

Обмен данными с другими операционными системами

Перевод: Роман Игнатов (ignatov@minix3.ru)

Если Вы используете две или больше операционные системы, в определенный момент Вы вероятно захотите обменяться данными между ними. Если они находятся на отдельных компьютерах, и компьютеры связаны в сеть, Вы можете использовать программы подобно **ftp(1)** (доступную для большинства, если не для всех операционных систем) или (между Unix подобными системами) **rcp(1)**. Между несвязанными в сеть компьютерами Вы можете использовать [kermit\(9\)](#) (также доступную для большинства, если не для всех операционных систем), чтобы передать данные по модему или по последовательному подключению.

Обмен данными между различными операционными системами, запускаемыми в разное время на том же самом компьютере (или в одно и тоже время используя виртуальную или эмулированную машину, типа VMWare и Bochs) или на двух компьютерах, которые не могут быть связаны проводом, требует принятия мер, чтобы с одной системы или другой, чтобы можно было бы прочитать и записывать файлы на "чужие" гибкие диски или разделы жесткого диска. С гибкими дисками единственная проблема состоит в том, чтобы интерпретировать структуры данных, используемые чужой файловой системой. С разделами жесткого диска может быть другая проблема, которую сначала надо решить: хотя все операционные системы, которые используют ту же самую аппаратную платформу, они должны использовать метод определения первичного раздела диска, который поддерживается системным загрузочным кодом ROM, проектировщики операционной системы, изобрели различные пути дальнейшего деления первичного раздела на подразделы. Этот документ имеет дело с проблемой интерпретации иностранных файловых систем; для информации относительно проблемы подразделов см. [совместимость разделов](#).

Первичная цель MINIX состоит в том, чтобы проиллюстрировать принципы операционной системы. Требовалось поддерживать MINIX достаточно маленьким и простым, чтобы вписаться в голову студента в течение семестра - или годового курса. В частности, файловая система MINIX поддерживает монтирование носителей, содержащих только файловую систему MINIX.

Доступ к чужим файловым системам из MINIX

В мире Unix монтирование файловой системы имеет определенное значение: смонтированная файловая система - часть отдельного дерева каталогов с / в его корне. Но только потому что MINIX не может обработать чужие файловые системы не подразумевает, что к ним нельзя обратиться. Стандартный дистрибутив MINIX содержит утилиты, чтобы позволить читать и записывать файлы и каталоги в MS-DOS и файловых системах Windows, и для того, чтобы прочитать файлы и каталоги в файловой системе ISO-9660, как на CD-ROM. Эти программы - **dos*** утилиты, **mtools**, и **iso*** утилиты.

Доступ к MS/DOS и файловым системам Windows из MINIX:

Старый путь (перед MINIX 2.0.3) - MINIX **dosdir/dosread/doswrite** утилиты. Они могут обработать гибкие диски и относительно небольшие FAT разделы.

[dosdir](#) - прочитать каталог

[dosread](#) - прочитать файл

[doswrite](#) - записать файл

Если ваша родная файловая система Windows не может быть обработана **dos*** утилитами, есть один путь решения проблемы - сделать дополнительный маленький раздел FAT

Windows, с которым утилиты MINIX могли бы оперировать, чтобы его использовать как плацдарм для переданных файлов между операционными системами.

Новый путь – команда **mtools**. К сожалению, с версии MINIX 2.0.3 нет man-страницы для mtools. Однако, использовать ее довольно просто. Mtools берет первый параметр стандартной команды DOS из командной строки, типа *copy*, *dir* и т.д. Дополнительные параметры являются DOS и путями MINIX. Путь MINIX выражен в стандартном синтаксисе MINIX. Путь DOS выражен устройством MINIX, двоеточие (':'), и путь с разделителем типа косой черты Unix. Например:

```
# mtools copy /dev/fd0:/dir1/file1.txt /usr/local/doc
```

Mtools содержит его собственное справочное средство, напечатайте "mtools-?". Mtools имеет недостаток, он жаден до памяти. По умолчанию это - chmem-редактор, чтобы использовать больше чем 10 мегабайтов. Но большинство систем с дисковыми разделами, слишком большие для семейства утилит dosread, имеют много памяти.

Также, см. статью [using DOS-MINIX as a bridge](#) между MINIX и Windows.

Доступ к ISO-9660 (CD-ROM) файлам из MINIX:

[isoinfo](#) - получить информацию о CD-ROM

[isodir](#) - прочитать каталог

[isoread](#) - прочитать файл

Конечно, чтобы использовать утилиты **iso*** Вы должны знать устройство MINIX, которое соответствует вашему CD-ROM. Оно будет похоже на /dev/c0d1 или /dev/c0d2 на типичном компьютере с одним жестким диском и одним CD-ROM. (Это - для MINIX 2.0.3 или позднее, поскольку более ранние версии используют /dev/hd5 или /dev/hd10). Вы можете использовать часть команды или экспериментировать с isoinfo с различными параметрами, чтобы идентифицировать диск CD-ROM.

Доступ к файлам в Linux ext2 файловой системы из MINIX:

Terry McConnell перенес пакет, названный [ext2tools](#), первоначально написанный для MS DOS, в MINIX. Как первоначально было написано Claus Tondering, этот пакет обеспечивал утилиты для обращения к файловой системе ext2 из MS-DOS. Порт обеспечивает те же самые функциональные возможности для того, чтобы обратиться к файлам Linux из MINIX. Включены следующие программы:

e2cp: скопировать файл с файловой системы ext2.

e2cat: выдать содержимое файла из файловой системы ext2.

e2ls: вывести содержимое каталога для каталогов файловой системы ext2.

Конечно, с 2003 года, в Linux ext2 больше не наиболее широко используемая Linux файловая система. Один способ иметь дело с этим, сделать подобно тому, что было предложено выше для Windows: сделать дополнительный раздел Linux, что использует ext2, или возможно даже в Linux файловую систему Minix, и использовать ее как точку для организации файлов, которые будут переданы.

Доступ MS DOS и Windows к файловой системе MINIX

Terry McConnell расширил пакет первоначально названный MinDOS, чтобы сделать [mintools](#),

который дает доступ к файлам MINIX из приглашения MS-DOS.

Обращение MINIX к файловым системам Linux

Линус использовал тогда текущую MINIX файловую систему как основу для Linux, и хотя теперь жесткие диски в Linux обычно используют ext3 или другую более новую файловую систему, поддержка файловой системы MINIX обычно включается, если требуется доступ к гибкому диску - более новые файловые системы не подходят для гибких дисков, и файловая система MINIX используется для гибких дисков по умолчанию. Только убедитесь, когда Вы конфигурируете Linux, что Вы включаете поддержку файловой системы MINIX.

Передача файла между гостевой и ведущей ОС на виртуальных или эмулированных системах

С виртуальной или эмулированной ведущей системой, как делает Bochs или VMWare, часто возникают потребности перемещать данные из ведущей ОС до гостевой ОС или наоборот. В зависимости от возможностей этих двух систем, могут быть несколько способов сделать это.

Сетевое подключение: Если и ведущая и гостевая системы подключены к сети, то передать файл просто. С MINIX на Bochs под Windows, ведущая система может использовать **ftp**, чтобы получить и поместить файлы в файловое пространство гостевой MINIX. Я полагаю, что это будет также работать с MINIX на VMWare под Windows, но поддержка MINIX типа эмулированной карты ethernet представленной VMWare гостевой ОС - еще не часть стандартного дистрибутива MINIX. Если прямое сетевое подключение между гостевой и ведущей системами не возможно, передача файлов через файловое пространство другой машины в сети могла бы быть самым простым путем.

Гибкие диски: Если гостевая ОС может записывать физические гибкие диски, они могут использоваться, чтобы переместить файлы. MINIX может использовать **mtools** или **dosread** и **doswrite** программы, чтобы иметь доступ к MS-DOS, форматировавший гибкий диск. Из ведущей Windows **mintools** может использоваться, чтобы обратиться к дискам MINIX. Для слишком больших файлов, чтобы поместиться на гибком диске, утилита **fdvol**, выложенная в каталоге **dosutils** на сайте MINIX, может использоваться на стороне ведущей Windows, чтобы записать много дисков, которые гостевая MINIX может прочитывать командой **vol**. Со стороны MINIX **dd** может использоваться, чтобы разбить большой файл на меньшие куски, которые **doswrite** или **mtools** может записать на гибкие диски MS-DOS, и на стороне Windows команда **copy** MS-DOS может использоваться, чтобы собрать меньшие файлы в большой файл.

Виртуальные диски: С Bochs на Windows 98 доступ к физическим гибким дискам из гостевой ОС не поддерживается, так что только файлы использованные как виртуальные диски могут читаться и записываться гостевой ОС. Но во многих случаях, виртуальные диски являются более легкими в использовании чем физические гибкие диски, во всяком случае при доступе к гибким дискам из Bochs - очень медленно. К счастью, в отличие от виртуального жесткого диска, новые виртуальные гибкие диски могут быть созданы при выполнении Bochs, и к многочисленным виртуальным гибким дискам можно обратиться в течение отдельного сеанса. Также, **mtools** может использоваться, чтобы форматировать виртуальный гибкий диск файловой системой FAT MS-DOS, и со стороны Windows есть бесплатная программа, **DiskExplorer**, которая может перемещать файлы назад и вперед между виртуальным диском и файловой системой ведущей Windows.

Неизвестные вещи... Поскольку я пишу этот раздел, большая часть моего собственного опыта была связана виртуальными системами VMWare и Bochs на Windows. Я хотел бы добавить к этому разделу больше информации о передаче файла между гостем MINIX на VMWare или Bochs на Linux и системах Unix, так же с гостем MINIX на Bochs на ОС Mac. Информация от читателей была бы ценна.

Ссылки на виртуальные и эмуляторы систем: Вот - источники подробной информации о некоторых из тем, упомянутых в этом разделе.

Patrick Wagstrom's [MINIX under VMWare How-to](#)
[MINIX to host file transfers on VMWare](#)(Jeff Absher).

[File Transfer to and From MINIX-under-Bochs](#)(Bradley C. Kuszmaul).

[MINIX on Bochs 2.1 on Linux](#)(Rob van der Leek).

[MINIX on Bochs on Windows How-to](#).

[Getting Started with MINIX on Bochs on Mac OS](#)