

Exploring the ntfsprogs toolset

DISK TOOLS

The ntfsprogs toolset lets you manage NTFS resources from the Linux command line.

BY YURA PACHUCHIY

The Linux-NTFS project is a team of developers dedicated to providing full-featured access to NTFS from Linux [1]. One of the goals of the project is to develop and support a collection of utilities for managing NTFS resources from the Linux command line. The ntfsprogs toolset offers a variety of tools for getting around in NTFS, including:

- **mkntfs** – creates NTFS volumes.
 - **ntfsmount** – provides read/write access to NTFS through the FUSE module.
 - **ntfsresize** – resizes NTFS volumes.
- The ntfsprogs collection also includes other tools for managing ntfs disk resources [2]. Many of the ntfsprogs utilities are ready to use right now, and a few are still in development. This article describes some of the tools in the ntfsprogs toolset and shows how to use these tools to create, mount, recover, and resize NTFS volumes. As with any disk utilities, you should use these utilities with caution – and always back up any critical data before you begin.

Some History

The first NTFS driver for Linux was created in 1995 by Martin von Loewis and

others. In December 1997, the NTFS driver was merged into the mainline kernel (2.1.74). The driver provided safe read support, but experimental write support was unreliable and often corrupted file systems. Because of these problems, NTFS under Linux received a bad reputation. But things changed a lot in 2001. Anton Altaparmakov, with the assistance of Richard Russon and others, created a new Linux NTFS driver absolutely from scratch. This new driver was merged into the mainline kernel in April 2002 (2.5.11).

The new NTFS driver is very stable for both read and write access, however, write capabilities are very limited and still under development. I joined the project in 2004 and started to implement write support in the libntfs library.

Working in userspace is much easier than working in the kernel, thus I achieved a lot more progress in write support. In 2005, when FUSE (File System in Userspace) [3] was about to be included into the mainline kernel, I realized that it was much easier to write a new userspace driver based on libntfs and FUSE rather than porting new functionality into the current kernel driver; that's how ntfsmount appeared. Both the new kernel driver and ntfsmount are very stable and safe to use. We have not received any bug reports about volumes corrupted by our drivers. In addition to drivers, the Linux-NTFS project provides the ntfsprogs utilities for manipulating NTFS. Current core team members are (in alphabetical order): Anton Altaparmakov, Mario Emmanlauer, Yuval Fledel, Yura Pakhuchiy, Richard Russon and Szakacsits Szabolcs.

Getting ntfsprogs

Source code for the ntfsprogs toolset (as well as other Linux-NTFS project tools and drivers) is available at the download page of the Linux-NTFS project website [4]. The download page also provides



Device Files

This article will use `/dev/hd a1` when referring to NTFS volumes, but this may differ on your system. First of all, you should determine your hard drive name; this may be `/dev/hdX` (for PATA drives) or `/dev/sdX` (for SATA and SCSI drives):

- **hda** – primary master
- **hdb** – primary slave
- **hdc** – secondary master
- **hdd** – secondary slave

- **sda** – first SATA or SCSI drive
- **sdb** – second SATA or SCSI drive

Then you need to know on which partitions you have NTFS:

```
fdisk -l /dev/hda | grep NTFS
(replace /dev/hda with your drive)
```

This command will print a list of volumes on which you have NTFS (Figure 1).

.rpm packages. Several popular Linux distributions makes their own ntfsprogs packages available to users, including Debian, Suse, Gentoo, and Ubuntu. You may find that your Linux distribution already includes some or all of the ntfsprogs utilities. Consult your vendor documentation, or visit the Linux-NTFS website for more information on downloading the ntfsprogs toolset.

Creating an NTFS Volume

You can create an NTFS volume using the ntfsprogs mkntfs command:

```
mkntfs -f -L 'Data' /dev/hda1
```

The `-f` option tells mkntfs to skip volume zeroing and skip the check for bad sectors. Eliminating the check is a good idea if you know that your hardware is OK, because this option makes mkntfs much faster.

The `-L` switch sets the volume label. Do not be lazy and specify no volume label. Programs like ivman will mount your disks under `/media` with nice names if you set labels, but with no labels, you will have to remember which partition number contains the required files.

With the `-N` switch, you can select the NTFS version. mkntfs supports version 3.1 (default) and 1.2. However, using version 1.2 does not make any sense, because the versions are backwards and upwards compatible; thus, you can use any NTFS version in any Windows version.

After running mkntfs, you should update your partition table so that Windows will recognize the new NTFS volume:

```
fdisk /dev/hda
```

At the prompt, type `t`, then the volume number of the newly created volume, then `7`.

Read/Write Mounting

ntfsmount is a userspace NTFS driver that comes with the ntfsprogs package. It relies on FUSE, so you need a kernel with FUSE support. (FUSE has been included in the Linux kernel since 2.6.14, so just switch on the option in the config file; for previous versions, you will need to build the module.)

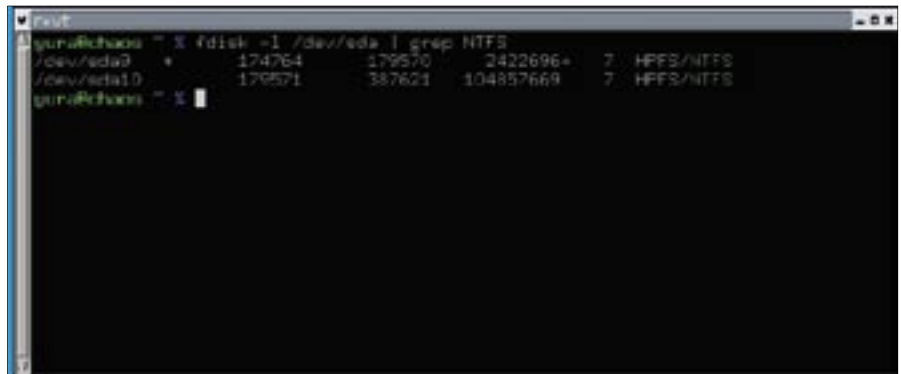


Figure 1: fdisk -l showing a report of NTFS volumes on the hard drive.

After you install FUSE and ntfsprogs, you can mount your volumes by typing:

```
ntfsmount /dev/hda1 /mnt/win_c
```

Alternatively, you can add the following string to `/etc/fstab`:

```
/dev/hda1 /mnt/win_c 7
ntfs-fuse defaults 0 0
```

You will probably want to add the following options:

- `noatime` – do not update access time (adds some speed boost)
- `silent` – do not return error on `chmod` and `chown` operations
- `fmask = 0111, dmask = 0` – allow everyone to have full access to the volume

The latest stable ntfsmount version (1.13.1) has some limitations, although the team is currently working on fixing some of these problems.

For now, the following facts give an indication on where we are with ntfsmount:

- There are no restrictions on writing to files; you can write to files as much as you want.
- Creating files and directories (50% success).
- Removing files and directories (90% success).

When an operation fails, it fails safely, with no damage to the volume.

The reason for these limitations is that creation and deletion of files in NTFS is very complex. The process can

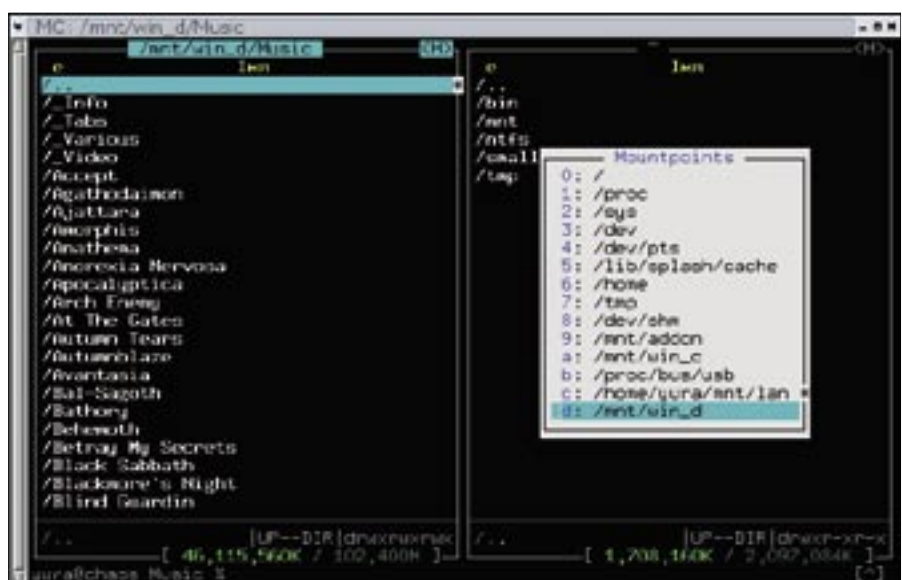


Figure 2: Volumes mounted with ntfsmount appear exactly the same as volumes mounted with the kernel driver. Midnight Commander shows a directory listing of NTFS volumes and a list of mount points with this volume.



Figure 3: root-tail shows ntfsmount messages on the desktop.

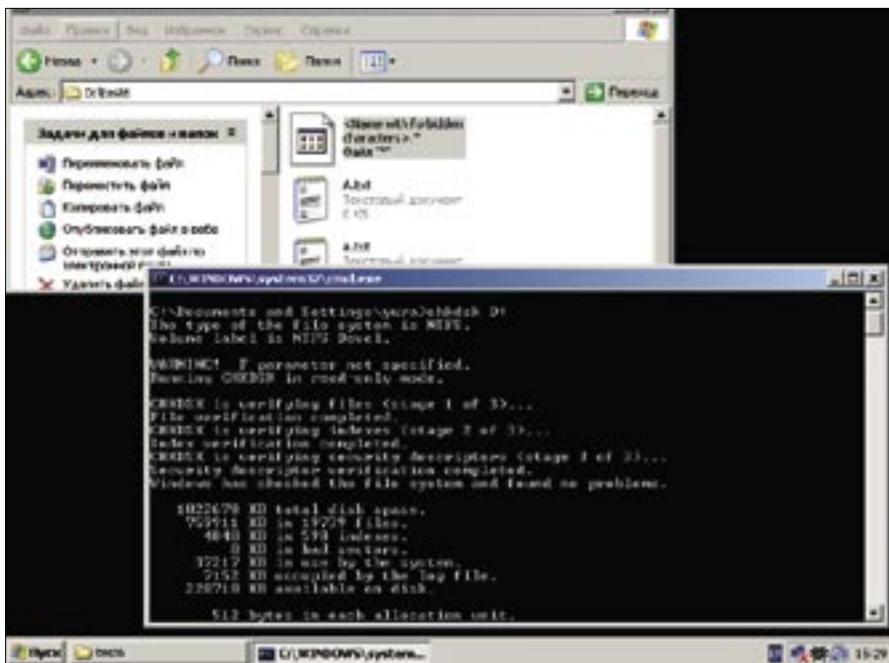


Figure 4: Windows Explorer showing filenames with case-sensitive names and forbidden characters. `chkdsk` says that there are no errors on the volume.

be divided into two parts: updating the MFT (Master File Table) file and updating the directory index. Current stable versions have fully implemented the step of updating the MFT, and they provide a proof of concept for updating the index. Obviously, if one of the subparts fails to do its job, performing the whole operation is impossible.

Since `ntfsmount` daemonizes after mount, it has no terminal to which it can print errors or other information. Thus, `ntfsmount` prints to `syslog`. You can type `cat /var/log/messages | grep ntfsmount`

to see the output; or alternatively, you can use the root-tail program to create dynamic wallpaper with `syslog` content on your desktop (Figure 3).

Some Interesting NTFS Features

NTFS allows the existence of several data streams in a single file. By default, the only data stream is the file content itself. User can access named data streams under Windows by using the `filename.txt:stream_name` semantics. But standard Windows applications will prevent

you from using this form and will claim that you are using the wrong filename. However, you can use the Cygwin command line utilities or FAR Manager to access NTFS data streams. `ntfsmount` provides two named data stream access interfaces. (See “`man ntfsmount`” for details.)

Win32 does not allow characters like `<`, `>`, `*`, `?`, and so on, but NTFS supports any character in the filename except `\0 (NULL)` and `/`. Thus you can use `ntfsmount` to create NTFS filenames that you cannot access from Windows.

There are four kinds of namespaces for filenames: DOS, WIN32, WIN32_AND_DOS and POSIX. (The WIN32_AND_DOS namespace is for WIN32 filenames that also conform to DOS namespace rules.) The first three namespaces are case insensitive, but not POSIX. Windows uses POSIX only for hard links, but `ntfsmount` can create all files in POSIX namespace; thus, it is theoretically possible to have several files in one directory whose names differ only in case. Note, however, that the feature that allows special characters in filenames and case-sensitive filenames is not very usable at the moment.

Recovering Files

It sometimes happens that you delete some file by accident. This might not be a big problem if you delete a program you can reinstall, but things look much worse if you delete some important document that you have been working on for several days. For this case, we provide the `ntfsundelete` utility. `ntfsundelete` will do its best to restore the deleted file, but it cannot do magic and will not help if clusters associated with the deleted file are overwritten.

`ntfsundelete` has two modes useful for end users: `scan` (the default) and `undelete`. At the beginning, you have to scan the volume for files that can be recovered:

```
ntfsundelete /dev/hda1
```

You can see the sample output in Figure 5.

If you want to recover a file named `article.txt`:

```
ntfsundelete /dev/hda1 2
-m article.txt -u
```

Other ntfsprogs Utilities

The `ntfsprogs` collection contains several other utilities for managing NTFS resources. Some of the other tools are:

- **ntfsfix** – at present, we do not have a full featured volume checker, but `ntfsfix` is enough for most cases. `ntfsfix` does some basic recovery (so Windows will be able to recognize the ntfs volume) and then schedules a `chkdsk` check. You can then reboot to Windows and `chkdsk` will perform the check.
- **ntfscat** – displays the contents of a file. You can specify the path to the file or the inode number. Also, `ntfscat` can dump named data streams.
- **ntfsdecrypt** – same as `ntfscat`, but for encrypted files. You should submit a `.pfx` file with a key to make it work.
- **ntfslabel** – lets you show and change the volume label.
- **ntfsis** – just like the `ls` command.
- **ntfscp** – overwrites files on the NTFS volume.
- **ntfstuncate** – changes the file size to a requested setting.
- **ntfsinfo** – displays all metadata for a selected inode or file selected by path in human readable form. Mostly for NTFS experts, but some fields like file attributes and access times can be understood by end users.
- **ntfscmp** – just like the `cmp` command but for NTFS volumes. `ntfscmp` compares volumes inode to inode and attribute to attribute, which makes it extremely useful when used with `ntfsinfo`.

Or alternatively, you can use the inode number (the first column):

```
ntfsundelete /dev/hda1 2
-u -i 39633
```

These commands will create *article.txt* in the current directory with original content. Future versions of *ntfsundelete* may also support file undeleting to the file's previous location.

Volume Resizing

NTFS resizing using command line utilities is rather tricky, and an explanation of the resizing process from beginning to end would take much more than this article. Below is a brief overview of how to resize an NTFS partition. You should refer to the *ntfsresize* man page and the excellent FAQ written by Szakacsits Szabolcs [5] for details.

Resizing consists of two steps: resizing the device and resizing the filesystem. In the case where you are expanding a volume, you should:

1. Run *fdisk*, delete the entry for the partition on which the volume is located, and recreate the entry at the same place, but with a larger size. Run *ntfsresize /dev/hda1*. (Do not forget to replace *hda1* with your actual partition.) *ntfsresize* will automatically detect the new partition size.

In case you are shrinking a volume:

1. Run *ntfsresize /dev/hda1 < new_size >*.
2. Run *fdisk* and recreate the partition at same place, but with the new size.

Cloning

If you are doing backups of your NTFS volumes, and you want to clone operating systems on several machines or move your system to new hard drive, then *ntfscclone* is the tool you need.

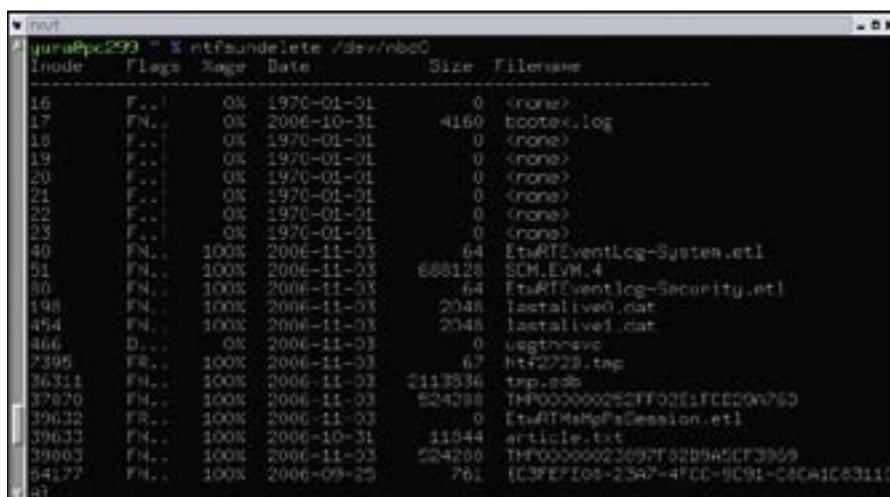
If you want to clone your system from an old PATA drive to new SATA one:

```
ntfscclone --overwrite 2
/dev/sda1 /dev/hda1
```

If the new partition is larger than the old one, you probably want to run:

```
ntfsresize /dev/sda1
```

Note that *ntfscclone* is designed to perform an exact copy of a volume, but



Inode	Flags	Stage	Date	Size	Filename
16	F..	OK	1970-01-01	0	<none>
17	FN..	OK	2006-10-31	4160	bootex.log
18	F..	OK	1970-01-01	0	<none>
19	F..	OK	1970-01-01	0	<none>
20	F..	OK	1970-01-01	0	<none>
21	F..	OK	1970-01-01	0	<none>
22	F..	OK	1970-01-01	0	<none>
23	F..	OK	1970-01-01	0	<none>
40	FN..	100%	2006-11-03	64	EtaRTEventLog-System.etl
51	FN..	100%	2006-11-03	688128	SCH.EVM.4
80	FN..	100%	2006-11-03	64	EtaRTEventLog-Security.etl
198	FN..	100%	2006-11-03	2048	!astalived.dat
454	FN..	100%	2006-11-03	2048	!astalived1.dat
466	D...	OK	2006-11-03	0	usgthnave
7395	FN..	100%	2006-11-03	67	htf2729.tmp
36311	FN..	100%	2006-11-03	2113536	tmp.edb
37670	FN..	100%	2006-11-03	524288	THF0000002CFF02E1FCE09A760
39632	FN..	100%	2006-11-03	0	EtaRTMetaFileSession.etl
39633	FN..	100%	2006-10-31	11544	article.txt
39603	FN..	100%	2006-11-03	524288	THF0000002C697F82B9A5CF3959
54177	FN..	100%	2006-09-25	761	{C3E7E08-23A7-4FCC-9C91-C8CA1C83110

Figure 5: Sample output from the *ntfsundelete* utility.

sometimes it happens that not all parameters in the new environment are the same as the equivalent parameters in the old environment. To make Windows boot again, you can use the *relocntfs* utility [6].

Also, *ntfscclone* has a *--metadata* option that is very useful for submitting bug reports. In this mode, *ntfscclone* will copy only the volume structure, but not any contents of the files. If you are encountering problems with our software and you can reproduce this problem, please do the following:

```
ntfscclone --metadata 2
--output ntfsmeta.img /dev/hda1
gzip2 ntfsmeta.img
```

You will receive rather small file (0.5-10 MB). Please host it somewhere and write email to us with a link to this file and steps for how to reproduce your problem.

Present and Future

In July 2006, Szakacsits Szabolcs completed functions for directory operations, performed several optimizations, and released a new *ntfsmount* version named *ntfs-3g*. Because of differing views on project maintenance, *ntfs-3g* recently became separate project [7].

I ported *ntfs-3g* functionality into the *ntfsprogs* CVS in August. Now Yuval Fledel and I are working on porting other minor changes, fixing problems, and adding some new features and enhancements.

We will release a stable snapshot with a new version of *ntfsprogs* as soon as it

is ready. Check out our project website for updates (www.linux-ntfs.org). The snapshot may be ready by the time you read this article.

Anton Altaparmakov is also working on write support for the kernel driver. But due to a commercial funding agreement, the code for this update will not be released until the Spring of 2008 at the earliest. ■

THE AUTHOR

Yura Pachuchiy was born 2 September 1987 in Minsk, Belarus. He enjoys listening to metal and some latino music, riding his motorcycle, and hacking NTFS. He joined Linux-NTFS project in June 2004. At present, Yura is studying Computer Science at Belarusian State University and working at SaM-Solutions, a company that now sponsors his work for Linux-NTFS.



INFO

- [1] Linux-NTFS Project: <http://www.linux-ntfs.org/>
- [2] *ntfsprogs*: <http://wiki.linux-ntfs.org/doku.php?id=ntfsprogs>
- [3] FUSE: <http://fuse.sourceforge.net/>
- [4] Linux-NTFS download page: http://sourceforge.net/project/showfiles.php?group_id=13956
- [5] *ntfsresize* FAQ: <http://mlf.linux.rulez.org/mlf/ezaz/ntfsresize.html>
- [6] *relocntfs* utility: <http://wiki.linux-ntfs.org/doku.php?id=contrib:relocntfs>
- [7] *ntfs-3g*: <http://www.ntfs-3g.org/>