**Common Linux Commands**

**Navigating Linux:**

**/** (refers to the root directory on the server)

**./** (the current directory that you are in)

**../** (parent directory of your current directory)

**pwd** (shows what you current directory is - giving the full path)

**ls** (lists all the files in your current directory)

**ls -al** (lists filenames + information)

**ls -alR** (lists filenames + information in all subdirectories)

**ls -alR | more** (lists filenames + information in all subdirectories,

**ls -alR > result.txt** (lists filenames + information in all subdirectories

and outputs the results to a file instead of the screen)

**ls \*.html** (lists all files ending with .html)

**ls -al /home/usr/bob/** (lists files + info for /home/usr/bob)

**cd** (changes you to a new directory)

**cd images**

**cd /** (changes you to the root directory)

**cd /home/usr/images**

**cd ..** (goes back one directory)

**Moving, Copying, and Deleting Files:**

**mv [old name] [new name]** (move/rename a file)

**cp [filename] [new filename]** (copy a file)

**rm [filename]** (delete a file)

**rm \*** (delete all files in your current directory)

**rm \*.html** (delete all files ending in .html

in your current directory)

**Creating, Moving, Copying, and Deleting Directories:**

**mkdir [directoryname]** (creates a new directory)

**ls -d \*/** (lists all directories within current directory)

**cp -r [directoryname] [new directoryname]** (copy a directory and all

files/directories in it)

**rmdir [directoryname]** (remove a directory if it is empty)

**rm -r [directoryname]** (remove a directory and all files in it)**Searching Files and Directories**

**find / -name [filename] -print** (search the whole server for a file)

**find . -name [filename] -print** (search for a file starting with

the current directory)

**find / -name [directoryname] - type d -print**

(search the whole server for a directory)

**locate [filename]** (search the whole server for a file using

an indexed database)

**updatedb** (update the file database)

**grep [text] [filename]** (search for text within a file)

**sed s/[oldtext]/[newtext]/g [filename]** (search file and replace all occurrences

of [oldtext] with [newtext]

**File and Directory Permissions**

There are three levels of file permission: read, write, and execute. In

addition, there are three groups to which you can assign permission,

The file owner, the user group, and everyone. The command chmod followed

by three numbers is used to change permissions. The first number is

the permission for the owner, the second for the group, and the third

for everyone. Here is how the levels of permission translate:

0 = --- (no permission)

1 = --x (execute only)

2 = -w- (write only)

3 = -wx (write and execute)

4 = r-- (read only)

5 = r-x (read and execute)

6 = rw- (read and write)

7 = rwx (read, write, and execute)

**chmod** Change access permissions for file/s or directories

SYNTAX*:*

**chmod [OPTION]... MODE[,MODE]... FILE...**

**chmod [OPTION]... NUMERIC\_MODE FILE...**

OPTIONS

**-v, - -verbose** output a diagnostic for every file processed

**-c, - -changes**  like verbose but report only when a change is made

**-R, - -recursive**  change files and directories recursively

**Numeric mode:**

From one to three octal digits; any omitted digits are assumed to be leading zeros.

**Examples**  
 Allow read permission to everyone: **chmod 444 *file***  
 Make a file readable and writable by the group and others: **chmod 066 *file***   
 Allow everyone to read, write, and execute the file: **chmod 777 *file***

**Symbolic Mode**

The format of a symbolic mode is [**ugoa**...][[**+-=**][**rwxXs­t**...]...][,...].   
  
 Multiple symbolic operations can be given, separated by commas. A combination of the letters ugoa controls which **users** access to the file will be changed: The user who owns it (u), users in the files group (g), Other users not in the files group (o), All users (a). If none of these are given, the effect is as if a were given, but bits that are set in the umask are not affected. all users (a) is effectively user + group + others  
  
 The operator + causes the permissions selected to be added to the existing permissions of each file; - causes them to be removed; and = causes them to be the only permissions that the file has.  
  
 The letters **rwxXstugo** select the new **permissions** for the affected users: Read (r), Write (w), Execute (or access for directories) (x), Execute only if the file is a directory or already has execute permission for some user (X), Set user or group ID on execution (s), Save program text on swap device (t), The permissions that the user who owns the file currently has for it (u), The permissions that other users in the files group have for it (g), Permissions that other users not in the files group have for it (o).

**Examples** Deny execute permission to everyone: **chmod a-x *file***  
 Allow read permission to everyone: **chmod a+r *file***  
 Make a file readable/writable by the group/others: **chmod go+rw *file***  
 Allow everyone to read, write, and execute the file: **chmod a=rwx**   
  
 Notes: When chmod is applied to a directory: read = list files in the directory  
 write = add new files to the directory, execute = access files in the directory