Сергей Яремчук

Игра на чужом поле

Когда-то никто из пользователей серьезно и не задумывался о поддержке файловой системы NTFS в операционной системе GNU/Linux. С выходом Windows XP ситуация кардинально изменилась, теперь поддержка NTFS является одним из требований при выборе дистрибутива.

В принципе работа по поддержке NTFS в ядре Linux ведется уже давно. Так, еще в 1995 году был доступен патч для работы с этой файловой системой для ядер серии 2.0, а начиная с версии 2.1.74 поддержка NTFS была включена стандартно. На сегодняшний день практически все производители оснастили поддержкой NTFS свои дистрибутивы. Проблем с доступом к разделам NTFS не обнаружат пользователи Mandriva, SUSE, ALT Linux, ASPLinux, Slackware, Debian и прочих популярных программных продуктов. Исключение составляют лишь Red Hat, Fedora и их производные, которые, руководствуясь лицензионной чистотой, не включили поддержку данной ФС, хотя для них существуют RPM-пакеты, позволяющие получить доступ к ее разделам. А некоторые дистрибутивы, среди которых Phat Linux (phatlinux.com) и TopologiLinux (www.topologilinux.com), могут быть установлены на раздел, отформатированный под NTFS.

На сегодняшний день известны три проекта, позволяющие получить доступ к разделам NTFS. Два из них распространяются свободно, третий является коммерческим. При этом первые два пошли путем написания драйвера, а третий пытается решить эту проблему с помощью эмуляции.

Драйверы проекта Linux-NTFS

В настоящее время для GNU/Linux существуют два драйвера NTFS, разработанные в рамках этого проекта. Первый по умолчанию используется в ядрах серии до 2.4.18 и в 2.5.0-2.5.10. Он имеет ограниченные возможности по записи в раздел NTFS, но использовать его очень опасно, так как это может привести к разрушению файловой системы. Последняя версия — 1.1.22, разработка первой ветки уже приостановлена. Все усилия сосредоточены на второй версии, которая была фактически переписана заново одним из разработчиков — Антоном Алтапармаковым. Этот драйвер имеет некоторые возможности по записи, но они весьма ограничены. Так, он может записать данные поверх уже существующего файла, но при этом не может изменить размер, добавить новые или удалить имеющиеся файлы. Новый драйвер, разрабатываемый Linux-NTFS (linux-ntfs.sourceforge.net), поддерживает NTFS версий 1.2, 3.0 и 3.1, Unicode и сжатые файлы. Свои проблемы есть и здесь. Драйвер не работает с зашифрованными файлами и квотами, игнорирует информацию безопасности. Положительная же его сторона — умение работать в мультипроцессорных конфигурациях.

Впрочем, винить разработчиков в этой не очень радужно складывающейся ситуации не следует. Строение NTFS довольно сложное и напоминает базу данных: изменение в одном месте влечет за собой достаточное количество замен и во многих других объектах файловой системы, иначе она попросту будет разрушена. При этом система не документирована, и драйверы приходится писать фактически вслепую.

Новый драйвер включается в ядро начиная с версии 2.5.11, поэтому если у вас не сложилось с его поддержкой, то необходимо перекомпилировать ядро, включив нужные пункты. Для более ранних версий доступен патч.

Используемую версию драйвера узнать очень просто:

```
# dmesg | grep -i ntfs
NTFS driver v2.1.6b [Flags: R/O MODULE]
NTFS volume version 3.1.
```

Или другой вариант:

```
# grep -i ntfs /var/log/messages
```

Если вывод этих команд ничего не дал, вероятнее всего, NTFS попросту не поддерживается, что можно проверить, набрав:

```
# cat /proc/filesystems
```

Владельцам Red Hat и Fedora Core можно сразу идти на страницу linux-ntfs.sourceforge.net/rpm/index.html, где доступны откомпилированные RPM-пакеты с необходимыми модулями. Выбираете нужный и устанавливаете. После чего даете команду /sbin/depmod -a для загрузки модуля. Самая последняя версия драйверов доступна по адресу linux-ntfs.sourceforge.net/snapshots либо через репозитарий Bitkeeper (linux-ntfs.bkbits.net).

Монтирование разделов NTFS

Пробуем смонтировать раздел, но для начала узнаем его название:

```
# fdisk -l | grep -i ntfs
/dev/hda7        3522      3649     1028128+          7
HPFS/NTFS
```

Теперь можем монтировать этот раздел:

```
# mount /dev/hda7 /mnt/temp/ -t ntfs
```

Обратите внимание, что ключ -t, указывающий на файловую систему, в данном случае обязателен, иначе система не сможет сама определить ее и выдаст примерно такое сообщение:

```
# mount /dev/hda7 /mnt/temp/
mount: wrong fs type, bad option, bad superblock on /dev/hda9,
or too many mounted file systems
```

Посмотрим на опции, с которыми раздел монтируется по умолчанию; для этого введем:

```
# cat /proc/mounts | grep -i ntfs
/dev/hda7 /mnt/temp ntfs
rw,uid=0,gid=0,fmask=0177,dmask=077,nls=koi8-r,errors=con-
tinue,mft_zone_multiplier=1 0 0
```

Мы узнали, что раздел смонтирован в режиме чтение/запись. Его можно изменить только на чтение, задав в командной строке опцию -r (или -o ro). Владельцем является root (uid=0, gid=0), опция errors указывает, как будет себя вести система при возникновении ошибок. Существуют два варианта — continue (продолжает работу) и recover (пытается восстановить). В настоящее время поддерживается только замена загрузочного сектора резервным. Опции fmask и dmask задают параметры доступа к файлам и каталогам, соответственно, возможно использование общей опции umask, задающей доступ к файлам и каталогам одновременно. NTFS хранит имена в Unicode, но драйвер переводит их в ASCII. Чтобы указать используемый язык, применяются две конструкции: -o iocharset=, или в новом варианте -o nls=. Для отображения русских имен используется koi8-r, возможно задание utf8 (если ядро не поддерживает Unicode, то дополнительно используйте utf8=true). В нашем случае nls была выбрана автоматически, потому что при конфигурировании ядра эта кодировка оказалось прописанной по умолчанию (меню «File Systems → Native Language Support → Default NLS Option»). Параметр mft_zone_multiplier указывает на размер зарезервированной в master file table части, которая содержит информацию о файле. Первоначально он задается при форматировании NTFS, но в процессе эксплуатации может изменяться на лету. Цифра 1 является значением по умолчанию и соответствует 12,5% зарезервированного объема, 2 — 25%, 3 — 37,5%, 4 — 50%. Из всего вышесказанного следует, что полная опция монтирования может быть указана в таком виде (то, что следует после второй -o, в большей части не нужно и дано для примера).

```
# mount /dev/hda7 /mnt/temp/ -t ntfs -r -o nls=koi8-r -o
uid=500, gid=user, umask=0222
```

После этого с чтением данных проблем быть не должно, все имена будут отображаться нормально, а о записи читайте выше.

Утилиты для работы с NTFS

Кроме драйвера проект Linux-NTFS предоставляет также ряд утилит для работы с NTFS под Linux. Большая их часть ориентирована на разработчиков. Все утилиты доступны в пакете ntfsprogs (linux-ntfs.sourceforge.net/snapshots/ntfsprogs-200506061652.tar.bz2). После компиляции и установки в системе появятся десять утилит:

- **mkntfs** предназначена для создания NTFS 1.2 (поддерживается операционными системами Windows NT/2000/XP);
- **ntfsfix** создана для установки измененных драйвером разделов NTFS — нечто вроде scandisk, которая должна использоваться после каждой записи, чтобы предотвратить возможную потерю информации;
- **ntfscat** является аналогом стандартной Unix-утилиты cat и предназначена для чтения файлов в разделах NTFS;
- **ntfsclone** служит для клонирования, копирования, сохранения, создания резервного образа или восстановления раздела с файловой системой NTFS. Работает на уровне секторов диска и сохраняет только используемые данные; неиспользуемые данные заполняются нулями, что позволяет эффективно сжимать полученные образы. Утилита полезна для создания точных копий раздела и восстановления системы. Вот несколько примеров ее применения:

```
# ntfsclone --output system.img /dev/hda1      # создание копии
раздела
# ntfsclone --overwrite /dev/hda1 system.img  # восстановление
раздел
# mount -t ntfs -o loop system.img /mnt/ntfsclone        # а так
можно заглянуть внутрь созданного образа
```

- **ntfscluster** — утилита для идентификации файлов в указанном разделе или области NTFS. Работает в трех режимах: info (режим по умолчанию, показывает общую информацию об области NTFS), sector (отображает список файлов в заданном диапазоне секторов) и cluster (делает то же, что и предыдущий режим, только выводит список файлов в группе).

```
# ./ntfscluster /dev/hda7
bytes per sector:              512
bytes per cluster:             4096
sectors per cluster:           8
bytes per volume:              2212564992
sectors per volume:            540177
clusters per volume:           67522
initialized mft records:       56
mft records in use:            40
mft records percentage:        71
bytes of free space:           2207653888
```

Настроить работу драйвера NTFS от Captive можно с помощью графического интерфейса

sectors of free space: 4311824
clusters of free space: 538978
percentage free space: 99
bytes of user data: 151552
sectors of user data: 296
clusters of user data: 37
percentage user data: 0
bytes of metadata: 4759552
sectors of metadata: 9296
clusters of metadata: 1162
percentage metadata: 0

- **ntfsinfo** выводит атрибуты по номеру inode или имени файла;
- **ntfslabel** выводит или устанавливает метку файловой системы NTFS (может содержать до 128 Unicode-знаков);
- **ntfsls** — аналог Unix-утилиты ls (для Windows – dir) — выводит список файлов в разделе NTFS (монтировать необязательно);
- **ntfsresize** — а вот это действительно полезная утилита, предназначенная для изменения размера файловой системы NTFS, причем без потерь данных (о том, как работать с ней, мы поговорим чуть позже);
- **ntfsundelete** восстанавливает удаленные файлы на разделе NTFS; имеет три режима работы: scan (задан по умолчанию, осуществляет просмотр файловой системы на предмет наличия удаленных файлов, при нахождении которых выводит их список), undelete (пытается, насколько это возможно, восстановить утраченные данные, кроме сжатых и зашифрованных файлов) и copy (полезен по большей части при отладке, сохраняет данные MFT в файл):

```
# ./ntfsundelete /dev/hda7
Volume is dirty.
Run chkdsk and try again, or use the --force option.
```

Сообщение «Volume is dirty» может возникнуть после записи в раздел, изменения размера раздела и других возможных операций, связанных с изменением данных. После каждой такой операции во избежание ошибок рекомендуется проверка раздела средствами Windows:

```
# ./ntfsundelete /dev/hda7 --force
Volume is dirty.
Forced to continue.
Inode   Flags   %age   Date         Size   Filename
-----------------------------------------------------------
16      F..!    0%     1970-01-01   0      <none>
...
51      FN..    100%   2005-06-18   5878   test.jpg
Files with potentially recoverable content: 9
```

Пробуем восстановить один из найденных файлов:

```
$./ntfsundelete /dev/hda7 -s -m test.jpg --force
Volume is dirty.
Forced to continue.
Inode   Flags   %age   Date         Size   Filename
-----------------------------------------------------------
51      FN..    100%   2005-06-18   5878   test.jpg
Files with potentially recoverable content: 1
```

Продолжается разработка еще нескольких утилит. Это ntfswipe, заполняющая нулями свободные части диска; ntfsdefrag — дефрагментатор файлов, каталогов и MFT; для проверки же диска будет использоваться ntfsck. Еще одна утилита — nttools — в перспективе позволит просмотреть, создать, изменить, копировать и находить требуемые значения (эквивалентны командам ntfscp, ntfsgrep, ntfstouch, ntfsrm, ntfsrmdir). В дальнейших планах разработка утилиты ntfsdiskedit, которая позволит производить операции с дисковыми структурами NTFS.

Проект Captive NTFS

Иным путем пошли разработчики проекта Captive (www.jankratochvil.net/project/captive). Они написали оболочку для драйвера, используемого в Windows, что-то вроде Wine для NTFS. Поддерживаются как чтение, так и запись, правда, работает такой «драйвер» все-таки медленнее, чем предыдущий. Для работы Captive потребуются файл ntfs.sys и системный модуль ядра NT ntoskrnl.exe, которые можно взять с установленной Windows XP. Для защиты от возможного краха используется работа в пространстве пользователя при помощи модуля LUFS — Linux Userland File System (http://lufs.sourceforge.net/lufs). Стоит отметить, что Captive работает не со всеми версиями драйверов Windows (полный список проверенных в работе доступен на сайте). Во всяком случае, со всеми файлами, взятыми в русифицированных версиях Windows XP, работать не получилось.

Тогда необходимо будет взять файлы с сайта Microsoft. Но ситуация с их использованием двоякая: с одной стороны, они находятся в свободном доступе, с другой — предназначены для пользователей Windows. Во всех известных нам дистрибутивах, имеющих Captive, их приходится добывать самостоятельно. Для работы их следует положить в каталог /var/lib/captive. Так как драйверы Windows требуют особых привилегий вроде прямого доступа к железу, то полную стопроцентную эмуляцию реализовать не получится, потому что Unix-системы, естественно, будут защищаться от таких процессов. Во время установки создаются новый пользователь и группа captive, от имени которых и будут работать процессы, а сама установка по умолчанию запускается в изолированной среде CORBA sandbox и chroot-окружении. К сожалению, эти ограничения сделали невозможной одновременную работу сразу с несколькими разделами. Сами драйверы доступны как в исходных кодах, так и в перекомпилированном виде со статической линковкой. Работают они одинаково, проблема может возникнуть только с модулем lufs.o, который должен быть собран под определенную версию ядра. После установки для первоначального конфигурирования необходимо запустить утилиту captive-install-acquire,

которая проверит наличие необходимых библиотек и при необходимости закачает все нужное. Если все компоненты готовы к использованию, можно монтировать раздел:

 # mount -t captive-ntfs /dev/hda7 /mnt/utils/

В случае возникновения ошибок вся информация будет доступна в файле /var/log/messages:

 # cat /var/log/messages | grep captive-lufs

Если вам требуется автоматическое монтирование при загрузке системы, используйте скрипт captive-install-fstab с параметром -add, который автоматически добавит в файл /etc/fstab используемый вами раздел.

Драйвер от Paragon

Несмотря на то что практически вся информация по этому продукту закрыта, а сам драйвер является коммерческим, разработка от Paragon Software Group является наиболее функциональным решением для работы с NTFS-разделами под Linux, что, в общем, и не должно вызывать удивления, учитывая опыт работы этой компании. Доступен он в двух версиях — персональной и профессиональной. Как и у предыдущих проектов, поддерживаются все версии файловой системы NTFS, сжатые файлы и каталоги, а также жесткие диски объемом до 127 Гбайт. В персональной версии работа возможна только в режиме чтения, а в профессиональной доступна и запись. Демонстрационную версию драйвера (лицензия стоит $69,95), поддерживающую только чтение, которую производитель разрешает использовать без регистрации в течение 30 дней, можно скачать с сайта www.ntfs-linux.com. Для установки потребуются исходные тексты ядра. Сам процесс очень прост. После распаковки архива запускаете скрипт install.sh (в интерактивном режиме — ./install.sh -interactive, опцией -iocharset=koi8-r можно установить кодировку по умолчанию для раздела NTFS).

После этого, если не выбрано автоматическое монтирование при загрузке, можно смонтировать раздел вручную:

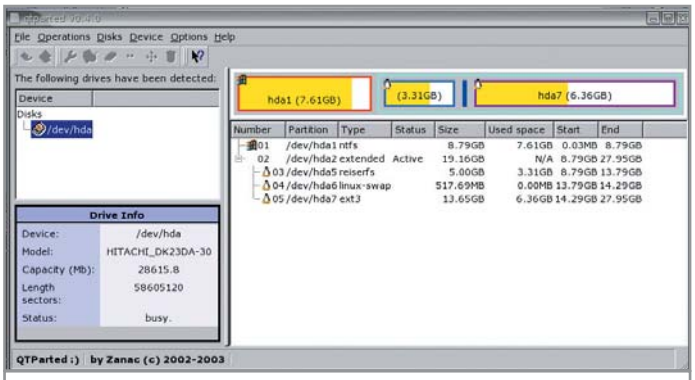
 # mount -t ufsd /dev/hda7 /mnt/test_ntfs

 # mount -t ufsd -o iocharset=koi8-r /dev/hda7 /mnt/test_ntfs

Изменение размеров NTFS в GNU/Linux

Теперь давайте попробуем разобраться, можно ли изменить размер раздела с файловой системой NTFS прямо под GNU/Linux, не прибегая к помощи посторонних утилит вроде Partition Magic? Доступная с 2002 года и входящая в проект Linux-NTFS утилита ntfsresize позволяет изменить размер раздела, не разрушив при этом данных. Поддерживаются все версии NTFS. Разработчиками сделано все, чтобы свести риск потери данных к минимуму. Для подстраховки проводятся всевозможные проверки, включая проверку на непротиворечивость данных. При возникновении проблем или каких-либо подозрений утилита отказывается производить изменение размера. При этом следует учесть, что ntfsresize не манипулирует размерами разделов, поэтому для начала необходимо воспользоваться утилитой fdisk.

Хотя уже имеются графические средства, использующие ntfsresize, будет полезно разобраться с работой первоисточника, поскольку в некоторых дистрибутивах изменить раздел можно только из командной строки. Для работы ntfsresize драйвер поддержки NTFS в ядре не нужен, утилита обращается напрямую к диску. Также, если вы хотите использовать ее в спасательной дискете или вам не требуются все утилиты, можно взять статически слинкованную версию ntfsresize по адресу linux-ntfs.sourceforge.net/info/ntfsresize-static-1.9.4.tgz. Теперь давайте посмотрим информацию о NTFS-разделе.



При наличии пакетов mkntfs и ntfsresize менеджер разделов QTParted работает и с NTFS

```
# ./ntfsresize -i /dev/hda7
ntfsresize v1.9.1
NTFS volume version:        1.2
Cluster size:               1024 bytes
Current volume size:        1052803584 bytes (1053 MB)
Current device size:        1052803584 bytes (1053 MB)
Checking filesystem consistency ...
100.00 percent completed
Accounting clusters ...
Space in use:               5 MB (0,4%)
Estimating smallest shrunken size supported ...
File feature        Last used at        By inode
$MFT:               1 MB                0
You might resize at 4526080 bytes or 5 MB (freeing 1048 MB).
Please make a test run using both the -n and -s options before
real resizing!
```

Команда выдала все о разделе NTFS и сообщила, что можно уменьшить раздел вплоть до 5 Мбайт. Но перед реальным изменением желательно прогнать тест.

 # ./ntfsresize --no-action --size 500M /dev/hda7

Если в результате появится сообщение «The read-only test run ended successfully.», можно смело приступать к изменению размера. Если же команда выдала «ERROR:», лучше для начала исправить ошибки: спешка в данном случае ни к чему хорошему не приведет.

```
# ./ntfsresize -s 500M /dev/hda7
ntfsresize v1.9.1
NTFS volume version:        1.2
Cluster size:               1024 bytes
Current volume size:        1052803584 bytes (1053 MB)
Current device size:        1052803584 bytes (1053 MB)
New volume size:            499999232 bytes (500 MB)
Checking filesystem consistency ...
```

```
100.00 percent completed
Accounting clusters ...
Space in use:          5 MB (0,4%)
Needed relocations:    4394 (5 MB)
WARNING: Every sanity check passed and only the DANGER-
OUS operations left.
Please make sure all your important data had been backed up in
case of an
unexpected failure!
Are you sure you want to proceed (y/[n])? y
# если все данные сохранены, то соглашаемся
Are you sure you want to proceed (y/[n])? y
Schedule chkdsk for NTFS consistency check at Windows
boot time ...
Resetting $LogFile ... (this might take a while)
Relocating needed data ...
100.00 percent completed
Updating $BadClust file ...
Updating $Bitmap file ...
Updating Boot record ...
Syncing device ...
Successfully resized NTFS on device '/dev/hda9'.
You can go on to shrink the device e.g. with 'fdisk'.
IMPORTANT: When recreating the partition, make sure you
1) create it with the same starting disk cylinder
2) create it with the same partition type (usually 7, HPFS/NTFS)
3) do not make it smaller than the new NTFS filesystem size
4) set the bootable flag for the partition if it existed before
Otherwise you may lose your data or can't boot your computer
from the disk!
```

Проверить результат преобразования можно командой ./ntfsresize --info --force /dev/hda7. Как вы сами можете убедиться, утилита свою работу выполнила, но изменила только размер самой файловой системы NTFS, размер же дискового раздела остался прежним (это можно легко проверить, введя команду fdisk -l | grep -i ntfs). Теперь же необходимо проделать еще несколько важных шагов. В начале неплохо на всякий случай сохранить MBR:

```
# dd if=/dev/hda of=hda.mbr bs=512 count=1
```

Теперь запустим fdisk. Конечно, было бы нагляднее воспользоваться cfdisk, но если указать ей новое значение раздела в мегабайтах, то она округлит его до ближайшего значения, которое, вполне возможно, будет меньше требуемой величины, что приведет к невозможности работы с разделом. А вот fdisk вполне корректно обрабатывает ситуацию.

```
# fdisk /dev/hda
Command (m for help): p
Disk /dev/hda: 30.0 GB, 30020272128 bytes
255 heads, 63 sectors/track, 3649 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Device Boot      Start         End      Blocks   Id
System
/dev/hda1   *           1         242     1943833+
83  Linux
....
/dev/hda7        3522        3649     1028128+   7
HPFS/NTFS
Partition table entries are not in disk order
```

Удаляем раздел, на котором размещается NTFS, здесь это 7.

```
Command (m for help): d
Partition number (1-7): 7
```

Создаем на его месте новый раздел размером 500 Мбайт, при этом начальный цилиндр обязательно должен совпадать, то есть, судя по выводу выше, он должен иметь номер 3522:

```
Command (m for help): n
First cylinder (3522-3649, default 3522): 3522
Using default value 3522
Last cylinder or +size or +sizeM or +sizeK (3522-3649, default
3649): +500M
Command (m for help): t
Partition number (1-7): 7
```

Устанавливаем тип раздела, для NTFS – 7:

```
Hex code (type L to list codes): 7
Changed system type of partition 7 to 7 (HPFS/NTFS)
```

Записываем изменения и выходим:

```
Command (m for help): w
The partition table has been altered!
Calling ioctl() to re-read partition table.
```

Если все эти операции были проделаны вами с LiveCD или дискетного дистрибутива, то вы сразу же можете просмотреть информацию о разделе:

```
# ./ntfsresize -i -f /dev/hda7
...
Current volume size: 499999232 bytes (500 MB)
Current device size: 509935104 bytes (510 MB)
```

Если же процесс происходил в работающей с жесткого диска системе, необходимо будет заново перечитать данные, для чего придется перезагрузиться.

Также для нормальной работы обязательно необходимо проверить файловую систему средствами Windows.

Как говорилось выше, уже имеются графические утилиты, в том числе и инсталляторы, позволяющие изменить раздел в наглядном и понятном любому пользователю виде, которые используют код ntfsresize. К таким приложениям относятся DiskDrake от Mandrakesoft, который может встретиться и в других производных от Mandrake дистрибутивах; а также YaST от SUSE, доступный и после установки. Некоторые дистрибутивы используют коммерческие утилиты для разбиения разделов: в ASPLinux это PartitionExpert, Xandros использует PQDisk. Нам кажется, что самыми удобными свободными графическими фронтендами, предназначенными для изменения разделов диска, являются QTParted (http://qtparted.sourceforge.net) и GParted (http://gparted.sourceforge.net).

Заключение

К сожалению, сам собой напрашивается вывод, что работа GNU/Linux с файловой системой NTFS пока еще далека от идеала, и когда появится достойное решение, удовлетворяющее всех пользователей, сказать пока трудно. Впрочем, процесс медленно, но верно движется вперед.