# AN INTRODUCTION TO THE LINUX COMMAND LINE

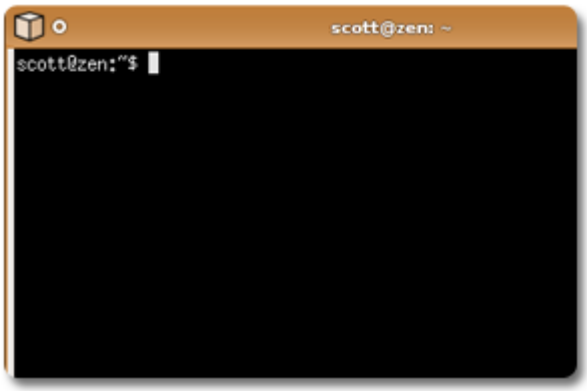
> There are a number of useful commands and utilities for the Linux command line that really take advantage of its power

One persistent, though entirely unfounded myth about Linux is that you need to use the command line to do ... well, just about everything. That's just not true. Window managers -- likeKDE, GNOME, or xfce -- give you a fully-graphical way of carrying out any number of tasks.

That said, even a cursory knowledge of the command line can make your life in Linux much easier. Sometimes, it's just faster and easier to jump out to the command line, perform a task, and then return to the warm coziness of your window manager. Like when? Say, for example, you want to archive a bunch of files in a directory. Using the command line utility zip is often quicker than using a graphical archiving tool.

Anyway, if you want another opinion on this then you definitely should read this article. But since you've read this far, I'm sure you're more than just a little curious about the command line. So, let's get started.

### Getting to the command line




While you can start Linux in command line mode, chances are that you'll pop out to the command line from within a window manager. How you get to the command line will depend on the window manager you're using. Regardless, you'll work at the command line in a*terminal window* which gives you access to the command line.

Usually, you select the terminal program from a menu. If you're using KDE, you can press CTRL-T on the

keyboard and a terminal window will open in the directory that you happen to be in. GNOME doesn't have this feature built-in, but you can download a script that adds it. In xfce, a popular and lightweight window manager, all you need to do is right click in the file manager and a select **Open Terminal Here**.

## Moving around

So, you're at the command line. Now what? Chances are you want to go somewhere. To do that, use the **cd** command, plus the name of the directory to which you want to move. Most often, when you open a terminal window, you're in your home directory (which contains your personal files and settings) -- for example, /home/scott. Let's say you want to move to a directory called pictures, which holds your photos. Just type **cd photos** and you're there.

You can also use **cd** to move into and around subdirectories. Say you're in your home directory, and you want to go todocuments/letters. To do that, type **cd documents/letters** and you're there. If, on the other hand, you want to move to a directory that's outside of /home, that's easy to do. Just type **cd** and the full path to the directory. For example, to a common directory for executable files on your system, type cd / usr / local / bin.

Once you're in a directory, it's simple to move around any directories that are above it. if you're in the directory pictures/cottage, but decide that you want to move one level up to the directorypictures/London2006, just type **cd ../London2006**. The ../ tells the**cd** command to move up one level and then change to the directory that you specify.

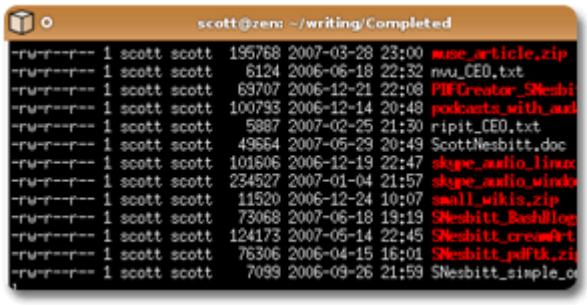
You can use ../ as many times as you need. So, if you want to move from pictures/London2006 to the directory documents/letters type**cd ../../documents/letters**. The double ../ moves you up two levels in the directory structure and the **cd** command puts you in the directory in which you want to go.

## Listing directories and files

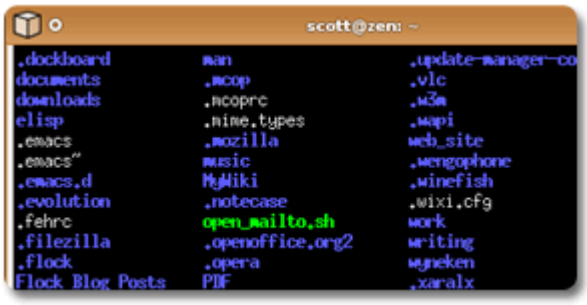
Moving around the command line can get pretty boring pretty quickly. One of the many things that you can do in a terminal window is list the files in a directory. How is this useful? Well, I write a lot. The directories holding what I'm working on and what I've completed fill up quite quickly. By listing the contents of the

directory, I can determine which files to archive and which to move elsewhere (more on this later).

To do the deed, type **ls**, which gives you a multi-column listing of the files in a directory.


If you want a little more detail about the files, type**ls -l**. This command lists the file names, their sizes (in kilobytes), the date on which they were created or modified, who owns them, and the permissions that the files have.




Linux also has a number of hidden files and directories. These mainly contain configuration information for various programs. If you want to view a listing of them, type **ls -a**. As with the ls command, you get a basic listing of both visible and hidden directories but no detailed information.

Of course, you might also want to get a listing of only certain types of files in a directory. You do that by using a wildcard. For example, to see only the OpenOffice.org Writer files in a directory, type **ls *.odt**.

### Copying and moving files

One of the things that you can do as well at the command line as you can in a graphical window manager is shuffle files around. Putting a copy of a file somewhere -- to make it available to all users of your computer or just to have a backup -- is done using the cp command. So, to copy the file Your_Life_in_Web_Apps.pdf in your home directory to the directory eBooks, type **cp Your_Life_in_Web_Apps.pdf eBooks/Your_Life_in_Web_Apps.pdf**. You must always specify the target file name. If you don't, the command won't work. But you can also use a different target name when copying a file. I generally do this when

making a backup copy of a word processor or spreadsheet file that I'm working on.

Moving a file can mean one of two things: removing it from one directory and placing it in another, or renaming the file. Both are done with the **mv** command. To move a file, type **mv** plus the name of the file, followed by the destination directory -- for example, **mvYour_Life_in_Web_Apps.pdf /eBooks**. You can optionally add the name of the destination file.

Renaming a file with the **mv** command uses the same syntax, only this time you must include the new name of the file in the command. Just type **mv <filename> <new_filename>**. I often use this command to make a file name more descriptive or to strip out spaces in the name.

### Deleting files

You often wind up with any number of files that you don't want or no longer need. Instead of trying to click all of them in a graphical file manager, dispose of them at the command line instead. The command for deleting files is **rm**, followed by the name of the file. You can also delete more than one file by typing their names followed by a space -- for example, **rm index.html 404.html banner.gif**.

As with many other Linux command, you can use wildcards with **rm**. For example, you can delete all of the JPEG images in a directory by typing **rm *.jpg**. Remember to use the **rm** command carefully. Once you delete a file at the command line, it's practically impossible to recover it. There's no trash or recycle bin at the command line!

### Conclusion

The Linux command line can be a powerful tool. There are a number of useful commands and utilities for the Linux command line that really take advantage of its power. This TechTip really only pricked the surface of what the command line can actually do. A future TechTip will look at some other useful Linux commands, as well as at some neat little utilities that can make the command line an easier and more flexible place at which to work.

Source: http://www.geeks.com/techtips/2007/techtips-02DEC07.htm