Command Line Primer

Directories

`$ pwd`

Show path of the current working directory

`$ cd <directory>`

Change directory to <directory>

`$ cd ..`

Move up one directory.

`$ ls`

List current directory contents

`$ ls -la`

List directory contents with metadata and hidden files

`$ mkdir <directory>`

Create <directory>

Files

`$ rm <file>`

Delete <file>

`$ rm -r <directory>`

Delete <directory> including files

`$ mv <orig-file> <new-file>`

Rename <orig-file> to <new-file>

`$ mv <file> <directory>`

Move <file> to <directory> (possibly overwrite file of same name)

`$ cp <file> <directory>`

Copy <file> to <directory> (possibly overwrite file of same name)

`$ cp -r <directory1> <directory2>`

Copy <directory1> and contents to <directory2> (possibly overwriting contents)

Permissions

`$ chmod 755 <file>`

Change permission of <file> to 755 (execute + read for all users)

`$ chmod 644 <file>`

Perms of <file> to 644 (owner: read-write; everyone: read)

`$ chown <user>:<group> <file>`

Change ownership of <file> to specified <user> and <group>

Output and Edit

`cat <file>`

Display the contents of <file> in the terminal

`less <file>`

Display <file> contents (paginated)

`head <file>`

Display first 10 lines of <file>

`tail <file>`

Display last 10 lines of <file>

`<cmd> > <file>`

Redirect output of <cmd> into <file> (creating file if nonexistent)

`<cmd> >> <file>`

Append output of <cmd> to <file> (creating file if nonexistent)

`<cmd1> | <cmd2>`

Pipe the output of <cmd1> to be the input of <cmd2>

`clear`

Clear the terminal window

`nano <file>`

Open the file in a simple text editor

Processes

`$ ps ax`

Display all running processes

`$ ps aux`

Display all running proceses with associated usernames

`$ top`

Show running processes (interactive – hit 'q' to quit)

`$ kill <pid>`

Kill process with process id <pid>

Search

`$ find <dir> -name "<file>"`

Find all files named <file> in <dir>

`$ grep "<text>" <file>`

Find all instances of <text> in <file>

`$ grep -rl "<text>" <dir>`

Find files containing <text> in <dir>

Network

`$ curl -O <url>`

Download file (via HTTP[S])

`$ ssh <username>@<host>`

Connect to <host> with <username> securely over SSH

`$ scp <file> <user>@<host>:/remote/path`

Copy local <file> to remote host location /remote/path

Help and Documentation

Type `man <cmd>` to see the manual page for any command (try `man cp` as an example)

Adjusting Permissions

In some cases it is necessary to run programs (including disk imaging programs) with elevated administrator ("superuser") permissions. In such cases, these programs may output files that can only be read, deleted, or modified by an administrator.

In Linux distributions, the "root" account is the primary administrator. Depending on how the system is configured, other users may be allowed "do-as-superuser" or "super-user-do", or (using the command line syntax) "sudo" privileges, allowing them to execute commands as if they were the "root" user.

When a file with restricted permissions is produced, you may see that the icon includes a padlock, or has some other indication that you cannot modify, move, or delete it. Imagine we have just created the file `image1.raw` on the Desktop of your user, and it is marked with a graphical padlock. Opening a terminal, we examine the permission using the "ls" command:

```
yourusername@ubuntu:~$ ls -la /home/yourusername/Desktop/image1.raw
-rw-r--r-- 1 root root 0 Dec 11 22:04 /home/yourusername/Desktop/image1.raw
```

This indicates that the owner of the file is "root", the associated group is "root", and the permissions are "read+write" for the root user (-rw), "read" for any user in the "root" group (-r-), and "read" for all system users (-r-).

We can change the owner and group for this file as follows:

```
sudo chown yourusername:yourusername image1.raw
```