему не так просто добраться. Если вы должны добираться к файлу каждые 15 минут, вероятно, он не должно быть в библиотеке, за исключением резервного копирования.

# Небольшое отступление относительно спецификаций файлов

При определении спецификации файла, подстановочные символы могут использоваться свободно, однако, в отличие от большинства программ, NULU не настаивают на том, что "." должен отделять имя файла от типа файла. Например, в NULU спецификация файла "\*\*" означает то же самое, что "\*.\*". С другой стороны, спецификация файла "FRED.TXT" должна быть введена точно так же, как это. "FREDTXT" заставил бы NULU думать, что тип файла не был определен. Правило простое: используйте точку в любое время, как вам нравится, но она обязательно необходима, только когда это требуется для ясности.

При определении диска и/или пользовательской области синтаксис очень либерален. "A15:\*\*", означает то же что "15/A:\*.\*", как делает "1A5:\*\*". Двоеточие абсолютно необходимо. Если требуется другая область пользователя для диска по умолчанию, спецификатор диска может быть опущен, как в "15:\*\*", так же, как область пользователя может быть опущена как в "A:\*\*".

Если спецификация диска/пользователя сделана, но спецификация файла не обозначена, генерируется спецификация файла ????????.???. Поэтому, чтобы указать диск по умолчанию, текущую область пользователя и спецификацию файла ????????.???, просто введите двоеточие само по себе (например, ":") !

# Функционирование NULU

При загрузке NULU автоматически проверяет собственную CRC. Этот процесс приводит к задержке в 2.5 секунды на моей машине 4 МГц. Однако из-за безопасности, которую предоставляет эта функция, я надеюсь, что вы найдете, что это утешительная пауза. Обратите внимание, что CRC не зависит от нормальных пользователей патчей.

NULU может быть вызван с или без хвоста команды. Хвост команды может состоять из любой комбинации допустимых операторов NULU до 128 символов (ограничение, наложенное CCP. После запуска NULU, командные строки могут быть длиной до 254 символов).

Как и в LU, всем командам NULU предшествует тире. Это - копия меню командного режима работы:

-A Add members -B Brief toggle

-C Close the library -D Delete members/files

-E Extract members -F Filesweep mode

-G Get filespec -H Help

-K Krunch library -L List members

-N Rename members -O Open a library

-P Print members/files -Q Unsqueeze members

-R Replace members -S String search

-T Replace/Add members -U Drive/User change

-V View members/files -W Wildcard open

-X Exit NULU -Y Disk directory

-< Redirect input -> Redirect output

Команды могут быть представлены в виде одной строки, каждая из которых завершается пробелом. Они обрабатываются слева направо. Все символы преобразуются в верхний регистр.

Диск/область пользователя по умолчанию выводятся на экран вместе с именем текущего командного режима каждый раз, когда консоль запрашивает ввод.

Команды будут описаны в порядке их появления выше. Все элементы в [квадратных скобках] указывают дополнительные параметры. Слово "**filename**" (имя файла) относится к однозначной спецификации файла, в то время как термин "**filespec**" (спецификация файла) относится к потенциально неоднозначной спецификации файла. Параметры спецификации файла, заключенные в (круглые) скобки, указывают ссылки на элементы удаленных файлов. Три точки ("...") после параметра указывают бесконечное повторение.

Во время выполнения большинства команд, которые получают неоднозначную спецификацию файла в качестве параметра, ввод ^C в консоли, прервет текущий процесс.

# Команды

**-A** **Добавление элементов**

Синтаксис: **-a filespec[ filespec...]**

Используйте эту команду для добавления файлов с диска в текущую библиотеку. NULU выполнит ряд проходов через каталог, добавляя файлы, по мере их нахождения, до исчерпания соответствующего списка. Если сопоставленное имя файла уже существует в качестве активного элемента в библиотеке, его имя будет выведено на экран, и файл не будет добавлен.

При добавлении всех файлов и операциях замены, если удаленная запись имеет идентичный размер и может быть использована, то будут использоваться дисковое пространство и запись каталога, вместо выделения нового места для файла. Это означает, что реже будет возникать необходимость в освобождении места на диске.

Для каждого добавленного файла будет сгенерирован CRC в стиле LU.

**-B Краткий переключатель**

Синтаксис: **-b**

Переключает режим вывода запроса. Дистрибутивная версия NULU настроена для печати полного имени подсказки текущего режима, как в "-Add members A0:>". Если BRIEF установлен в состояние ON (включен) пользователь просто увидит "-A A0:>". NULU может быть навсегда пропатчен для присвоения BRIEF состояния ON или OFF. См [*Модификация NULU*](#_Модификация_NULU).

**-C Закрытие библиотеки**

Синтаксис: **-c**

Эта команда закрывает текущую библиотеку, записывая ее каталог на диск, если какие-либо изменения были внесены в каталог. Каталог библиотеки никогда не записывается, за исключением случаев закрытия библиотеки, так что не забудьте сделать это. Если вы забыли сделать это и удалили диск, NULU снова предложит вам вставить диск и попытается восстановиться.

Некоторые другие операции, приводящие к закрытию текущей библиотеки:

**-k**, **-o**, **-w**, **-x**

**-D Удаление элементов/файлов**

Синтаксис: **-d filespec[ filespec (filespec)...]**

Синтаксис: **-d du:filespec[ du:filespec...]**

Если присутствует спецификация диска/пользователя, функция удаления применится к файлам на диске, иначе она применяется к файлам элементам библиотеки.

Элементам файлов, соответствующих данной спецификации файла, в каталоге библиотеки будет присвоено состояние удаленного, кроме тех случаев, когда спецификация файла включена в круглые скобки. В этом случае соответствующие удаленные элементы получат активное состояние. Т.е. они будут восстановлены после удаления. Если имя файла соответствующего удаленного элемента файла, уже существует в качестве активного элемента, имя файла будет выведено на экран, и файл не будет восстановлен после удаления.

Если элемент удаляемого файла имеет размер равный 0 секторов, его запись будет удалена из каталога библиотеки полностью, создавая тем самым новый, свободный вход.

**-E Извлечение элементов**

Синтаксис: **-e filespec[=newfilespec filespec du:filespec...]**

Используйте эту команду для извлечения файлов активных элементов. Если требуется извлечение в текущую область пользователя текущего диска, никакого дополнительного синтаксиса не требуется. Однако, для указания другого диска, определение диска/области пользователя назначения может быть включено в исходную спецификацию файла. Пример:

**-e a5:\*\***

извлечет все активные элементы на диск A в область пользователя 5. Файлы могут быть переименованы, а также перенаправлены, указав спецификацию файла вместо или вместе с пользовательской спецификацией диска после знака равенства. Примеры:

1. **-e \*asm=\*bak**

2. **-e \*asm=5:\*txt**

3. **-e fred.txt=sam.txt jane.inf=c8:girls.dbf**

Примеры привели бы к следующим результатам:

1. Извлекает все файлы с типом .ASM на диск/область пользователя по умолчанию, переименовывая их с типом файла .BAK.
2. Извлекает все файлы типа .ASM в область пользователя 5 на диск по умолчанию, переименовывая каждый с типом файла .TXT.
3. Извлекает элемент-файл FRED.TXT на диску/область пользователя по умолчанию под именем SAM.TXT и извлекает элемент-файл JANE.INF в область пользователя 8 на диске C: с именем файла GIRLS.DBF.

Необходимо помнить два правила перенаправления: 1) если введена спецификация файла назначения, это не может быть более неоднозначной, чем исходная спецификация файла, т.е., "\*asm=\*bak" допустима, в то время как "\*\*=\*bak" не допустима; и 2) любая спецификация файла после знака равенства имеет приоритет над любой спецификацией диска/области пользователя исходной спецификации файла, т.е., "-e 6:\*asm=\*bak" извлечет все элементы файлов на текущий диску/область пользователя, переименовав их, и проигнорирует спецификацию "6:".

Во время извлечения для каждого файла генерируется CRC. После закрытия файла назначения, сгенерированный CRC сравнивается с CRC для этого элемента, сохраненного в каталоге библиотеки, если его значение ненулевое. Если результаты не идентичны, будет выведено сообщение об ошибке, до обработки следующего элемента файла или спецификации файла. При возникновении этой ошибки файл назначения не удаляется с диска.

**-F Режим filesweep**

Синтаксис: **-f**

Эта команда переводит NULU во второй режим работы: режим filesweep. Он позволяет пользователю перемещаться по каталогу файлов активных элементов, как если бы они были отдельными файлами, исследуемыми программой типа NSWEEP. Список команд в режиме filesweep выглядит следующим образом:

A Next member B Previous member

C Close library D Delete member

E Extract member F Find member

L Log new drive/user M Mass operations

O Open new library P Print member

Q Unsqueeze member R Rename member

T Tag member U Untag member

V View member W Wildcard rename

X Exit NULU Y Disk directory

Z NULU command mode ? Help!

Из-за сильного сходства между этими командами и командами NULU в командном режиме (command mode), приводится только краткое описание каждой команды.

|  |  |  |
| --- | --- | --- |
| **A** | - | **Переход к следующему элементу** (клавиша "пробел", cr или lf приведет к тому же результату) |
| **B** | - | **Назад к предыдущему элементу** |
| **C** | - | **Закрыть текущую библиотеку** |
| **D** | - | **Удалить файл** **текущего элемента** |
| **E** | - | **Извлечь файл текущего элемента** (подсказка позволяет перенаправление) |
| **F** | - | **Поиск первого элемента**, соответствующего входной спецификации файла |
| **L** | - | **Смена диска/пользователя по умолчанию** (указатель файла возвращается в верхнюю часть списка файлов и снимает пометку со всех элементов файлов.) |
| **M** | - | **Массовые операции для всех помеченных файлов** (позволяет удаление, извлечение, печать, распаковку или просмотр) |
| **O** | - | **Открыть новую библиотеку** (закрывает текущую библиотеку) |
| **P** | - | **Вывод файла текущего элемента на LST:**  (распаковывает в случае необходимости) |
| **Q** | - | **Извлечение файла текущего элемента**, распаковывая, если необходимо (подсказка позволяет перенаправление) |
| **R** | - | **Переименование файла текущего элемента** |
| **T** | - | **Пометка текущего элемента** |
| **U** | - | **Снятие пометки с текущего элемента** |
| **V** | - | **Извлечение файла текущего элемента** (распаковывая, если необходимо) |
| **W** | - | **Подстановочное переименование** (подсказка для oldname и newname) |
| **X** | - | **Выход из NULU** (закрытие всех файлов) |
| **Y** | - | **Получение дискового каталог диска/пользователя по умолчанию** (Возвращает указатель файла в верхнюю часть списка файлов и снимает пометку со всех файловых элементов) |
| **Z** | - | **Возврат в командный режим NULU**  (текущая библиотека остается открытой) |
| **?** | - | **Печать меню режима filesweep** |

Когда введен режим filesweep или после того, как библиотека была закрыта, в данный момент нет открытых библиотек, режим filesweep сообщит:

**No library open.**

и будет принимать только следующие команды:

**L**, **O**, **X**, **Y**, **Z**, **?**

Аналогично, если библиотека открыта, но имеет только каталог без других активных элементов, выводится сообщение

**No member files.**

и будут приняты только упомянутые выше команды

Во время работы filesweep каждый файл будет перечислен в порядке, в котором он найден в каталоге, вместе с размером в килобайтах, который файл будет занимать при извлечении на диск по умолчанию.

Если режим filesweep завершается возвратом в командный режим NULU, будут выполняться любые команды, которые следовали за командой -F в предыдущей командной строке NULU.

**-G Получить спецификацию файла**

Синтаксис: **-g filespec**

NULU будет искать указанную спецификацию файла. Если она найдена, обработка продолжается. В противном случае пользователю будет предложено вставить диск, содержащий эту спецификацию файла. Затем диск сбрасывается и осуществляется повторный поиск. Программа выполняет запросы бесконечно, пока она не получит надлежащую спецификацию файла или пока не будет введен ^C, вынудив NULU продолжать выполнение без указанной спецификации файла.

Эта команда может быть полезна при попытке управлять NULU с помощью утилиты инициализации выполнения программ, например, SUBMIT.COM DRI. Например, можно ввести:

**nulu -o a:asm -g b5:-work.005 -e b10: -g a0:-5.005 -x**

После загрузки NULU откроет библиотеку под названием ASM.LBR на диске A: в текущей области пользователя. Далее он будет искать в пользовательскую область пользователя 5 на диске B: спецификацию файла WORK.005, пока она не будет найдена. Затем все файлы активных элементов будут извлечены в область пользователя 10 на диске B:. Наконец, NULU осуществит поиск в области пользователя 0 на диска A: поскольку спецификация файла с именем 5.005, пока не найдена. После чего NULU завершит работу. Заметьте здесь, что ASM.LBR не должна быть закрыта перед поиском заключительной спецификации файла, потому что в каталог библиотеки изменения не вносились. Если изменения были сделаны, после того, как новый файл был загружен, NULU потребовал бы диск с ASM.LBR назад, таким образом, это могло обновить каталог. Поэтому операция GET будет фактически сведена на нет.

**-H Справка**

Синтаксис: **-h**

Печать меню командного режима.

**-K Уплотнение библиотеки**

Синтаксис: **-k[ < количество разрешенных записей]**

Несмотря на то, что все элементы всегда находятся в алфавитном порядке, при удалении элемента дисковое пространство, которое он занимает не освобождается для операционной системы до тех пор, пока не выполняется команда KRUNCH (уплотнения). Уплотнение состоит из следующих шагов:

1. Если файл с таким же именем, как библиотека уже существует на диске/области пользователя по умолчанию, **и файл библиотеки происходит из другого диска/пользователя**, будет выведено сообщение, указывающее, что файл существует, и команда уплотнения (KRUNCH) будет пропущена. Вы должны удалить мешающий файл или изменить диск/ область пользователя, прежде чем сможет быть осуществлено уплотнение.
2. Библиотека закрыта.
3. На диске/области пользователя по умолчанию открывается новая библиотека под названием WORK-LBR.$$$ и пользователю предлагается ввести число файлов, которые она должна содержать. Если пользователь хочет иметь достаточное число записей для содержания только активных элементов, может быть введено число 1 или любое число, меньше или равное текущему числу активных записей. Если введен необязательный параметр, упомянутый выше, он будет использоваться в качестве количества записей. (Здесь можно прервать процесс уплотнения, введя RETURN или 0. Обратите внимание, что каталог библиотеки считается как запись и автоматически будет учитываться NULU.)
4. После открытия WORK-LBR.$$$, файлы активных элементов из исходной библиотеки будут скопированы один за другим в новую библиотеку. Подпрограмма копии использует рекурсивный процесс, чтобы скопировать как можно больше файлов за один раз. При копировании для каждого элемента вычисляется CRC. Если это число не соответствует CRC, записанному в исходном каталоге библиотеки, после рассматриваемого имени файла появится сообщение:

**CRC ERROR-Continue? (y/n)**

Ответ "N" приведет к удалению рабочей библиотеки и повторному открытию исходной библиотеки. В противном случае, вновь созданный CRC файла будет храниться в каталоге новой библиотеки.

1. Старая библиотека удаляется, и WORK-LBR.$$$ переименовывается с именем старой библиотеки.

Примеры:

1. **-k**

2. **-k <1**

3. **-u b5: -k <63**

4. **-u b5: -k <1 -u a0: -k <63**

Приведенные выше примеры приведут к следующим результатам:

1. Библиотека будет уплотнена на диск/область пользователя по умолчанию. Пользователю будет предложено ввести число разрешенных записей.
2. Библиотека будет уплотнена на диск/область пользователя с числом записей достаточным для содержания всех активных в данный момент элементов файлов.
3. До начала уплотнения изменяется диск/область пользователя по умолчанию на B5:. Затем библиотека уплотняется с 63 разрешенными записями.
4. До начала уплотнения диск/область пользователя по умолчанию изменяется на B5:. Уплотнение начинается с разрешенным количеством записей, достаточным для содержания только всех активных элементов. Диска/область пользователя изменяется на A0:. Затем библиотека снова уплотняется, разрешая 63 записи.

**-L Список элементов**

Синтаксис: **-l[ filespec]**

Синтаксис: **-l[ (filespec)]**

Используйте эту команду для перечисления содержимого каталога библиотеки. Каждое имя элемента файла будет распечатано с последующим стартовым относительным номером сектора в файле библиотеки, размер элемента в секторах, размер в килобайтах, которые файл будет занимать при извлечении на диск/область пользователя по умолчанию и CRC файла. И, наконец, повторно распечатываются размеры всех перечисленных элементов файлов, вместе с количеством секторов, занятых удаленными файлами.

Упомянутые выше параметры могут использоваться для выбора отображаемых файлов. Примеры:

1. **-l**

2. **-l \*\***

3. **-l (\*\*)**

4. -**l \*asm**

5. -l **(fred.txt)**

Вышеупомянутые примеры приводят к следующим результатам:

1. Перечисляются все активные элементы файлов.
2. Перечисляются все активные элементы файлов.
3. Перечисляются все удаленные элементы файлов.
4. Перечисляются все активные элементы файлов, соответствующие спецификации файла ????????.ASM.
5. Перечисляются все удаленные элементы файлов с названием FRED.TXT. (Обратите внимание, что может быть несколько удаленных файлов с тем же именем.)

**-N Переименование элементов**

Синтаксис: **-n oldfilespec=newfilespec[...]**

Синтаксис: **-n (oldfilespec)=newfilespec[...]**

Эта команда используется для переименования активных или удаленных элементов файлов. Подстановочные знаки полностью поддерживаются, но newfilespec не может быть менее неоднозначной, чем oldfilespec. Удаленные элементы могут быть переименованы, заключив oldfilespec только в круглые скобки.

Примеры:

1. **-n \*asm=\*bak**

2. **-n (fred.txt)=sam.txt**

3. **-n \*\*=\*\***

Показанные выше примеры приводят к следующим результатам:

1. Все активные элементы, соответствующие спецификации файла ????????.ASM переименованы с типом файла .BAK.
2. Первый удаленный элемент файла под названием FRED.TXT переименован в SAM.TXT. Все другие элементы, соответствующие этой спецификации файла, перечислены вместе с сообщением о том, что они не могут быть переименованы.
3. Все файлы активные элементы файлов перечислены, каждый с сообщением, указывающим, что они не могли быть переименованы, так как имя уже существует.

**-O Открытие библиотеки**

Синтаксис: **-o filename[ <количество разрешенных записей]**

За исключением команды filesweep открытия библиотеки (Open Library), это - единственный способ открыть или создать библиотеку. NULU будет искать обозначенное имя файла. Если оно будет найдено, она будет открыта и появится сообщение с указанием размера буфера передачи данных. Это количество секторов в памяти, которое NULU будет использовать при извлечении, добавлении или копировании элементов файлов.

Если файл не найден, пользователю предложат задать количество записей разрешенное в новой библиотеке. Если будет передан необязательный параметр, указанный выше, число будет использоваться для определения размера каталога библиотеки. Все каталоги могут содержать некоторое кратное 4 число записей. Все введенные числа будут округлены до ближайшего числа, кратного 4. Помните, что сам каталог рассчитывает как одна запись и что NULU автоматически выделит место для него.

Передаваемое имя файла должно быть однозначным.

**Примечание по размеру каталога:**

В зависимости от доступной памяти (теоретически) могут быть открыты библиотеки ориентировочно с 800 записями. Но следите за размером буфера данных! Если он меньше 9 секторов, вы не сможете распаковать любые файлы. С пользовательским TPA 58K я был в состоянии открыть библиотеки с общим количеством элементов превышающим 584 элемента и с размером буфера 9 секторов. (См. **-<** и как она влияет на память библиотеки.)

**-P Печать элементов/файлов**

Синтаксис: **-p filespec[ filespec (filespec)...]**

Синтаксис: **-p du:filespec[ du:filespec...]**

Если присутствует спецификация диска/пользователя, функция печати применяется к файлам на диске, иначе она применяется к элементам файлов библиотеки.

С помощью этой команды содержимое ASCII файлов, соответствующих заданной спецификации файла, будет выведено на устройство печати. Если файл сжат, NULU распакует его. Удаленные элементы библиотеки могут быть перечислены, заключив надлежащую спецификацию файла в круглые скобки.

**-Q Распаковка элементов**

Синтаксис: **-q filespec[=newfilespec filespec...]**

Эта команда идентична в работе и синтаксисе команде извлечения (EXTRACT), за исключением того, что соответствующие спецификации сжатые элементы будут распакованы.

**-R Замена элементов**

Синтаксис: **-r filespec[ filespec...]**

Эта команда идентична в работе и синтаксисе команде добавления (ADD), за исключением того, что соответствующие файлы будут добавлены в библиотеку, **только если** они уже существуют в библиотеке. Они будут удалены, затем будут добавлены новые файлы.

**-S Поиск строки**

Синтаксис: **-s filespec "строка для поиска"**

Синтаксис: **-s du:filespec "строка для поиска"**

Если присутствует спецификация диска/пользователя, функция поиска применяется к файлам на диске, иначе она относится к элементам файлов библиотеки.

Команда позволяет осуществить **простой** поиск заданной строки в группе файлов. Если "строка для поиска" найдена, вы будете уведомлены. Перечисляются все просмотренные файлы. Отдельная строка, на которой был обнаружен соответствующим элемент, не отображается, так как поиск соответствия не выполняется на основе строки. Т.е. вы можете искать в .COM файлах или сжатые .OVR файлах так же легко как в текстовых файлах. Однако, по общему признанию реализация поиска довольно слаба, в данное время управляющие символы и символы табуляции вырезаются из командной строки, прежде чем функция поиска будет видеть команду. Наслаждайтесь тем что имеется, я буду улучшать его позже.

**-T Замена/добавление элементов**

Синтаксис: **-t**

Эта команда является комбинацией добавления (ADD) и команды замены (REPLACE). Если файлы отсутствуют в библиотеке, они будут добавлены. Если существуют, они будут заменены.

**-U Смена диска/пользователя**

Синтаксис: **-u new drive/user**

Используйте эту команду для изменения диска/области пользователя по умолчанию. Двоеточие (":"), должно следовать за спецификацией диска/пользователя. Например:

**-u a5:**

переключит диск/область пользователя по умолчанию на A5:.

**-V Просмотр элементов/файлов**

Синтаксис: **-v filespec[ filespec (filespec)...]**

Синтаксис: **-v du:filespec[ du:filespec...]**

Если присутствует спецификация диска/пользователя, функция просмотра (VIEW) применится к файлам на диске, иначе она применяется к элементам файлов библиотеки.

Эта команда выводит содержание всех ASCII файлов, соответствующих заданной спецификации файла на консоли. Удаленные элементы библиотеки могут быть обозначены, заключив спецификацию файла в круглые скобки. Если файлы будут сжаты, они будут распакованы. На каждой странице NULU принимает одну из следующих команд:

1. **^X** пропускает вывод файла до следующего, соответствующего спецификации
2. **^C** прерывает последующий вывод
3. **L** вывод одной строки
4. **CR**, **пробел** или **LF** вывод другой полной страницы

В конце каждого файла выводится сообщение

**"Press RETURN"**

и NULU будет ожидать нажатия клавиши прежде, чем перейти к следующему файлу соответствующему спецификации.

**-W Неоднозначное открытие**

Синтаксис: **-w filespec commands to process**

Только для опытных пользователей.

Это одна из самых мощных, и возможно самая опасная из всех команд NULU. Она позволяет пользователю определить группу библиотек, на которые можно воздействовать с помощью ряда команд. Простой пример:

**-w a5:\*lbr -l**

откроет все найденные файлы библиотек на диске A в области пользователя 5, и перечислит содержание каждой. Это будет происходить последовательно для всех файлов в порядке, в котором они найдены в каталоге. Обратите внимание, что эта команда имеет тот же эффект, что и команда **-O**, за исключением того, что она обрабатывает подстановочные спецификации файла библиотеки и что управление пользователя каждой открытой библиотекой ограничено одной строкой, которая вызывает команду **-W**. Т.е. все команды, применяющиеся к каждой открытой библиотеке, должен помещаться в одной командной строке.

Причиной появление этой команды является то, что некоторые команды, которые раньше имели ограничения для перенаправления .NCF файлов теперь доступны на командном уровне. Рассмотрим этот пример:

**-w \*lbr -k <5 -p \*\* -l -' "hit enter for next file " -:**

Эта команда приведет к следующему для каждого соответствующего спецификации файла:

1. Библиотека уплотняется до 5 элементов.
2. Все элементы файлов отправляются на LST:.
3. Печать каталога библиотеки.
4. Подсказка, "PRESS ENTER FOR NEXT FILE " (Нажмите ENTER для следующего файла) отображается в CON:
5. NULU ждет, пока пользователь нажмет RETURN перед передачей управления обратно к -W, которая затем сделает то же самое для следующей библиотеки.

Будьте осторожны при использовании данной команды. Она является новой даже для меня, но я считаю ее чрезвычайно полезной. Мне очень нравится эта:

**-w \*lbr -f**

которая позволяет мне просматривать большое количество библиотек с помощью filesweep, без необходимости знать их имена.

Я пробовал подобную этой:

**-w \*lbr -< command.ncf**

но с очень неутешительным результатом. Она не работает. Я буду работать над этим позже, если появится возможность.

**-X Выход из NULU**

Синтаксис: **-x**

Эта команда устанавливает флаг, указывающий, что, текущая командная строка NULU исчерпана, NULU должен завершиться. Но команда на самом деле является переключателем. Если она введена в строке дважды, второй ввод инвертирует первый. Обратите внимание, что поддержка объединения команд в цепочку была удалена.

Примеры:

1. **-x**

2. **-x -x**

Вышеупомянутые примеры приведут к следующим результатам:

1. Флаг завершения будет установлен, и NULU завершится, когда текущая командная строка будет исчерпана.
2. Флаг завершения установлен и затем сброшен. NULU не завершается.

**-Y Каталог диска**

Синтаксис: **-y[ filespec]**

Эта команда распечатает несортированный каталог диска, обозначенный переданным параметром, или полный список, если параметр отсутствует. Объем каталога может быть ограничен, указав надлежащую неоднозначную спецификацию файла. Если определены строки с инверсным изображением или пониженной яркости, NULU будет использовать эти характеристики, чтобы указать, какие конкретные биты атрибутов файла установлены в каждом имени файла.

**-Z Удаление файлов на диске**

Синтаксис: **-z filespec**

Используйте эту команду для удаления файлов на диске во время работы NULU. Каждый файл, соответствующий заданной неоднозначной спецификации файла, будет перечислен и удален. Библиотеки могут быть удалены, так же легко, как и любой другой тип файла, так что будьте осторожны. Единственная имеющаяся защита состоит в том, что открытая в данный момент библиотека, если имеется, никогда не будет удалена.

**-< Перенаправление ввода**

Синтаксис: **-< filename**

Эта команда заставит NULU открыть файл с указанным именем и начнет принимать из него команды вместо получения их из консоли. Файлы этого типа фактически представляют собой, командный файл NULU. По умолчанию тип файла примет значение ".NCF", если не определен другой тип файла. Синтаксис команд в этом файле во всех отношениях совместим с синтаксисом, используемым при нормальной работе NULU. Были добавлены дополнительные функции для упрощения разработки файлов NCF. Каждая командная строка в файле должна быть завершена символами CR, LF, будет преобразована в верхний регистр и отражена в консоли перед обработкой. Последняя строка в файле должна закончиться по крайней мере двумя комбинациями CR, LF. После обработки всех команд NCF, управление возвращается консоли. Любые команды, появляющиеся после команды "-<", будут проигнорированы.

Если в файле встречается другая команда "-<", текущий командный файл будет приостановлен, и открыт новый файл. Когда все команды в этом новом файле будут обработаны, управление "возвратится" к файлу который "вызвал" новый файл. Такая вложенность может продолжаться до уровня, разрешенного числом входных файлов патчем, описанным в разделе [*Модификация NULU*](#_Модификация_NULU_1). (На самом деле вложенность может иметь большую глубину, если отсутствуют открытые библиотеки, но какой смысл в этом? Хм..., может быть учебник...)

Передаваемое имя файла, должно быть однозначным.

Следующие команды допустимы только, когда получены из файла NCF: -j, -z, -". Другие могут использоваться в любое время, но включены здесь и отсутствуют в меню из-за своего программного характера.

**-J Переход к строке**

Синтаксис: **-j номер строки**

Используйте эту команду, чтобы начать читать командные строки NULU из текущего файла NCF с указанного номера строки. Например:

**-j 1**

заставит NULU начать читать строки из текущего файла NCF, начиная со строки с номером 1. Допустимы ссылки вперед и ссылки назад. Ссылка на строку с номером 0 будет проигнорирована. Любые ссылки на не существующие номера строк приведут к завершению текущего файла NCF, тем самым возвращаясь к следующему более высокому уровню NCF, или в командный режим, если это самый высокий уровень NCF.

Обратите внимание, что, если вы добавляете или удаляете какие-либо строки в файле NCF, который полностью испортит любые команды перехода в том файле. Это - ошибка, а не функция.

**-Z Переключение консоли**

Синтаксис: **-z**

Эта команда переключает весь вывод на устройство консоли. Т.е. если эта команда введена один раз, весь вывод на консоль закрыт, за исключением специальных команд, перечисленных ниже. Повторный ее ввод приведет к повторному включению вывода на консоль.

**-' Печать текста**

Синтаксис: **-' "текст для печати"**

Используйте эту команду, чтобы напечатать текст в консоли, независимо от состояния переключателя вывода консоли. Текст для печати должно быть заключено в кавычки. Открывающая кавычка должна присутствовать, но закрытие кавычка необходима только для разграничения любых последующих команд.

**Подсказка и цепочка к предыдущему режиму**

Синтаксис: **-mode to chain to -" "prompt" -failure mode**

Мой метод синтаксиса диаграмм совершенно не соответствует описание этой команды, я знаю. Давайте рассмотрим ее подробнее. Эта команда позволяет программисту NULU определять режим, которому он хочет передать параметр, запрашивает у пользователя этот параметр и также определяет действия, которые будут выполнены, если ответ от пользователя не получен. Теперь для примера...

**-y \*lbr -o -" "open which library? " -j 65535**

Будет получен следующий результат:

1. Список каталога всех файлов с типом .LBR будет перечислен на устройство консоли, если консоль была включена.
2. Явно вводится режим открытия библиотеки (OPEN LIBRARY) пока только, чтобы NULU принял его во внимание для последующих действий.
3. На консоль будет выведено сообщение:

**(CR, LF)**

**OPEN WHICH LIBRARY?**

и NULU будет ожидать ввода пользователем строки с текстом, **не превышающим** длину подсказки (!). Если пользователь введет только RETURN или пустую строку, подсказка снова появится на следующей строке. Если пользователь введет допустимую строку, первое слово в той строке (разделенное пробелом, символом табуляции, и т.д.) будет передано режиму OPEN LIBRARY, и остаток строки будет проигнорирован. Если с другой стороны пользователь ведет ESCAPE, сопровождаемый RETURN (1bh, 0dh) будут выполнены команды расположенные после подсказки, и управление будет передано к строке с номером 65535, почти наверняка не существующей строке, таким образом завершив текущий файл NCF.

Некоторые команды, которые обычно принимают параметры не могут принимать параметры из этого режима, так как они декодируются по-разному. К ним относятся:

**-:**, **-;**, **-l**, **-n**, **-x**, **-y**, **->**

**-: Ожидание RETURN**

Синтаксис: **-: -failure mode (режим отказа)**

Эта команда вынудит NULU ожидать символ RETURN (0dh), который будет введен с устройства консоли. Если будет получен RETURN, остаток командной строки будет проигнорирован. Если будет введен ^C, будут выполняться остающиеся команды в строке.

**-; Комментарий**

Синтаксис: **-; comment about this kludge of a JCL**

Эта команда заставит NULU игнорировать весь текст, появляющийся после команды в той же самой командной строке.

**-> Перенаправление вывода**

Синтаксис: **-> filename**

Синтаксис: **->**

При первой форме этой команды вывод NULU будет отправлен в обозначенное имя файла. Если тип файла не задан по умолчанию будет использоваться Тип файла ".NOF". Если файл уже существует, он будет удален. Также в файл будут отправлены все специальные символы, которые пользователь, возможно, исправил в NULU (см. ниже, [*Модификация NULU*](#_Модификация_NULU_2)), за исключением символа EOF, 26 (1ah). Единственный вывод, не повторенный в файл, будет вывод вызванный, просмотром или распечаткой элемента файла. Даже если консоль выключена файлом NCF, весь консольный вывод по-прежнему будет отправляться в этот файл.

Передаваемое имя файла, должно быть однозначным.

При второй форме этой команды, будет закрыт (при наличии) текущий выходной файл. При завершении работы NULU, будет закрыт текущий выходной файл вместе с текущей библиотекой, если таковые имеются.

# Обработка ошибок

Программа NULU никогда не должна больше "падать" в CP/M. Когда NULU входит в систему, она перенаправляет вектор BIOS "теплой" загрузки, тем самым предотвращая завершение NULU из-за ошибок BDOS. В CP/M Plus она идет еще дальше, отключает консольные сообщения об ошибках, обрабатывая их внутренне. Поэтому, если NULU пытается выполнить некоторую операцию с диском, когда диск не смонтирован, можно обоснованно ожидать (в системах 2.2) выдачу BIOS некоторого сообщения об ошибке. Пользователь тогда нажимает RETURN или ^C, и NULU попытается восстановиться. Она может думать, что диск был полон, хотя реально этого не было, но она должна выполнить последовательность действий, необходимую для восстановления.

# Сообщения об ошибках

Большинство сообщений об ошибках NULU имеет форму:

**ERROR XX: объяснение**

где "XX" некоторое число от 0 до 255, и "объяснение" - это ключ к пониманию причины ошибки. Каждое сообщение об ошибке упомянуто ниже, вместе с объяснением. Приведены, где это уместно, предложения о способах обработки ошибок.

**ERROR 1: reading unwritten data** (чтение незаписанных данных)

|  |  |
| --- | --- |
| Проблема: | Сектор, необходимый для получения доступа к файлу обозначен CP/M как незаписанный. |
| Решение: | Возможно каталог библиотеки или сама библиотека в какой-то момент были повреждены. Удаление записи и уплотнение, вероятно, сделают ее неповрежденной, но самый безопасный путь-это получить новую копию библиотеки. |

**ERROR 2: disk full** (диск переполнен)

|  |  |
| --- | --- |
| Проблема: | Диск библиотеки стал полным во время добавления или уплотнения файла, или диск назначения заполнился при выполнении команды извлечения. |
| Решение: | Если ошибка произошла во время уплотнения, NULU должен восстановиться самостоятельно, повторно открыв старую библиотеку. Если он не восстановился, то вы удалили исходный диск, прежде чем уплотнение было завершено, или возникла серьезная ошибка чтения.  Если ошибка происходит во время добавления файла в библиотеку, операция не должна причинять вред, но никакие файлы, которые требуют дополнительного места на диске не могут быть добавлены в библиотеку, пока некоторое пространство на диске не будет очищено. Обратите внимание, что удаленные записи могут все еще быть перезаписаны входящими файлами пока совпадают размеры файлов. Входящий файл, который вызвал ошибку, будет записан в каталоге библиотеки как удаленная запись. Если ошибка происходит в результате попытки замены элемента, исходный элемент библиотеки будет невредимым и останется активным элементом.  Если ошибка происходит во время извлечения или распаковки файла файл назначения удаляется, потому что он является неполным файлом. Извлечение может продолжаться как прежде. |

**ERROR 4: reading unwritten data** (чтение незаписанных данных)

|  |  |
| --- | --- |
|  | Смотрите ERROR 1. Это - то же, за исключением того, что BDOS отметил, что отсутствует весь экстент (extent). |

**ERROR 5: can't make file** (не удается создать файл)

|  |  |
| --- | --- |
| Проблема: | Была предпринята попытка создания файла на диске при отсутствии свободных записей в каталоге. |

**ERROR 10: media changed** (носители изменились)

|  |  |
| --- | --- |
| Проблема: | Вы вытащили дискету в середине решающей операции. |
| Решение: | Не делайте этого. Это, вероятно, неустранимая ошибка. |

**ERROR 11: disk i/o error** (ошибка дискового ввода-вывода)

|  |  |
| --- | --- |
| Проблема: | Если вы получите это сообщение, вы, вероятно, используете CP/M Plus. Оно означает, что операционная система обнаружила физическую проблему на вашем диске и чтение или запись не выполнены. |
| Решение: | Попробуйте снова. Я восстанавливался при этой ошибке настойчивостью. |

**ERROR 12: disk r/o** (диск только для чтения)

|  |  |
| --- | --- |
| Проблема: | Вероятно, операционная система обнаружила, что у вас есть наклейка для защиты от записи на вашей дискете. |
| Решение: | Удалите ее. |

**ERROR 14: bad drive** (плохой диск)

|  |  |
| --- | --- |
| Проблема: | Был выбран недопустимый диск для вашей системы (см. ниже, [*Модификация NULU*](#_Модификация_NULU_3)), или была указана буква больше, чем "P". |
| Решение: | Выберите другой диск или повторно исправьте NULU. |

**ERROR 17: password error** (ошибка пароля)

|  |  |
| --- | --- |
| Проблема: | Вы пытались получить доступ к файлу, для которого требовался пароль. NULU не поддерживает доступ к таким файлам. |

**ERROR 18: file exists** (файл существует)

|  |  |
| --- | --- |
| Проблема: | Вы сделали попытку уплотнения библиотеки из одной области пользователя в другую, в которой уже находился файл с тем же именем. |

**ERROR 19: ambiguity error** (ошибка неоднозначности)

|  |  |
| --- | --- |
| Проблема: | Это означает, что введена неоднозначная спецификация файла, где требовалось однозначное имя файла, или что спецификация файла назначения в команде извлечения или распаковки, была менее неоднозначной, чем спецификация исходного файла. См. *-E извлечение элементов* для получения дополнительной информации. |

**ERROR 70: file table full** (таблица файлов заполнена)

|  |  |
| --- | --- |
| Проблема: | Вы попытались открыть больше файлов NCF, чем поддерживает ваш патч вложенности. См. [*Модификация NULU*](#_Модификация_NULU_3) |

**ERROR 74: checksum error** (ошибка контрольной суммы)

|  |  |
| --- | --- |
| Проблема: | Во время распаковки файла сгенерированная внутренняя контрольная сумма не соответствует, заданной в явном виде контрольной сумме, присутствующей в таблице декодирования сжатого файла. |
| Решение: | Получите новую копию элемента файла. Ваш был поврежден тем или иным образом. |

**ERROR 75: CRC error** (Ошибка CRC)

|  |  |
| --- | --- |
| Проблема: | При добавлении в библиотеку каждого элемента файла, для него создается CRC и сохраняется в каталоге библиотеки. Если, при извлечении или распаковке, CRC, сгенерированный в это время, не соответствует исходному CRC в каталоге, который означает, что повреждены секторы на которых эта запись библиотека находится. Существует небольшой шанс, что вместо них повреждена запись CRC в каталоге библиотеки, однако это условие, вероятно, также генерирует ERROR 107. |

**ERROR 77: not enough memory** (недостаточно памяти)

|  |  |
| --- | --- |
| Проблема: | Была предпринята попытка открыть библиотеку с слишком большим каталогом, с учетом доступной памяти. |
| Решение: | Откройте библиотеку на компьютере с большим TPA и реорганизуйте ее, чтобы она мола быть обработана меньшей системой. |

**ERROR 78: user cancel** (прервана пользователем)

|  |  |
| --- | --- |
| Проблема: | Проблемы отсутствуют. Просто NULU сообщает пользователю причину отмены операции. Большинство операций, которые работают с неоднозначными спецификациями файла, могут быть прерваны, вводом ^C во время их работы. (вероятно, вам придется удержать клавишу некоторое время.) Так может быть прервана также операция уплотнения. |

**ERROR 83: bad syntax** (недопустимый синтаксис)

|  |  |
| --- | --- |
| Проблема: | Использован неподходящий синтаксис при попытке выполнения некоторой операции. |

**ERROR 85: file not found** (файл не найден)

|  |  |
| --- | --- |
| Проблема: | Спецификация файла, обозначенная для некоторой операции, не может быть найдена. |

**ERROR 86: incompatible cp/m version** (несовместимая версия CP/M)

|  |  |
| --- | --- |
| Проблема: | Была предпринята попытка выполнить NULU в системе CP/M с номером версии меньше 2.0. |

**ERROR 100: bad library directory** (плохой каталог библиотеки)

|  |  |
| --- | --- |
| Проблема: | Указанный файл не может быть открыт как файл библиотеки. |
| Решение: | Возможно плохой каталог, но скорее всего просто файл не является библиотекой. Проверьте его с помощью некоторого дискового редактора, например, SPZ или EDFILE. |

**ERROR 107: library directory CRC error** (ошибка CRC каталога библиотеки)

|  |  |
| --- | --- |
| Проблема: | CRC, вычисленный ранее и сохраненный в каталоге библиотеки, был изменен и не соответствует CRC, вычисленному при открытии библиотеки в этот раз. |
| Решение: | В большинстве библиотек это действительно ошибка. В этом случае единственным выходом является получение новой копии библиотеки. Однако, есть некоторые библиотеки, которые были созданы при помощи LU, затем изменены, но никогда не уплотнены NULU 1.0. В этом случае, вполне вероятно, что старый исходный CRC, сохраненный в каталоге библиотеки, все еще там, несмотря на то, что он вышел из употребления. Мое предложение, когда вы в первый раз получаете NULU 1.5, пройдите через все ваши .LBR файлы, если вы ранее использовали NULU 1.0 и уплотните их. Таким образом, в дальнейшем вы будете уверены, что все CRC, сохраненные в каждой библиотеке правильные. |

**ERROR 116: squeeze decode table error** (ошибка таблицы декодирования)

|  |  |
| --- | --- |
| Проблема: | Файл содержит таблицу декодирования сжатия, но в какой-то момент таблица была повреждена или, возможно, отсутствует. |
| Решение: | Нет решения. Получите новую копию файла. Если размер буфера данных уменьшился менее 9 секторов, то причина могла состоять в том, что таблица декодирования не могла быть считана в память полностью. |

**ERROR XX: undefined** (неопределена)

|  |  |
| --- | --- |
| Проблема: | Неизвестна. |
| Решение: | Запишите всю информацию, представленную на терминале. Число после слова ERROR особенно важно. Получить информацию для меня. Отправьте ее по почте или по телефону, и я постараюсь помочь. |

Самая серьезная, неисправимая ошибка:

**NULU CRC error**

может произойти при первоначальной загрузке NULU 1.5. Она означает, что NULU не соответствует своему собственному CRC. Вы должны получить новую копию. Или, возможно, вы пробовали какой-нибудь патч не описанный ниже ...?

# Модификация NULU

Патчи перечислены в порядке, в котором они появляются в NULU.COM. Каждый патч объяснен, и значения по умолчанию перечислены рядом с адресом патча. Все адреса абсолютные шестнадцатеричные.

| **Патч** | **По умолчанию** | **Объяснение** |
| --- | --- | --- |
| 00163h | 50h | Количество символов которое может содержать каждая строка нормальной яркости. Не обязательно совпадает с числом, которое вы можете увидеть. |
| 00164h | 50h | Количество символов, видимых в строке одновременно. |
| 00167h | 07h | Этот байт указывает количество файлов, которые NULU может иметь открытыми одновременно. Начальное количество равняется 4 и никогда не должно быть меньше. К нему добавлено значение 3, чтобы позволить 3 уровня вложения файлов NCF. Если значение NCF увеличивается или уменьшается, идентичные изменения должны быть внесены в этот байт. (См. Патч по адресу 00220h.) |
| 00168h | 0010h | Это неясное слово управляет количеством перемещаемых задач, которыми могут управлять подпрограммы управления памятью NULU. Поскольку каждый файл является перемещаемой задачей, здесь применяются модификации, аналогичные предыдущему байту. Начальное значение этого слова - 000dh и никогда не должно быть ниже той точки. (См. Патч по адресу 00220h.) |
| 0016ah | 10h | Количество смежно пронумерованных дисководов в вашей системе. Если вы имеете диски A и B, устанавливаете это значение равным 2. Это препятствует получению доступа к недопустимому диску. |
| 0016bh | 10h | Количество смежных областей пользователя, к которым NULU позволят получить доступ. В нормальной системе CP/M 2.2, Turbo DOS или ZCPR, я понимаю, что это число может быть не более 20h. Для CP/M Plus или большинства систем, которые используют метки времени/даты, это число, не должен превышать 10h. |
| 0019ch | 0005h | Это - адрес вектора BDOS. Все вызовы операционной системы проходят через этот адрес. Если у Вас некоторая странная, нестандартная реализация CP/M, вы можете исправить это, чтобы попытаться заставить NULU работать с нею. |
| 0021ch | 00h | Набор в ненулевое значение, чтобы сделать NULU только для чтения. Не будут выполняться любые операторы, который могут изменить библиотеку или дисковые каталоги. |
| 0021dh | 00h | LU.COM был настроен, чтобы выйти автоматически завершаться при использовании хвоста команды, во время вызова LU. В версии NULU программа принимает команду -X для выхода. Установите этот байт равным 01h, чтобы заставить его действовать как LU в этом отношении. Т.е. автоматически завершаться после выполнения переданного хвоста команды, если он присутствовал. |
| 0021eh | 01h | Установите равным 00h, чтобы NULU запускался в режиме BRIEF ON (не печатать длинные сообщения). |
| 0021fh | 01h | Обнулите, чтобы запретить пользователям возможность запроса ESC, RET из **подсказки и цепочки к предыдущему режиму** в файлах NCF. (также обязательно измените значения в адресах 00167h и 00168h.) |
| 00220h | 03h | Этот байт указывает допустимый уровень вложенности файлов NCF. Я понятия не имею, каков верхний предел, но наблюдайте за размером буфера данных! |
| 00221h | '-Y \*LBR -O',0 | Здесь начинаются 39 байтов, в которых могут быть сохранены любые допустимые операторы NULU. Эта строка будет автоматически выполняться, при загрузке NULU, до передачи других параметров из командной строки. Они могут использоваться для создания версии NULU, который, после загрузки, устанавливает значения по умолчанию для диска/пользователя в B0:, распечатывает меню, сканирует диск с файлами библиотек, затем переключается в режим открытия библиотеки. Эта командная строка может быть похожа на следующую:  **-U B: -H -Y \*LBR –O**  Заметьте, что все символы в этой строке **должны** быть **верхнем регистре**. Вся строка **должна** быть завершена нулевым байтом. Только 39 байтов доступны для текста... ни байтом больше. 40-й байт для нулевого разделителя. |
| 0026fh | 3ch | Это - количество строк, которые NULU распечатает на каждой странице во время печати элемента файла. Измените это значение в соответствии с построчного принтера патчем ниже. Обнулите его, чтобы NULU полностью игнорировал подсчет строк. |

Начиная с 0024ch имеются 5 областей данных, используемые для определения строк, необходимых для управления атрибутами консоли. Каждой строке предшествует байт идентификации и байт, указывающий активную длину. Затем 5 байт доступные для фактического определения. Поэтому в общей сложности 7 байт используются для каждого элемента управления видео. Атрибуты, располагаются следующем порядке:

* Включение изображения пониженной яркости или инверсного изображения (Пониженная яркость предпочтительна)
* Выключение изображения пониженной яркости или инверсного изображения
* Включение подчеркивания
* Выключение подчеркивания
* Очистка экрана

Каждый элемент управления становится активным, как только он будет исправлен.

Кроме того, есть одна строка, определенная для устройства LST:, которая будет выдаваться непосредственно перед печатью файла. Она начинается в 00271h и имеет тот же формат, что и строки выше.

# Заключительные замечания

Пожалуйста, сообщайте о любых ошибках или проблемах Мартину Мюррей {Martin Murray}, почтовый ящик 35972, Даллас, Техас 75235 или позвоните по номеру (214) 351-6117, только вечером и в выходные. Пожалуйста, укажите версию NULU, с которой вы работаете.

Как всегда, пожертвования будут с благодарностью приняты. Потребовалась большая работа, чтобы произвести NULU и если она вам нравится, и она приносит вам пользу, пожалуйста, выразите свою признательность чеком! Примечание: я больше не работаю в SYSTEM SOLUTIONS, поэтому, пожалуйста, не выписывайте чеки им. Сейчас я работаю в INOVA, компании созданной специально для поддержки 8-разрядных компьютеров под управлением CP/M. Ищите новые и интересные розничные продукты от этого источника. (Предшествующее является официальной рекламой.)

**01.11.85**

Я надеюсь, что вам понравится новый выпуск NULU. Даже при том, что я не реализовал все функции, которые запланировал в этой версии, как поддержка времени и даты и несколько открытых библиотек, этот выпуск потребовался из-за некоторых ошибок, которые были обнаружены в NULU 1.1. Также я добавил некоторые новые функции.

Сообщите мне, работаете ли вы с библиотеками, которые имеют более 200 элементов. Я серьезно рассматриваю некоторые будущие обновления NULU, которые ограничат максимальный размер библиотеки в диапазоне 150 - 200 элементов. Однако, если достаточное количество людей, используют более крупные библиотеки, чем эти, я отрекусь.

Очень важно, чтобы я получил информацию от пользователей NULU, относительно самых полезных функциях, какие функции все еще желаемы и какие ошибки обнаружены. Позвольте мне получить известие от вас.

Кстати, я добавлю сжатие к NULU, как только я получу хороший алгоритм, написанный на языке ассемблера. Если у вас есть, чем вы можете поделиться, отправляйте его тоже!

Спасибо,

Мартин Мюррей

Почтовый ящик 35972

Даллас, Техас 75235

(214) 351-6117 только вечером и выходные

**Заплатка для NULU версии 1.5**

Сообщение от Мартин Мюррей с выражением признательности Говарду Гольдштейну {Howard Goldstein}

09.01.86

Заплатка была найдена для NULU v1.5. Однако, как все хорошие заплатки, она вызывает определенные проблемы и может рассматриваться в качестве временного решения, пока я не выпущу версию 1.6. Проблемой была функция BDOS 37, которую NULU15 использует прежде, чем писать на диск в случае, если он был изменен. При определенных обстоятельствах, BDOS теряет разум, когда эта функция вызывается и начинает раздавать ранее выделенные блоки диска. Это никуда не годится. "Решение" заключается в **не использовании** BDOS 37. Теперь проблема состоит в том, что вы не можете сказать NULU зарегистрировать в системе новый диск. Ну хорошо, попытайтесь жить с этим, пока я не получу новый, который будет поддерживать исправление.

Скомпилируйте с помощью вашего любимого ассемблера и объедините полученный файл .HEX в NULU15 с помощью MLOAD.

**MLOAD NULU151.COM=NULU15.COM,NULUFIX.HEX**

Результат будет NULU 1.51 и он будет иметь дату 01/09/86.

Больше, чем особую благодарность приношу Говарду Гольдштейну {Howard Goldstein} за эту заплатку. Он нашел проблему. Он протестировал исправление. Он пробирался через весь мой самоизменяющийся код. Его семья будет признательна за любую помощь в оплате счетов психиатра.

Версия 1.5.1 NULU была выпущена 9 января 1986 и является исправленной версией 1.5. Эта библиотека содержит обе версии, и вы должны прочитать примечания к заплатке 1.5.1 в файле NULUFIX.ASM прежде, чем решить, какую вы будете использовать. Если вы используете систему с жестким диском и/или CP/M Plus то вы, вероятно, не обеспокоитесь 1.5.1. У меня не было **проблем** с 1.5.

Версии до 1.5 не должны использоваться вообще.

John H-B 28/5/86.

Это - новый, но 'неофициальный' выпуск NULU Мартина Мюррей {Martin Murray}. Исходная документация для NULU версии 1.5 все еще применяется за исключением раздела озаглавленного *Модификация NULU*, где изменились адреса патчей. Они подробно описаны в файле NULUMOD.DOC в этой библиотеке.

Я также включил длинный файл с названием NULU152.INF, который показывает сделанные мной изменения. Прежде всего, они предназначены для использования Мартином Мюррей и могут быть не интересны пользователю. Однако, содержат сведения о том, как преодолеть известную ошибку функции BDOS 37 (на самом деле функции BDOS 14) и в программе, желающей использовать эту функцию и простой способ, исправив BDOS (не рекомендуется).

Кроме того, в этом файле есть описание недокументированной функциональности BDOS CP/M 2.2 и CP/M Plus, которая, кажется, доставила неприятность многим авторам программного обеспечения (включая MicroPro в Wordstar). Казалось бы, что Digital Research 'забыла' упомянуть эту бесценную функцию при документировании CP/M 2.2 и Plus. Теперь я раскрываю все ничего не подозревающему миру.

Наконец, хотя я достаточно тщательно проверили эту новую версию NULU, я не могу гарантировать, что она не имеет ошибок. Если кто-либо обнаруживает ошибку (которой не было в версиях NULU 1.5 и 1.51), сообщите мне, и я исправлю как можно скорее. Со мной можно связаться через MBBS Leconfield Англия (0401-50745 - V21/V22/V23 - 24 часа в день см. - бесплатная реклама Мартин]), или голосом по телефону 01-954-8732 (по вечерам - но определенно не 24 часа в сутки).

Если кто-либо находится в контакте со Штатами или Канадой, загрузите эту библиотеку (я не могу позволить себе это сам!).

**Дополнительная информация для NULU версии 1.52**

**15 июля 1987 года**

**Мик Уотерс {Mick Waters}**

Этот документ содержит дополнения и изменения NULU в версиях 1.5 и 1.51, чтобы сделать версию 1.52, в частности для модификации пользователем и историю изменений.

На NULU авторское право (с) 1984, 1985 и 1987 принадлежит Мартину Мюррей {Martin Murray}. Оригинальные авторские права автора остаются неизменными несмотря на эти изменения и условия ее использования остаются прежними.

**История с 1985 года**

Версия 1.5 NULU была выпущена в 1985 и была провозглашена как одна из самых полезных утилит для своего времени, быстрее, чем исходный LU и добавила возможности filesweep, сопоставимые с NSWP Дейва Рандса {Dave Rands}.

Через некоторое время после выпуска, пользователи заметили проблемы при извлечении данных из библиотеки на одном диске в файлы на другой. Это, как обнаружили, происходило из-за ошибки BDOS CP/M 2.2, связанной (как считалось) с функцией BDOS 37 (Сброс диска). Временное исправление было выпущено Мартином (как версия 1.51) которое эффективно предотвращало пользователям извлекать файлы на диски отличные от значения по умолчанию. Заметим, однако, что эта проблема не возникает при запуске под CP/M Plus. Впоследствии, я обнаружил ошибку в подпрограмме Krunch, которая как я думал в то время, при определенных обстоятельствах, приводила к повреждению хорошей библиотеки. Так как NULU была слишком хороша, чтобы выбросить и, так как новый выпуск, исправляющий ошибку функции BDOS 37, я решил внести необходимые изменения.

Теперь, спустя приблизительно два или три месяца, после дизассемблирования NULU версии 1.5 и нескольких бессонных ночей, я предлагаю использовать версию 1.52 вместо любого официального обновления от Мартина.

**Возникшие проблемы**

Как и было обещано в моем первоначальном сообщении об ошибке, я включаю (намеренно расплывчато) список обнаруженных проблем и внесенных изменений. В том отчете я также обещал добавить подпрограмму разуплотнения - боюсь, придется подождать, пока у меня не появится достаточно времени.

Проблема Krunch, как оказалось, была не столь серьезна, как я сначала предполагал. Как только я вошел в код Мартина, я обнаружил, что несмотря на то, что NULU будет, при этих обстоятельствах которые я описал, неправильно уплотнять библиотеку, она обнаружит ошибку (как ошибку CRC), и никакого вреда сделано не будет.

В дальнейшем мои ссылки на внутренние подпрограммы NULU по необходимости расплывчаты, так как без оригинального источника, я не могу ссылаться на осмысленные имена меток. Мой дизассемблер, конечно же, присвоил меткам имена в виде L0123 и J1234 как я предполагаю, довольно бессмысленные для Мартина. Однако вооруженный дизассемблером, он должен быть в состоянии видеть, внесенные мной изменения.

Исправление ошибки Krunch касается вызова подпрограммы, которая читает дисковый файл с вычислением CRC (исходный адрес 1BBAH). Эта подпрограмма вызывается с A=1, если требуется начальный поиск, для нахождения начала требуемого элемента файла. Изменение включает его сохранение в качестве флага и, если 1BBAH возвращает ошибку, которая содержит последующую запись на диск (условие полного буфера), флаг будет сброшен в его исходное состояние (1 или 0) при необходимости. Если какие-либо данные были считаны, флаг должен быть обнулен, чтобы избежать многократного поиска. Следующий фрагмент кода, использует присвоенные мной имена меток, но, я надеюсь, что Мартин распознает его:

LD A,1 ; Установить флаг для начального поиска

LD (SEEKF),A Добавлена

J1438: CALL SWPFCB ; Поменять местами FCB

; PUSH AF Удалена

LD A,0 ; Сохранить номер пользователя текущей библиотеки

L143D EQU $-1

CALL SUSRIR ; Установить пользователя, если уже не там

; POP AF Удалена

LD A,0 Добавлена ; Восстановить флаг поиска

SEEKF EQU $-1 Добавлена

CALL RDFCRC ; Чтение файла с CRC

CALL SWPFCB ; Поменять местами FCB

JP Z,J1460 ; Переход, если какие-либо ошибки

LD A,L ; Проверка записей для чтения

OR H

LD HL,(L1425) ; Получить текущее количество записей

ADD HL,DE ; Добавить толко что прочитанные записи

LD (L1425),HL ; Сохранить последнюю позицию в библиотеке

LD HL,(L146A) ; Получить записи для записи

ADD HL,DE ; Добавить толко что прочитанные записи

LD (L146A),HL ; Сохранить новое число записей для записи

JP Z,J13A6 ; Переход, если ничего не осталось для чтения

SUB A Добавлена

LD (SEEKF),A Добавлена ; Очистка флаг поиска для следующего чтения

LD A,77 ; Ошибка 77: недостаточно памяти

; (интерпретируется как полноый буфер)

J1460: OR A

JP Z,J13A6 ; Переход, если достигнут конец элемента файла

CP 77

JP NZ,J1495 ; Переход, если не из-за нехватки памяти

;

; Входной буфер теперь полон,

; мы должны записать его в 'WORK-LBR.$$$'.

;

LD DE,0 ; Получить количество записей для записи

L146A EQU $-2

LD HL,(DIRLEN) ; Длина библиотечного каталога в байтах

LD B,H ; Смещение от начала буфера в BC

LD C,L

LD HL,0 ; Получить выходную позицию произвольной записи

L1472 EQU $-2

PUSH HL ; Сохранить ее

ADD HL,DE ; Вычисляем положение следующий

; произвольной записи

LD (L1472),HL ; Сохранить обновл. позизию произвольной записи

POP HL ; Вернуть текущую произвольную запись

LD A,(CFUSER) ; Получить номер пользователя текущего файла

CALL SUSRIR ; Установить пользователя если еще не установлен

LD A,1 ; Установить переключатель поиска

CALL WRDECF ; Записать DE записей в библиотеку

JP Z,J1495 ; Переход при любой ошибке

LD HL,(DIRLEN) ; Длина каталога библиотеки в байтах

LD (FDBFOF),HL ; Смещение к первому файлу в буфере

CALL J1549 ; Очистить записи для зависи

LD A,(SEEKF) Добавлена ; установить если требуется начальный поиск

OR A Добавлена

JP Z,J1438 Добавлена ; Переход если не требуется начальный поиск

CALL GMBINF Добавлена ; Получить параметры текущего элемента

PUSH HL Добавлена

LD HL,(FDBFOF) Добавлена ; Новое смещение буфера в BC

LD B,H Добавлена

LD C,L Добавлена

POP HL Добавлена

; SUB A Удалена

JP J1438 ; Чтение файла с начальным поиском

-----------------------------

Другая проблема - преодоление ошибки BDOS более сложная. Самое простое решение - исправить BDOS и я включил патч для BDOS Digital Research для тех, кто захочет внести изменение. Однако, существуют опасности, связанные с исправлением BDOS, и я настоятельно рекомендую оставить все как есть. Мой аргумент за оставление как есть выглядит следующим образом:

BDOS CP/M 2.2- стабильный продукт и, гарантированно обеспечивает выполнение одинаковым способом для любого пользователя. Зная его ограничения, можно преодолеть проблемы, вызванные ошибкой (ошибками). Если все пользователи NULU исправят свой BDOS, они могут в один прекрасный день встретить программу, которая работает из-за ошибки, и, следовательно, не будет работать на их системах. Точно так же, любое программное обеспечение, разработанное на 'исправленном' BDOS, не может гарантированно работать на неисправленном BDOS. Тем не менее, я включил патч и решение за пользователями. Свобода выбора - право каждого.

Несмотря на то, что ошибка BDOS связана с функцией 37, на самом деле она находится не в этой функции, а в функции выбор диск (функция 14). Проблемы могут возникать только при следующих обстоятельствах:

1. Диск по умолчанию был сброшен, и
2. Данные были записаны на диск по умолчанию после сброса диска, и
3. На другом диске запрашиваются какие-либо дисковые операции.

BDOS сохраняет 16 разрядную переменную (вектор регистрации), которая показывает состояние регистрации в системе каждого из дисков A-P. В векторе, каждый бит представляет один диск и устанавливается, если диск зарегистрирован в системе. Когда записи записываются на диск, в BIOS используется вектор распределения для ведения учета этих дисковых блоков, выделенных файлу. Вектор распределения обновляется, по мере записи блоков на диск, но первоначально создается из каталога диска, при первоначальной регистрации. Функция 37 BDOS просто сбрасывает надлежащий бит (биты) в векторе регистрации в системе, что должно привести к повторной регистрации диска при следующем доступе.

К сожалению, Digital Research поддерживают внутреннюю переменную для запоминания доступа к последнему диску и проверяет ее перед регистрацией диска. Если требуемый диск совпадает с последним, то BDOS предполагает, что он уже зарегистрирован, и не проверяет вектор регистрации в системе вообще. Если мы сбросили диск по умолчанию, внутренняя переменная все еще скажет, что он зарегистрирован, даже при том, что бит вектора регистрации в системе сброшен. В этом дисководе все еще возможна запись на диск и вектор распределения по-прежнему будет обновляться.

Если мы теперь временно переключим диски, например, для чтения. Внутренняя переменная отразит это и в следующий раз, когда мы получим доступ к диску по умолчанию, BDOS выполнит проверку и теперь скажет "Ах, другой диск - проверьте вектор регистрации в системе". Когда это произойдет, он конечно обнаружит, что бит для этого диска сброшен и перечитает каталог диска для обновления вектора распределения.

Каталог диска обновляется только при прохождении границы экстента (каждый 16k) или при закрытии файла. Следовательно, если нам не повезет, то каталог диска не будет отображать последние несколько операций записи на диск, которые мы сделали, и эти биты будут сброшены в векторе распределения. Это означает, что касается BDOS, эти блоки свободны для использования, и он будет использовать их снова - даже если это означает выделение того же блока (блоков) дважды в один и тот же файл.

В контексте NULU, предположим, что библиотека, расположена на диске A, открыта и мы в данный момент зарегистрированы на диске B (т.е.: диск B является диском по умолчанию). Предположим, что мы хотим извлечь 100k элемент файла на диск B. NULU создает буфер при открытии библиотеки и 2k буфер для извлечения файла. Буфер для библиотеки зависит от остающегося размера TPA, но достаточно сказать, что он будет меньше нашего элемента файла. Затем мы заполняем буфер библиотеки первой частью элемента файла и сбрасываем диск B в случае, если диск был изменен (условие 1 удовлетворено). Теперь мы начинаем записывать элемент файла (условие 2), пока буфер библиотеки не исчерпан. Затем, мы далее читаем библиотеку с диска (условие 3).

Возможными вариантами ответов на этот вопрос являются:

1. Либо сделать все записи на диск кратными 16k (не всегда удобно), или
2. Отметить, когда диск по умолчанию был сброшен и также если он был записан с момента сброса. Тогда если происходит какой-либо вызов BDOS, приводящий к дисковой действиям на другом диске (дисковод А в приведенном выше примере), все файлы на диске по умолчанию должны быть закрыты (для обновления каталога диска) и затем вновь открыты. К счастью, у Мартина есть хорошая подпрограмма для этого.

Второй пример грубый, но будет работать со всеми перестановками операций чтения и записи и был выбран мной для использования в NULU.

Исправление этой проблемы использует оригинальную подпрограмму (адрес 2B46H) в NULU, которая обновляет диск до возможного изменения путем закрытия, и повторного открытия всех файлов на затронутом диске. Первое, что нужно сделать, это удалить окончательный переход к подпрограмме 'сброс диска' (адрес 32B9H) и изменить все вызовы на 2B46H, чтобы добавить вызов к подпрограмме 'сброс диска' впоследствии. т.е.:

CALL DSKUPD ; Обновить диск до изменения

CALL RESDSK ; Сброс диска

По другой теме в целом, добавление кода для сохранения и восстановления байта S2 для файлов с размером более 512 К. Интересно отметить, что Digital Research 'забыла' документировать что байт S2 является байтом переполнения экстента. Любой, кто попытался отредактировать текстовый файл в Wordstar, вероятно, обнаружили, что он рехнулся, когда размер файла превышает 512 К. Wordstar не одинок в этом, есть несколько других коммерческих и общедоступных программ, которые сильно страдают при работе с файлами больше 512 К.

Digital Research утверждает, что CP/M 2.2 может обработать файлы до 8 МБ, и CP/M Plus может обработать файлы до 32 МБ. Они также говорят, что байт экстента (в 2.2 и Plus) может измениться в диапазоне от 0 до 31, в CP/M 1.4, это было от 0 до 15. Теперь 32 раза по 16k далеко не 8 МБ не говоря уже о 32 МБ, на самом деле он подходит к тому волшебному числу 512k - это где байт S2 используется. Байт S2 считает количество 512k используемых в файле и может измениться от 0 до 15 в CP/M 2.2 (биты 0-3) и от 0 до 63 в CP/M Plus (биты 0-6). Бит 7 байта S2 используется BDOS в качестве внутреннего 'архивного' бита. При открытии файла, проверка байта S2 покажет значение 80H. Если какая-либо запись выполняется в рассматриваемый файлу, этот бит очищается. Когда вызывается функция закрытия файла, если бит установлен то, никаких действий не требуется. Если он очищен, дисковый каталог обновлен. Теперь вы видите причину добавленного кода.

Далее следуют примеры внесенных изменений. Снова, используемые метки мои. Указанные адреса взяты из оригинальной некодифицированной версии NULU 1.5.

;

; Эта подпрограмма готовится к смене диска, убедившись, что

; все файловые операции до сих пор точно записаны на диск.

; Это достигается за счет закрытия файлов и повторного их

; открытия. Оригинальный адрес 2B46H.

;

DSKUPD: CALL SAVALL

CALL GETUSR ; Получить номер текущего пользователя

PUSH AF

LD A,(FTBIDX) ; Текущий индекс таблицы файлов

PUSH AF

LD A,(DE) ; Получить диск для сброса

LD (L2B76),A

SUB A Добавлена ; Очистить флаги для предотвращения

; бесконечного цикла

LD (RESFLG),A Добавлена ; Очистка флага сброса диска по умолчанию

LD (WRTFLG),A Добавлена ; Очистка флага записи диска по умолчанию

;

; Пройдите через таблицу файлов, сбрасывающую

; буферы файлов, закрывая файлы (на затронутом

; диске) и вновь открывая их.

;

LD C,0 ; Начать с первого файла

J2B57: INC C

LD A,(MAXFIL) ; Макс возможное число открытых файлов

CP C ; Все файлы сброшены

JP NC,J2B6A ; Продолжить если нет

POP AF ; Восстановить индекс таблицы файлов

CALL SNEWCF ; Выбрать новый текущий файл

POP AF ; Восстановить текущего пользователя

; CALL SUSRIR Удалена

; JP RESDSK Удалена

JP SUSRIR ; Установить пользователя, если еще не установлен

J2B6A: LD A,C ; Получить индекс таблицы файлов

CALL SNEWCF ; Выбрать новый текущий файл

JP Z,J2B57 ; Jump if no file entry for this index

LD HL,(FFCBPT) ; Point to the current file FCB

LD A,(HL) ; Получить дисковод

CP 0 ; Диск тот же?

L2B76 EQU $-1

JP NZ,J2B57 ; Перейти если нет

EX DE,HL

LD HL,12 ; Точка экстента

ADD HL,DE

LD A,(HL) ; Получить экстент

PUSH AF

PUSH HL

INC HL ; Точка байта S2

INC HL

LD A,(HL) Добавлена ; Получить байт S2

PUSH AF Добавлена

PUSH HL Добавлена

INC HL ; Точка счетчика записей

LD A,(HL) ; Получить счетчик записей

PUSH AF

PUSH HL

CALL CLOSEF ; Закрыть файл (вызывает обновление диска)

CALL OPENF ; ... и повторно открыть его

POP HL

POP AF

LD (HL),A ; Обновить счетчик записей

POP HL Добавлена

POP AF Добавлена

OR 80H Добавлена ; Флаг нет обновления файла (пока!)

LD (HL),A Добавлена ; Обновить байт S2

POP HL

POP AF

LD (HL),A ; Обновить экстент

JP J2B57

Затем, добавьте код к подпрограмме 'записи BC записей в файл' (адрес 2CB8H), которая проверяет первый байт FCB в отношении диска по умолчанию и устанавливает флаг, если он тот же. После того, как этот флаг установлен, эта проверка не должна повторяться.

;

; Запись BC записей на диск

; Адрес DMA передается в HL

; Оригинальный адрес 2CB8H

;

WRFILE: CALL SETDMA ; Установить начальный адрес DMA

LD HL,WRTFLG Добавлена ; Точка флага записи диска по умолчанию

LD A,(HL) Добавлена

OR A Добавлена

JP NZ,WRF1 Добавлена ; Пропустить если флаг уже установлен

CALL GETDSK Добавлена ; Получить диск по умолчанию

EX DE,HL Добавлена ; Адрес FCB в HL

CP (HL) Добавлена ; Сравнить с требуемым диском

EX DE,HL Добавлена

JP NZ,WRF1 Добавлена ; Дереход если не запись на диск по умолчанию

LD (HL),A Добавлена ; Установить флаг соответственно

WRF1: LD HL,0 Добавлена метка ; Инициализировать счетчик записываемых записей

CALL RSUBEX ; Установить множественные вызовы WRSEQ и т.д.

DEFW RTRUE ; Возврат состояния true после завершения

;

; Следующий код выполняется BC раз

;

CALL WRSEQ ; Запись одной записи на диск

JP Z,INC6SP ; Сбросить стек, если какие-либо ошибки

INC HL ; Увеличить количество записанных записей

PUSH HL

CALL UPDDMA ; Добавить 128 к текущему адресу DMA

POP HL

RET

Добавьте код в подпрограмму сброс диска, которая устанавливает флаг, если диск по умолчанию сброшен. Я сделал это, помещая номер диска по умолчанию во флаг, если требуется, в противном случае оставляя его очищенным.

;

; Сброс обозначенного диска

; Оригинальный адрес 32B9H

;

RESDSK: CALL SAVALL

CALL GETUSR

PUSH AF ; Сохранить текущего пользователя

CALL GETDSK Добавлена ; Получить текущий диск

EX DE,HL Добавлена

CP (HL) Добавлена ; Диск по умолчанию сброшен?

EX DE,HL Добавлена

JP NZ,RSD1 Добавлена ; Переход если нет

LD (RESFLG),A Добавлена ; Установить флаг если диск по умолчанию сброшен

RSD1: LD A,(DE) Добавлена метка ; Получить диск для сброса

LD DE,L0001 ; Принять первоначально диск A

DEC A ; Сделать диапазон 0-15 для дисков A-P

CALL NZ,SLDEXA ; Преобразовать код диска в позицию бита

LD C,25H

CALL GOCPM ; Сбросить диск

SUB A Добавлена

LD (WRTFLG),A Добавлена ; Сброс выполненной записи после сброса флага

POP AF ; Восстановить текущего пользователя

JP SETUSR

Все подпрограммы, которые могут вызвать активность диска должны быть затем перенаправлены от их существующего вызова CP/M на новую функцию, которая будет проверять флаги и действовать соответствующим образом. Точкой входа для обычных процедур, которые передают адрес FCB в DE, является BUGCHK, в то время как подпрограмма SELDSK, передающая код диска в E, вызывает BUGTST. Примерами подпрограмм, для исправления являются открыть файл, закрыть файл, последовательное (произвольное) чтение, последовательная (произвольная) запись, произвольная запись с заполнением нулями, установка атрибутов, поиск первого (следующего), переименование файла, удаление файла, создание файла и выбор диска.

;

; Выбор используемого диска

; Оригинальный адрес 32D1H

;

SELDSK: CALL SAVXAF

LD E,A

LD A,(MAXDSK) ; Макс возможный номер диска

CP E

LD A,14 ; Ошибка 14: Плохой диск

JP C,RFALSE

DEC E

LD C,0EH

; CALL GOCPM Улалена

CALL BUGTST Добавлена ; Вызовите BDOS из процедуры проверки ошибок

INC A

LD C,14 ; Ошибка 14: Плохой диск

JP Z,CPMERR ; Обработка возвращаемого кода ошибки BDOS

RET

;

; У этой подпрограммы нет эквивалента в NULU 1.5

;

; Эта подпрограмма преодолевает проблемы,

; вносимые ошибкой в функции BDOS 14. Она

; делает это, закрывая и вновь открывая все

; файлы на диске по умолчанию, если диск был

; сброшен с помощью функции 37, и диск был

; записан так, так как он был сброшен, И BDOS

; осуществляет доступ к другому диску.

;

BUGCHK: LD A,(CPMVER) ; Получить номер версии CP/M

CP 30H ; Прверка на соответствие CP/M Plus

JP NC,GOCPM ; В CP/M Plus нет ошибки

CALL GETDSK ; Вернуть текущий диск

EX DE,HL ; Точка к требуемому диску

CP (HL) ; Диск тотже?

EX DE,HL

JP Z,GOCPM ; Никаких действий, если диск тотже

BCHK: LD A,(RESFLG) ; Установить если диск по умолчанию был сброшен

OR A

JP Z,GOCPM ; Никаких действий, если нет

LD A,(WRTFLG) ; Установить, если диск по умолчанию

; был записан после сброса

OR A

JP Z,GOCPM ; Никаких действий, если нет

PUSH DE ; Сохранить адрес FCB

LD DE,RESFLG ; Точка для диска который будет обновлен

CALL DSKUPD ; Убедиться, что каталог диска

; в актуальном состоянии

POP DE ; Восстановить адрес FCB

JP GOCPM ; Обработка обычного вызова BDOS

;

; Эта точка входа для выбранной дисковой функции,

; так как ее параметры не передаются в FCB.

;

BUGTST: LD A,(CPMVER) ; Получить номер версии CP/M

CP 30H ; Если это CP/M Plus

JP NC,GOCPM ; Нет действий если это он (ошибка не в CP/M+)

CALL GETDSK

DEC A

CP E ; Тот же диск?

JP Z,GOCPM ; Нет действий если тотже диск

JP BCHK

Приведенной здесь информации должно быть достаточно, чтобы позволить Мартину воссоздавать то, что сделал я (если он захочет). Пользователи, которые хотят исправить свой BDOS DRI для удаления ошибки, могут выполнить следующие действия, но имейте в виду, что вы можете помешать работе программного обеспечения.

**Исправление BDOS DRI**

Обнулите следующие области памяти в BDOS DRI, обратите внимание, что BDOS начинается за 6 байтов перед адресом содержащемся в ячейках 0006H и что ZSID/DDT и т.д. не дадут истинный адрес. Используйте что-то вроде Z3LOC или STATUS, чтобы определить ваш адрес BDOS. Байты в этих местах и что они делают приведены, так, что вы смело можете вносить исправления.

Расположение Байт в этом месте (Hex)

исправлений (xx = зависит от адреса BDOS)

BDOS + 0C45H 3A LDA xxD6H ; Получить требуемый диск

BDOS + 0C46H D6

BDOS + 0C47H xx

BDOS + 0C48H 21 LXI H,xx42H ; Получить последний диск

BDOS + 0C49H 42

BDOS + 0C4AH xx

BDOS + 0C4BH BE CMP M ; Тот же диск?

BDOS + 0C4CH C8 RZ ; Возврат, если да

Это исправление препятствует BDOS использовать свою внутреннюю сохраненную запись последнего доступа к диску и вынуждает его проверить биты вектора регистрации в системе, чтобы определить, нужно ли повторно зарегистрировать диск.

Патч выше работает, но на свой страх и риск!