**LINUX BASICS**

<http://www.computerworld.com/article/2598082/linux/linux-linux-command-line-cheat-sheet.html>

**Moving Around the Filesystem:** Commands for moving around the filesystem include the following.

* **pwd:** The pwd command allows you to know the directory in which you're located (pwd stands for "print working directory"). For example, pwd in the desktop directory will show ~/Desktop. Note that the GNOME terminal also displays this information in the title bar of its window.
* **cd:** The cd command allows you to change directories. When you open a terminal, you will be in your home directory. To move around the filesystem, use cd.
  + To navigate to your desktop directory, use cd ~/Desktop
  + To navigate into the root directory, use cd /
  + To navigate to your home directory, use cd
  + To navigate up one directory level, use cd ..
  + To navigate to the previous directory (or back), use cd -
  + To navigate through multiple levels of directories at once, use cd /var/www, for example, which will take you directly to the /www subdirectory of /var.

**Manipulating Files and Folders:** You can manipulate files and folders by using the following commands.

* **cp:** The cp command makes a copy of a file for you. For example, cp *file* foo makes an exact copy of the file whose name you entered and names the copy foo, but the first file will still exist with its original name. After you use mv, the original file no longer exists, but after you use cp, that file stays and a new copy is made.
* **mv:** The mv command moves a file to a different location or renames a file. Examples are as follows: mv *file* foo renames the original file to foo. mv foo ~/Desktop moves the file foo to your desktop directory but does not rename it. You must specify a new filename to rename a file.
  + To save on typing, you can substitute ~ in place of the home directory.
  + Note: If you are using mv with sudo, you will not be able to use the ~ shortcut. Instead, you will have to use the full pathnames to your files.
* **rm:** Use this command to remove or delete a file in your directory. It does not work on directories that contain files.
* **ls:** The ls command shows you the files in your current directory. Used with certain options, it lets you see file sizes, when files where created, and file permissions. For example, ls ~ shows you the files that are in your home directory.
* **mkdir:** The mkdir command allows you to create directories. For example,mkdir music creates a music directory.
* **chmod:** The chmod command changes the permissions on the files listed.
  + Permissions are based on a fairly simple model. You can set permissions for user, group, and world, and you can set whether each can read, write, and/or execute the file. For example, if a file had permission to allow everybody to read but only the user could write, the permissions would read rwxr--r--. To add or remove a permission, you append a + or a - in front of the specific permission. For example, to add the capability for the group to edit in the previous example, you could type chmod g+x file.
* **chown:** The chown command allows the user to change the user and group ownerships of a file. For example, chown jim *file* changes the ownership of the file to Jim.

**Searching and Editing Text Files:** Search and edit text files by using the following commands.

* **grep:** The grep command allows you to search inside a number of files for a particular search pattern and then print matching lines. For example, grep blah *file* will search for the text "blah" in the file and then print any matching lines.
* **sed:** The sed (or Stream EDitor) command allows search and replace of a particular string in a file. For example, if you want to find the string "cat" and replace it with "dog" in a file named pets, type sed s/cat/dog/g pets.

Both grep and sed are extremely powerful programs. There are many excellent tutorials available on using them, but here are a few good websites to get you started:

* [www.panix.com/~elflord/unix/grep.html](http://www.panix.com/~elflord/unix/grep.html)
* [www.itworld.com/Comp/2378/swol-1199-unix101/](http://www.itworld.com/Comp/2378/swol-1199-unix101/)

Three other commands are useful for dealing with text.

* **cat:** The cat command, short for concatenate, is useful for viewing and adding to text files. The simple command cat *FILENAME* displays the contents of the file. Using cat *FILENAME* file adds the contents of the first file to the second.
* **nano:** Nano is a simple text editor for the command line. To open a file, use nano filename. Commands listed at the bottom of the screen are accessed via pressing Ctrl followed by the letter.
* **less:** The less command is used for viewing text files as well as standard output. A common usage is to pipe another command through less to be able to see all the output, such as ls | less.

**Getting Help on the Command Line:** This section provides you with some tips for getting help on the command line. The commands --help and man are the two most important tools at the command line.Virtually all commands understand the -h (or --help) option, which produces a short usage description of the command and its options, then exits back to the command prompt. Try man -h or man --help to see this in action.

Some helpful tips for using the man command include the following.

* **Arrow keys:** Move up and down the man file by using the arrow keys.
* **q:** Quit back to the command prompt by typing q.