Reviewing Paragon's NTFS for Linux 5.0

IN THE WINDOW

Paragon's NTFS for Linux is a low-cost commercial alternative for accessing NTFS from Linux. **BY JAMES MOHR**

Linux drivers that allow access to Windows NTFS have been available for a number of years. Initially read-only, they have made some progress in the past few years. You'll learn about some of these alternatives in other parts of this issue, including Captive, which uses Wine to access NTFS through the original Windows driver, and the Linux-NTFS project, which is building a collection of tools around an Open Source driver.

A commercial equivalent to these tools is Paragon's NTFS for Linux. Of course, the benefits of NTFS for Linux compared to the Open Source alternatives will depend on your situation. But at under US$ 20 for the Personal Edition, NTFS for Linux does provide a low-cost and effective solution for many users.

What's It All About?

You can purchase the Paragon NTFS for Linux 5.0 and download it immediately from the Paragon web server. This download version provides you with the source code for the NTFS drivers, which you compile and load as a kernel module. You then have immediate access to your NTFS partitions without needing to reboot.

Written from scratch, the NTFS driver provides full read/write access to NTFS partitions. It was built on Paragon's experiences with their existing products, such as Paragon Partition Manager, and thus they already have a great deal of experience working with NTFS.

Since the driver is supplied as source code and thus must be compiled with your existing kernel source, it naturally means that you have both the necessary kernel source code and gcc (version 2.95 or later installed). As of this writing, kernel versions 2.4.x and 2.6.x are supported. However, the driver is only 32-bit. There is no mention of this limitation in the documentation and, according to the Paragon forums, a number of people have already run into this problem. Per Paragon, the next version due out by the end of the year *should* support 64-bit kernels. As of this writing, it supports NTFS versions 1.2, 3.0 and 3.1, and thus can support Windows NT 4.0, 2000, XP, and 2003.

A fairly detailed 70-page manual is included and, for the most part, the manual is easy to use. You are guided step-by-step through every step from the extraction of the tar-archive through the compilation of the kernel and finally to loading the new drivers. When discussing the most of the utilities, the manual presents the information with useful examples. In the case of the *cpntfs* backup utilities, the information is presented in a workshop format, where you are led step-by-step through activities demonstrating the features of the tool.

I did find the documentation lacking in details on some of the utilities, and the product as a whole suffers from one of my pet peeves: poor translation.

I do not want to say that either the documentation or product is bad, but the translator, who was obviously someone whose native language wasn't English, often tried too hard to keep the original syntax or simply wasn't careful enough. A harmless example appears at the end of the installation: "NTFS utilite was't made." However, there are a few other places in the manual where I was not really sure what is being said.

The product also comes as a professional and "embedded" version. The professional version lists for about US$ 150 and, in addition to a year's worth of upgrades and 60 days of free *installation* support, it provides a number of features and utilities not available with the personal version.

For example, the professional versions supports Microsoft Dynamic Volumes and Software RAIDs through the Paragon LDM (Logical Disk Manager) driver.

In addition to the driver, the product includes a fully functional, bootable ISO CD image with a live Linux version. This gives you read-write access to both Linux and NTFS file systems and thus can be used to recover damaged volumes. It can write to removable media as well as to network filesystems, thus providing an all-around recovery platform.

Also included with the Profession Edition are a number of different utilities to backup and restore NTFS volumes, files, and directories, as well as manage NTFS filesystems.

A Bumpy Installation

As described in the manual, the installation for the NTFS driver is very straight forward:

1. Unpack the tar-archive.
2. Run the *install.sh*.

These steps should build the drivers and modules, install the drivers, detect all the NTFS partitions and mount them, and then reconfigure the */etc/fstab* file to mount the filesystems automatically. However, after unpacking the tar-archive and running *install.sh* as described above, none of my NTFS file systems were mounted and the */etc/fstab* file was *not* modified.

There was a potentially useful message when running the *install.sh* script that said:

```
We can build module ➥
only for 2.6.x kernels
```

However, the kernel is 2.6.13 and the script copied the (Universal File System (UFSD) driver into the correct directory under */lib/modules/*. So this appears to be simply informative and not an error or warning. Still, the script did not do what was described by the manual.

Start the script with the *--interactive* option which allows the user to select "a Linux type" (distribution) and choose whether NTFS partitions should be mounted automatically. This feature was not 100% successful either.

The package of the Profession Edition I received (version 5.18) also contained the source code for the various utilities. Unfortunately, I could not simply run *make* in the source directory as described in the manual, as it spewed out a couple of different errors and did not seem to do anything.

I contacted Paragon with these problems and was sent a new driver file that contained version 5.28. This correctly performs all of the steps described in the manual. Note that, as of this writing, the version still available for download is 5.18. Per Paragon, version 5.28 should be available by the end of the year.

Making the drivers from this new file mounted my NTFS partition under */mnt/ntfs_0* and made the necessary changes to */etc/fstab*. Note that neither the unattended nor the interactive installation does any checking before changing */etc/fstab*. When I inadvertently ran the installation a second time, it repeated the change to */etc/fstab* with no warning.

Taking a Spin

After several tests, transferring large amounts of data, I did not notice any performance differences between my native Windows 2000 system and the Paragon driver or between native Linux filesystems and the Paragon driver. However, the paragon website reports a potential performance lost of 10-15% if the NTFS is "highly fragmented."

Having full read and write access to an NTFS partition from Linux is an obvious improvement over the current state of things. This is what the Personal Edition provides you at a reasonable cost. One would naturally hope for, if not expect, the ability to create an NTFS volume. This is possible, but only with the Professional version. For most home users, the US$ 150 is likely to be overly restrictive. However, for businesses, this could be a worthwhile investment.

As a word of caution, I need to mention that the Paragon website and documentation talks about the features of the bootable CD and NTFS for Linux's ability to make backups of your NTFS partitions. It is fairly clear that the NTFS driver is licensed for each individual system. However, it is unclear on just how many systems you can use the bootable CD. If necessary, check with Paragon directly for details.

Figure 1 shows the default of the *infntfs* command, which provides information about the specified NTFS partition and can set various basic values on the partition, such as the volume label, which I had just changed without a problem.

The *chkntfs* command is similar to Linux *fsck*, although the default behavior is simple to check without making any changes. As you can see in Figure 2, no errors were found. It's nice to know

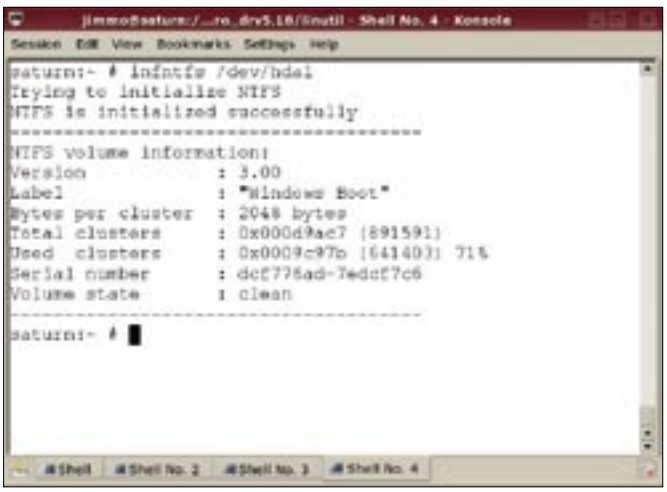

Figure 1: Default output of the infntfs command.

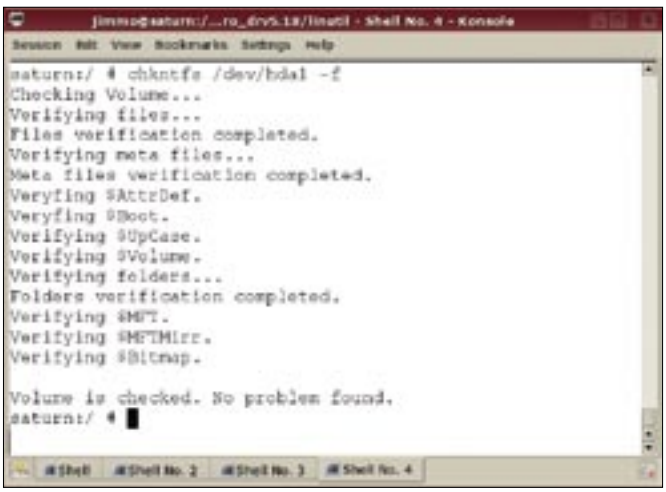

Figure 2: The chkntfs command performs some routine checks.

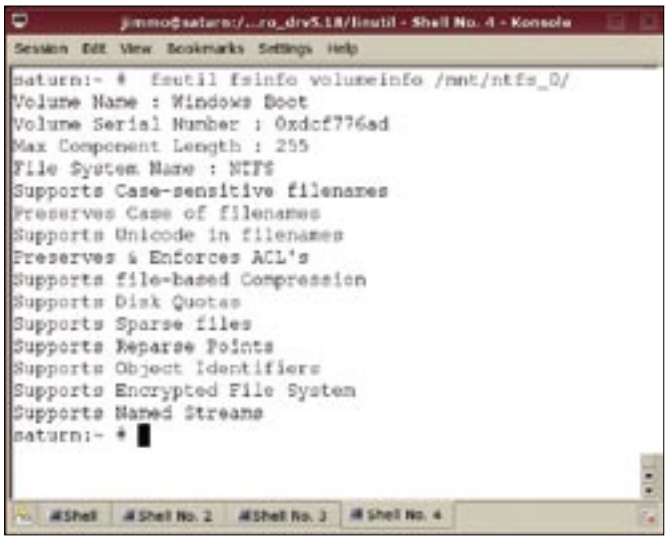

Figure 3: The fsutil utility can display basic volume information.

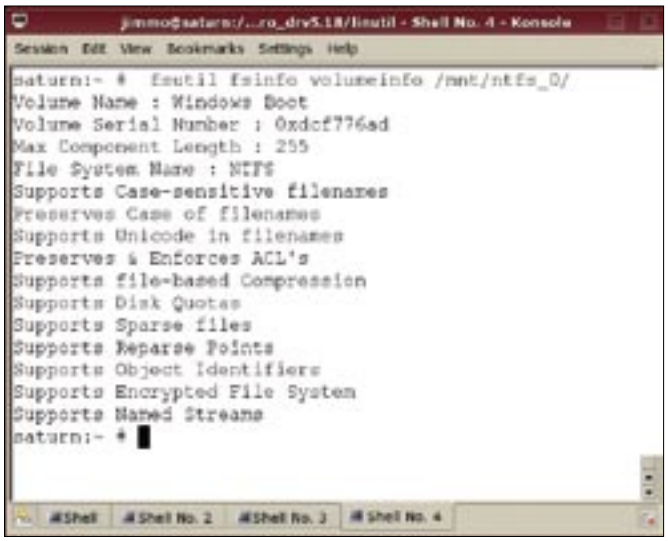

Figure 4: fsutil can also show filesystem details.

that my NTFS is OK, but it would have been nice to know what is being checked here and what each of the values mean. Granted this is Windows-specific information that you can find from other sources. However, considering the other kinds of information the documentation provides, the information on *chkntfs* was relatively sparce.

An NTFS volume is created using the *mkntfs* and, like its Linux counterparts, you have a range of options for various aspects of the filesystem. Naturally, you can define the volume label, but you also have the ability to change the size of the allocation units (i.e., block size), as well as the geometry (sectors and tracks).

Here too, the documentation is a little sparce. For example, it lists the various options, but leaves you guessing in some cases. Although I could usually guess what was meant, there was no explanation of what effects the option has, why it would be used, and so forth.

The *fsutil* is interesting and is described as a "file system utility for advanced users." Considering what information this tool provides and what can be done with it, I would definitely agree with that statement. You have the potential for doing a lot of different things in NTFS, as well as the potential for doing a lot of damage.

Figure 3 shows you the basic volume information. In essence, this is the behavior of the filesystem, and Figure 4 shows you more details of the filesystem itself. Just showing the information is not all this utility can do. It can change the default behavior shown in Figure 3, but also change very low-level aspects of files, such as changing things like their object ID or DOS short filename.

There are many other operations you could preform, most of which I could never imagine doing. Although the documentation does not explain why one might want to change certain values, this is one place where the documentation does good job of explaining what each option means and how to use it.

Perhaps the most useful tool for everyday work is the *cpntfs* backup utility. This utility allows you to copy and restore files on NTFS. This is not limited to copying files between the NTFS and Linux files systems, but also between NTFS partitions file systems should you have more that one on your machine, and with it you can completely restore an NTFS partition. One key aspect is the ability to maintain all of the NTFS attributes.

One might be tempted to think that simply mounting the filesystem(s) under Linux and copying the files would be sufficient, and thus one only needs the NTFS file system driver. The problem there lies in the fact that Linux does not know anything about the NTFS attributes. Thus, if you simply copy files, you end up losing these attributes. Further, the NTFS driver does not support NTFS streams directly and thus you *must* use the *cpntfs* utility in oder to preserve stream data.

This is another place where I found the documentation to be really nice. In addition to explaining how to use the *cpntfs*, the details of the various NTFS attributes are explained.

But Is It Worth It?

Whether this product is worth the money is definitely open for discussion. The biggest question is just how often you need to transfer data between Linux and Windows. I use the Windows on my system only for game playing. However, I created a second partition and formatted it as FAT32 in order to be able to easily copy data between the two systems. So, I can easily say that it is worth the money for the Personal Edition just so that I would be able to format both of the Windows partitions as NTFS rather than FAT32.

If you are curious about how well the driver works, you can first download a bootable ISO CD image with a live Linux system. This liver CD option is fairly easy to use and provides a quick introduction to the product and its features. If you need to go at it the other way around, Paragon also provides Ext2FS Anywhere. This gives you full access to both ext2 and ext3 filesystem from Windows. ■

Paragon NTFS for Linux

Personal Edition:
US$ 19.95
£ 10.95
EUR 14.95
Professional Edition:
US$ 149.95
£ 79.95
EUR 119.95
Website: *http://www.ntfs-linux.com*