Apart No More—Sharing Files between Linux and Windows Partitions

BY DAVID TANCIG

Once most Linux users experience feature-rich software applications such as StarOffice 5.1/5.2 and Applixware 5.0, they realize that the only reason they have a dual-boot installation (Windows and Linux) is so that they can have access to their Windows-based files.

This activity will show you how, while using Linux, you can read from and write to your Windows files in the Windows partition. This isn't a tricky, unstable stunt. It is stable and takes advantage of a well-designed feature of the Linux/Unix operating system that the Windows operating system doesn't have. This is the ability to mount another partition as part of the Linux file system.

Can you do it in reverse? Create a file in Linux and save it for use by a Windows program in the Windows partition? Yes. The technique is simple and relies on the fact that the Linux applications StarOffice and Applixware allow files to be saved with a Windows or DOS text format. Similarly, StarOffice and Applixware both can be instructed to read files created with various Windows and DOS text formats.

It is assumed that you have a dual-boot installation and need to access files created and saved in the Windows partition.

The Mount Command

The Linux/Unix kernel is designed so that additional file systems can be connected to it and be removed as the situation warrants. The file system is the storage scheme that is applied to each disk, CD-ROM, or partition that Linux is asked to fetch data from.

The Linux kernel must be configured to accept DOS file systems. In early distributions of Linux, this feature may not have been enabled. In all of the recent distributions tested, this feature has been enabled.

Before the arrival of Linux, DOS/Windows was the dominant operating system on PCs. Windows came automatically installed. Most computer users never installed an operating system, let alone even knew about disk partitions. Given that they believed that they were the only game in town, Windows developers designed Windows to always take over the first position in the master boot record (MBR).

Consequently, when someone tries to create a dual-boot system, one of two situations arises.

• If Windows currently exists on the disk and the user doesn't want to lose the files, the user makes partitions

David Tancig is the author of *Linux eTudes* and owns a software consulting firm.

with a utility such as Partition Magic. Windows keeps its position as the first partition on the hard drive and consequently, in Linux/Unix parlance, has the first hard drive partition designation of "hda1."

• If the user wants to create a dual-boot system from scratch, the user first uses fdisk to repartition the disk (which reformats and wipes out all the files on the disk). Then, if the user is savvy as to how Windows works, the user installs the Windows operating system and lets it take over the first position in the MBR. Again, Windows ends up in the partition that Linux calls "hda1."

The designation "hda1" is used in the mount command discussed below. If, for some reason, your Windows partition is located on a different partition, then use that particular designation. In the overwhelming majority of cases, the Windows partition is designated as "hda1."

Linux is made aware of file systems through the mount command. The entry to the file system is actually an address. The mount command lets Linux know that this address is actually the entry point to a file system - and associates it with a device-driver file that contains instructions about how to access that file system. The main file system, ext2, in its own partition on a hard drive, is mounted automatically by the script of operating system commands that controls system startup. The user can instruct Linux to mount other file systems after the system has started up.

Without getting into too much detail, hda1 is seldom put into the file system table (fstab) as an MS-DOS file system. This means that the complete command must be issued on the command line. This isn't too much of a bother after you do it once or twice. If you prefer, you can put the command in a script file that has a simple file name to type when you need access to your Windows files. A third option is to embed the script command in your .profile file. This file contains files that are executed every time you log in. You could also enter it into your /etc/fstab file, but that would be a little risky for a new Linux user.

Mounting the Windows Partition

Preparation

First, you need a directory, preferably off the root, that will represent the Windows files.

Step 1 OPEN a terminal emulator.

Step 2 TYPE **cd /**.

Step 3 TYPE **mkdir /WinRoot**.

Step 4 TYPE **ls -l WinRoot**.

> **You should observe your newly created directory.**

Note: The name WinRoot was chosen to remind you that, although you are saving files to what appears to be a subdirectory of Linux, that subdirectory actually represents the root directory of the Windows system.

Mounting the Windows Partition

Step 1 TYPE **B**.

Step 2 TYPE **ls /WinRoot**.

> **You should observe all of the Windows system root directory files and subdirectories.**

That's it. It's that simple. Now you have complete access to your Windows files.

Don't use DOS commands in this directory. Use Linux commands. If you want to do file operations in this directory, use a graphical file manager to do your file operations.

Unmounting the Windows Partition

Preparation

You cannot unmount this partition while it is the working directory. You will get a message that it is busy and cannot be unmounted. You must first change directories and make another directory the working (also called "current") directory.

Step 1 TYPE **cd /**.

Step 2 CHECK where you are.

> **This command should place you in the root directory. You can unmount from there.**

Unmounting

Step 1 TYPE **umount /WinRoot**.

> **That is not a misspelling. The command is umount.**

Step 2 TYPE **ls /WinRoot**.

> **Nothing appears. The file has been unmounted.**

Reading and Writing Files in the Windows Partition

Although Linux and DOS/Windows both use ASCII text characters, Linux and DOS/Windows files are not identical because DOS/Windows and Linux handle end-of-line markers differently. Linux uses the nonprinting ASCII character called a line feed. DOS/Windows uses two nonprinting ASCII characters—a line feed and a carriage return.

There are other file-conversion issues related to special characters. This conversion business can get out of hand very quickly. We need a simple, works-every-time solution.

The average user has an office suite installed. There is no reason not to. For the Linux user, StarOffice 5.1 is free and is easily installed. Or use Applixware 5.0 if you prefer that excellent office suite—or if you need to convert Office 2000 files. The works-every-time solution is below.

Note: All your conversions will be done in Linux.

Converting a Linux File to Use in a DOS/Windows Environment

Step 1 OPEN your office suite.

Step 2 From within the word processor of your office suite, OPEN the file you are exporting to a DOS/Windows environment.

Step 3 SAVE the file with the correct type for the DOS/Windows environment.

> **If it will be used in Windows Word, save it as a Word97 file.**
>
> **If it is to be used with an ASCII text editor (such as would be the case for a source code file), save it as Text DOS (for example, if you are using StarOffice 5.1).**

Step 4 SAVE the file to the appropriate folder in your MS-DOS subdirectory.

Converting a DOS/Windows File to Use in a Linux Environment

Step 1 OPEN your office suite.

Step 2 From within the word processor of your office suite, OPEN the file you want to be imported from the MS-DOS subdirectory.

Step 3 OPEN the file with the correct type for the DOS/Windows environment.

> **If it was a Win97 file, open it as a Word97 file. If was an ASCII text editor file (such as would be the case for a source code file), open it as Text DOS (for example, if you are using StarOffice 5.1/5.2).**

Step 4 CLICK on **Save As**.

Step 5 SELECT the correct data type if you are saving it in a Linux partition.

> **If you are saving it back to the WinRoot subdirectory, then just click Save. You will be saving it with the same data type you had in Windows.**

Note: Several experiments were conducted concerning these transfers. The above instructions will work every time but are a little more conservative than they really need to be.

Scenario #1

If you create an ASCII text file in Linux using a standard editor such as kedit and do not save it as Text DOS:

• If you open it using Notepad, the lack of carriage returns will cause the file to be one long continuous line. The line feeds will show up as black squares in the text.

• WordPad will correctly format the file created in Linux without any conversion. This can be important if you are transferring source code that you want to recompile and try in a DOS/Windows environment.

- Both Notepad and WordPad correctly recognize word wrap.

Remember, when you save a file to the directory where you mount the DOS/Windows partition, you are saving it to the root directory of DOS/Windows. It might be best to name that partition something like "WinRoot" to remind yourself.

Scenario #2

If you create a file in DOS/Windows using an editor:

- Kedit will not recognize word wrap, regardless of whether you save it as pure ASCII or as MS-DOS text using WordPad or Notepad.

- Kedit will correctly format lines, even though DOS/Windows includes a carriage return at the end of every line. This means source code can easily be imported from the DOS/Windows partition.

- To properly read word wrap in the DOS/Windows file, open the file using StarOffice 5.1. At the end of each line where StarOffice 5.1 forced the text to word wrap, press the Enter key to insert a carriage return. Save the file. When you open kedit, the file will be properly formatted.

- To our knowledge, there is no format you can save it in that will automatically recognize word wrap.