
## Accessing NTFS from a Linux live distro

# LIVE WIRE

A Linux live distro may be just what you need to recover a Windows computer brought down by a system problem or virus attack. Knoppix creator Klaus Knopper gives you some tips for accessing NTFS from live Linux. **BY KLAUS KNOPPER**

Live distros are designed to boot from the CD or DVD drive of a computer that may have a very different system on the hard disk. In many cases, the computer is a Windows PC with a hard disk formatted for NTFS. This article covers some important facts you may need to know when accessing NTFS from a live Linux system.

## Linux and NTFS

As you will learn elsewhere in this issue, there are currently at least three Open Source implementations of NTFS [1] support in GNU/Linux:

- libntfs, from the Linux-NTFS project [2], and a collection of user-space tools for handling NTFS similar to mtools. The tools include the ntfsmount utility, which uses the FUSE (Filesystem in USErspace) kernel module [3] to connect an NTFS partition to a mount point using libntfs. With libntfs, it is possible to not only read, but also write to NTFS, with the limitation that no index reorder is possible. This means that only up to 9 files or subdirectories can be created per directory, and that files can only be deleted under certain circumstances. This is sufficient for modifying files, and adding one or the other file or archive to a NTFS partition, but not for recursively copying a lot of files.
- The ntfs kernel driver, as a spinoff of libntfs, but with very much reduced functionality (only overwriting of existing files is possible; not deletion or creation of files).
- ntfs-3g [4], a fork from libntfs and ntfsmount. The functions for rebuilding the file system tree correctly have been implemented, thus ntfs-3g allows unrestricted read and write operations on NTFS, using the FUSE kernel module just like the aforementioned ntfsmount from libnfs.

Using FUSE as a bridge to access a file system has several benefits. First of all, it reduces the otherwise necessary effort for implemeting kernelspace functions for file access. In kernelspace, tasks like opening a file or writing to a file are not easy, since they essentially have to be implemented as a sequence of access and locking instructions for getting "pages" of data, which sometimes have to be reassembled and decoded before getting sent back to the program that did

### Knoppicillin

Knoppicillin is a special derivate of Knoppix that is designed as a very lightweight (about 200 MB) GNU/Linux live system-based virus scanner for Windows installations. There are usually a few different virus scanners installed, both commercial and free alternatives. Because of the proprietary licenses of some virus scanners and their databases, Knoppicillin (unlike Knoppix) is not freely distributable, but it can be found in the German c't magazine as a featured add-on. It appears that Knoppicillin has been partially translated into English, although most of the text is still in German.

Since a virus scanner should be able to not only detect viruses, but to also remove, or at least deactivate them, the early read/write functionality of ntfsmount was somewhat insufficient. Removing a file on NTFS, using ntfsmount, was not always possible, and the task was doomed to fail if the file was not a leaf node in the file system tree. In order to still be able to kill viruses in a reliable way, a patch was added to ntfsmount that, in case of being unable to actually delete an infected file, just truncated the file content to size zero and returned a success code to the scanner, making it believe that the removal went well (which in fact is not so much of a cheat in this case). Overwriting the file would have been less of a problem if a working copy of the file would have been available, which is not always the case.

ntfs-3g with unrestricted read/write support for NTFS from Szakacsits Szabolcs has made this patch unnecessary in the recent Knoppicillin release, since file deletion, including reordering the internal NTFS file index structures, is completely implemented.

the actual system call. The FUSE module allows you to use higher-level abstractions, such as library functions, and it implements the complicated direct or cached input/output, so that the filesystem programmer doesn't have to. This could explain why the ntfs kernel filesystem module is a little behind in functionality compared to the libntfs and ntfs-3g alternatives that work on top of FUSE (Figure 1).

On the other hand, mounting a filesystem with FUSE requires more than just the plain *mount* command, since the real filesystem interface has to be provided by a userspace tool. But there are ways you can still use *mount -t ntfs* with the FUSE tools, as this article will demonstrate.

## ntfs-3g

To access an NTFS volume with ntfs-3g, you first have to make sure that the following conditions are met in your GNU/Linux system. Most of these are security precautions required by the FUSE kernel module and the FUSE library:

- The FUSE module must be loaded, and */dev/fuse* must exist and be read/writable by the user.
- The device file or image to be mounted must have read/write permission for the user.
- The desired mount point must have read/write permission for the user.
- The user must also be the owner of the mount point.

If you have no real hard disk partition with NTFS, you can also use an image created by *dd if=/dev/ntfs-partition of=ntfs.img* on a different computer that has such a filesystem installed. Using themkntfs utility from ntfsprogs (see the article on ntfsprogs elsewhere in this issue), you can create an empty NTFS image file or partition by yourself.

Mount the filesystem image *ntfs.img* to */media/ntfs*:

```
ntfs-3g ntfs.img /media/ntfs
```

Now you should be able to access */media/ntfs* from any program and read

### Reading the Warnings

It can happen that, when you try to mount an NTFS partition in write mode with ntfsmount or ntfs-3g, you are greeted by an error message indicating that "Windows did not shut down properly" and therefore the NTFS file system is inconsistent, meaning that the file system cannot be mounted read-write. That error message even suggests that you should "reboot into Windows" in order to get the inconsistency fixed by Windows scandisk. Rebooting to Windows can be difficult if Windows doesn't shut down properly, or if you just simply have no Windows installation.

A thorough NTFS check-and-repair tool like ntfsck for Linux has not yet been written. But luckily, the libntfs-based ntfsprogs package has a tool called ntfsfix, which doesn't actually fix inconsistencies but cleans the NTFS filesystem journal, meaning that the partition can be mounted read/write again without needing any Windows tools to fix it.

and write through this directory. For unmounting the image later, just use the following command:

```
umount /media/ntfs
```

or

```
fusermount -u /media/ntfs
```

## Swapspace

If you boot to a system with Windows installed, you are likely to encounter an NTFS-only hard disk. If the computer is light on RAM, you may be in a situation where it helps to use files on NTFS as swapspace for your live Linux system.

Assuming that */dev/sda1* is a SATA hard disk partition with NTFS (and assuming that the FUSE-related conditions mentioned earlier are met), the commands to create a swapfile of 500 MB and start to swap on it would be as follows:

```
mkdir /tmp/sda1
ntfs3g /dev/sda1 /tmp/sda1
dd if=/dev/zero of=➥
/tmp/sda1/knoppix.swp ➥
bs=1M count=500
mkswap /tmp/sda1/knoppix.swp
(as root) swapon ➥
/tmp/sda1/knoppix.swp
```

which will add 500 MB to your virtual memory pool.

## Knoppix and NTFS

Since the first public release, Knoppix – which is designed as a live GNU/Linux system for everyday work rather than as a pure admin's tool – has frequently been used as a rescue system for non-booting or defective Windows installations because of its capability to at least read-only mount an NTFS partition and create backups of the filesystems contents.

To make it easy to use NTFS in write-mode, Knoppix takes advantage of the fact that the *mount -t filesystemtype* command that is used by virtually every application (and also KDE) for mounting disk partitions calls */sbin/mount.filesystemtype*, if it exists. */sbin/mount.ntfs* is a wrapper script in Knoppix, which will build the command lines for FUSE-based NTFS mounts from the common mount options and execute the actual mount call in a way that users get transparent access to NTFS partitions without having to know about fusermount, ntfsmount, or ntfs-3g specific options. Therefore, it's possible to just click an NTFS partition and get it writable by changing the *read/write status* from the icons menu on the Knoppix desktop.

The Knoppix-specific *tohd=* and *fromhd=* boot options allow you to automatically copy the live CD or DVD contents to the hard disk partition and then start from there to free the CD or DVD drive when working with Knoppix. Starting from Knoppix version 5.1, this is also possible with NTFS, which increased the size of Knoppix's initial ramdisk somewhat in order to hold the necessary drivers and tools, but also allows you to use NTFS as a filesystem holding a vitual harddisk or even swapfile for Linux.

Most live distros also use their own, currently most likely ntfs-3g based, wrapper-scripts to make it easy for the user to access NTFS partitions.

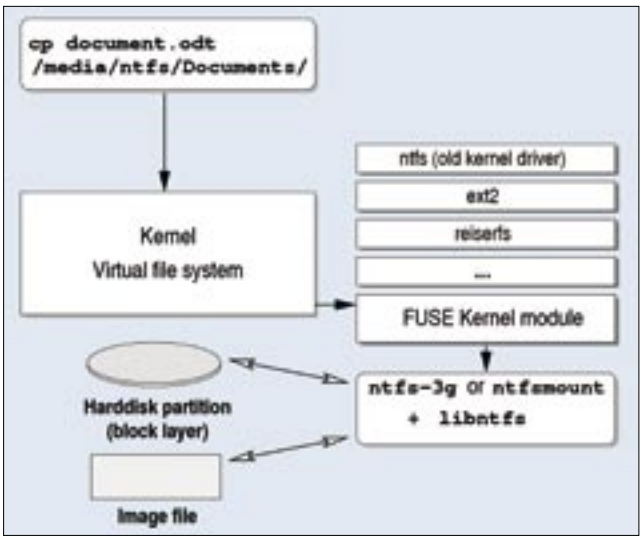


**Figure 1: FUSE lets you build a filesystem using higher-level abstractions.**

## Virus Scanners and Alternate Data Streams

One common use for a live distro with NTFS is to repair a Windows system that has been brought down by a virus. (See the box titled "Knoppicillin.") A problem for all virus scanners is still the fact that NTFS has a feature called "Alternate Data Streams" (ADS), which means that several different files can reside inside the same filename. Scanners tend to only check data from the first stream, which means that there is the possibility for a virus to remain undetected.

ntfs-3g supports two ways to handle alternate data streams. One method is to use a "Windows-like" way of handling ADS files. In order to do this, you need to call ntfs-3g as follows:

```
ntfs-3g -o ➥
streams_interface=windows ➥
/dev/ntfs-partition /media/ntfs
```

Then the alternate streams inside a file can usually be accessed by using

```
filename:streamname
```

instead of just the filename. Unfortunately, there is no option yet in ntfs3g that would allow you to see the ADS components in the *ls -l* directory listing directly, maybe due to possible conflicts with filenames that contain colons – which is also allowed.

An alternative method is the *streams_interface=xattr* option of ntfs-3g. This allows you to view and modify the ADS file contents using extended attributes. For example, *getfattr -n ntfs.streams.list filename* lists the named streams inside an ADS file.

While the ADS features is supported and works well with the Linux implementations of FUSE-based NTFS, thus allowing backups and extraction of such files, none of the Linux-based Windows virus-scanners supports this feature so far, and viruses inside ADS files will most likely go unnoticed. ■

### INFO

[1] Wikipedia on NTFS: *http://en.wikipedia.org/wiki/NTFS*

[2] Linux-NTFS project: *http://www.linux-ntfs.org/*

[3] FUSE project: *http://fuse.sourceforge.net/*

[4] ntfs-3g: *http://www.ntfs-3g.org/*