Linux Commands

Joshua Go

May 6, 2002

These commands will work with most (if not all) distributions of Linux as well as most (?) implementations of Unix. They're the commands that everybody knows. To be able to survive in Linux, you should know these. There aren't always handy-dandy tools for X that shield you, especially if you're managing your own system, stuff often goes wrong and you're forced to work with the bare minimum.

I hope I didn't scare you.

Navigation

Navigating around the files and directories of your hard drive could be a dreaded task for you, but it is necessary knowledge. If you were a user of command prompt interfaces such as MS-DOS, you'll have little trouble adjusting. You'll only need to learn a few new commands. If you're used to navigating using a graphical file manager, I don't know how it'll be like, but some concepts might require a little more clarification. Or maybe it'll be easier for you. Who knows? Everyone is different.

cd

As you might already have guessed, the `cd` command changes directories. It's a very common navigation command that you'll end up using, just like you might have done in MS-DOS.

You must put a space between `cd` and the `".."` or else it won't work; Linux doesn't see the two dots as an extension to the `cd` command, but rather a different command altogether. It'll come to make sense if it doesn't already.

ls

The letters `ls` stand for **l**i**s**t. It basically works the same way as the `dir` command in DOS. Only being a Unix command, you can do more with it. :-)

Typing `ls` will give you a listing of all the files in the current directory. If you're new to Linux, chances are that the directories you are commonly in will be empty, and after the `ls` command is run, you aren't given any information and will just be returned to the command prompt (the shell).

There are "hidden" files in Linux, too. Their file names start with a dot, and doing a normal ls won't show them in a directory. Many configuration files start with a dot on their file names because they would only get in the way of users who would like to see more commonly used items. To view hidden files, use the -a flag with the `ls` command, i.e. `ls -a`.

To view more information about the files in a directory, use the -l flag with ls. It will show the file permissions as well as the file size, which are probably what are the most useful things to know about files.

You might occasionally want to have a listing of all the subdirectories, also. A simple -R flag will do, so you could look upon `ls -R` as a rough equivalent of the dir /s command in MS-DOS.

You can put flags together, so to view all the files in a directory, show their permissions/size, and view all the files that way through the subdirectories, you could type `ls -laR`.

pwd

This command simply shows what directory you're in at the moment. It stands for "Print Working Directory". It's useful for scripting in case you might ever want to refer to your current directory.

File Management

A lot of people, surprisingly for me, prefer to use graphical file managers. Fortunately for me, I wasn't spoiled like that and used commands in DOS. That made it a bit easier for me to make the transition to Linux. Most of the file management Linux gurus do is through the command line, so if you learn to use the commands, you can brag that you're a guru. Well, almost.

cp

Copying works very much the same. The cp command can be used just like the MS-DOS copy command, only remember that directories are separated with slashes (/) instead of backslashes (\). So a basic command line is just `cp filename1 filename2`.

There are other extensions to the cp command. You can use the -f command to force it. You can use the -p command to preserve the permissions (and also who owns the file, but I'm not sure).

You can move an entire directory to its new destination. Let's say you want to copy a directory (and all of its contents) from where you are to be /home/jack/newdirectory/. You would type `cp -rpf olddirectory /home/jack/newdirectory`. To issue this command you would have to be in the directory where the subdirectory "olddirectory" is actually located.

ln

A feature of linking files is available in Linux. It works by "redirecting" a file to the actual file. It's referred to as a symbolic link. Don't confuse this term with the linking of programs, which is when binary programs are connected with libraries that they need to load in order to run.

The most simple way that I've ever used `ln` to create symbolic links is `ln -s existing_file link`. Evidently there's a hard link and a symbolic link; I've been using a symbolic link all along. You can also use the -f flag to force the command line to overwrite anything that might have the symbolic link's file name already.

To remove a symbolic link, simply type `rm symbolic_link`. It won't remove the file that it's linked to.

mv

The mv command can be used both to move files and to rename them. The syntax is `mv fileone filetwo`, where "fileone" is the original file name and "filetwo" will be the new file name.

You can't move a directory that is located in one partition to another, unfortunately. You can copy it, though, using `cp -rpf`, and then remove it with `rm -rf` later on. If you have only a single partition that makes up your filesystem then you have very little to worry about in this area.

rm

The `rm` command is used for removing files. You use it just like the `del` or `delete` command in MS-DOS. Let's say you want to remove a file called foobar in your current directory. To do that, simply type rm foobar. Note that there is no "Recycle Bin" like in Windows 95. So when you delete a file, it's gone for good.

To delete something in some other directory, use the full path as the file name. For example, if you want to delete a file called "windows" that's in the directory /usr/local/src/, you would type `rm /usr/local/src/windows`.

To remove an entire directory and its contents, type `rm -rf /directory/` where "/directory" is the path to the directory that you want to delete. If you're wondering, the "rf" stands for "recursive" and "force". Be very careful with this command, as it can wreak havoc easily if misused.

Editing

If you haven't figured out how important a text editor is, you soon will. Graphical interfaces can't shield you forever, and those utilities have their limits. Besides, if you're reading this page, I'm inclined to think that you want to be able to customize beyond the capabilities of graphical utilities. You want to work at the command prompt. I know you do.

The basic syntax to invoke these text editors is the same. Type the name of the editor followed by the file you want to edit, separated by a space in between. Non-existent files will be blank. Blank files will be blank as well.

emacs

To use GNU Emacs (or its counterpart, XEmacs), there are really only two commands you need to know. Heck, they're the only ones I know.

While you're editing a certain file with `emacs` or `xemacs`, you can save it with the `[Ctrl]-x [Ctrl]-s` keystrokes. Then to exit, type `[Ctrl]-x [Ctrl]-c`.

pico

The instructions for using `pico` are located on the screen. You save the file by using the `[Ctrl]-o` keystroke (for write-out) and exit with [Ctrl]-x.

As a permanent solution, you probably don't want to use `pico`. It lacks real power. Since I am such a wuss, however, I still have the bad habit of using `pico` once in a while. Why? By pressing `[Ctrl]j` I can get entire paragraphs wrapped into a nice justified block. I don't know how to do that with the other text editors.

vim

Most modern distributions include `vim`, derived from the infamously arcane Unix editor, `vi`. (It stands for **vi** I**m**proved, as a matter of fact.)

Using vim is different in that there are several modes in which you use it. To do actual editing of the files, press `[ESC] i` (one after the other). Then to save it, press `[ESC] : w`. Escape, the colon, and "w" should be keyed in one after the other. Finally, to quit, type `[ESC] : q`. The same rules apply as in previous vim commands.

You can use "w" and "q" at the same time to enable yourself to write to the file and then quit right afterwards. Just press `[ESC] : w q`.

If you don't have `vim` installed, try `vi` instead. In newer Red Hat releases, typing `vi` will actually load `vim`.

Monitoring Your System

An important part of system administration (especially with your own system) is being able to know what's going on.

tail

The program tail allows you to follow a file as it is growing. Most often, I use it to follow /var/log/messages. I do that by typing `tail -f /var/log/messages`. Of course, you can use anything else, including the other logs in /var/log/. Another file you may want to keep an eye out for is /var/log/secure.

If you want to leave that running all the time, I recommend having some sort of terminal program in X, logged in as root through `su`.

Another program you may want to look at is `head`. It monitors the top of the file specified, instead of the bottom.

top

This program shows a lot of stuff that goes on with your system. In the program, you can type:

1. `M` for memory usage information
2. `P` for CPU information
3. `q` to quit

Once you try it, you can see that top shows you the memory usage, uptime, load average, CPU states, and processes.

w

Typing `w` will tell you who is logged in. This can be helpful if you're the only one who uses your computer and you see someone logged in that's not supposed to be.

Another alternative is `who`.

Shutting Down and Rebooting

To shut down your system, type `shutdown -h now`, which tells the shutdown program to begin system halt immediately. You can also tell it to halt the system at a later time, I think, but you'll have to consult the shutdown manual page for that (man shutdown).

To do a reboot, you can either type `reboot` or `shutdown -r`. You can also use the famous Ctrl-Alt-Delete combination to reboot, which you might already be familiar with.

Shutting down and restarting properly (as described above) will prevent your filesystem from being damaged. Filesystem damage is the most obvious of the consequences, but there are probably other things out there that I don't know about. The point is, shut down your system properly.

There are (rare!) cases in which the machine might lock up entirely, and prevent you from being able to access a command prompt. Only then will your last resort be to do a forced reboot (just pressing the restart button on the case).