### cmp

Compares two files on a, byte by byte basis. Returns the first position (byte number and line) where the files differ.

### cut

cut -c3-7 filename

cut -c3,7 filename

cut -d ":" -f2 accounts

### diff

Compares two files on a line by line basis. Shows how the two files differ.

### echo

echo –n omits new line

echo –e lets you see escape characters

### grep

-v, -i, -w, -c

Inverse, case insensitive, match word, count

### sort

sort –t “:” –k1 file1

sort –r reverse

### tr

tr “:” “b ” replace : with b

[:lower:] [:upper:]

tr –d h deletes all h

tr –s b shrinks

### uniq

uniq –c myfile count how many times repeated and print that line

uniq –d myfile display the duplicated lines

### wc

-l, -w, -c, -m

Lines, words, bytes, characters

### chown

### du

Displays a list of directories and how much disk space, in blocks (512 characters) they occupy.

### file - Show the type of files

file \* Lists the files and file types.

### find – Find Files or Directories

find myFile finds myFile in the current directory

find . -name myFile finds myFile in current directory and its sub-folders

. indicates the current directory, which is the default

find . -size -10k finds any file less than 10k in size

find . -size +10k finds any file over 10k in size

find . -type f -size +10k finds any file over 10k. Type uses f for file, d for directory, l for link

find /usr /bin -name myFile finds an item in specific directories (item – file/directory)

find . -mtime -2 finds items modified during the past 2 days

find . -mmin -60 finds items modified in the past 60 minutes

find . -mmin +60 finds items modified more than 60 minutes ago

find . -name myFile –print -print is the default action and not always required

find . -name "The\*" -exec rm {} \; execute rm on all files starting with “The” in the current directory and every sub-directory.

find . -name "The\*" -exec rm -i {} \; an interactive version

### ln – link files or directories

ln targetfile linkname

ln –s targetfile linkname

### ls – List file

ls lists files

ls –l long listing, providing more information including permissions

ls –i inode number is included

ls –a lists all files, including hidden files.

ls –r reverse. Sorts listing in reverse order.

ls –R recursive, including subdirectories

-l, -i, -a, –r, -R

Long, inode, all files, reverse sort, recursive.

Pwd print working directory

Remove

Rm –r -i

### stat

Stat –c%s filename – filesize in bytes

Stat –c%i filename – inode of filename

<<< string

<< block of text

< file

Basename just name of file

Dirname directory name

Readlink –m $blah full path of file

## Process Commands

If a process is running in the foreground then it is the current job. When a process is running in the foreground, the system expects any user input to be handled by the current job.

### &

You can also start command running in the background using &.

### bg

To resume a stopped job so it runs in the background, use bg.

background – You can change stopped jobs to running in the background using bg

### CTRL-c

CTRL-c will terminate the current job. After CTRL-c, the job will not appear in the jobs list and not appear in ps.

### CTRL-z

To suspend the current job, use CTRL-z. After CTRL-z, the job will appear as "stopped" in the jobs list, and the process will appear in ps.

### fg

fg will resume a stopped job so it runs in the foreground (as the current job).

fg can also bring a job that is running in the background to the foreground.

### jobs

To list all jobs, use the jobs command. It will list stopped jobs and jobs running in the background.

### kill

To kill jobs, use: kill %jobnumber

To kill a process use: kill pid

### ps

List processes.

### sleep

Causes the script to pause for a specified number of seconds.

## System Commands

### clear

Clears the screen and places the cursor at the top of the screen.

### date

Retrieves the system date and time

export - Export command lets you create your own environment variables. Example:  
MYENV\_VAR=abc  
export MYENV\_VAR

### env

Lists the environment variables.

See [Environment Variables](#_Performing_Calculations)

### eval

### lsof

Lists processes and the files the processes have open.

### printenv

Lists the environment variables (Same as env)

### set

set Lists all the shell and environment variables and functions

set andy bruce charlie Sets $1, $2, and $3 to andy, bruce, and charlie, respectively

set –x Turns on trace mode

### sh

Runs a script in the default shell.

sh myscript

sh –x myscript Runs the script with xtrace (tracing statements).

### users

Shows who is logged on.

### w

Shows who is logged on and what they are doing.

### which cmd

Shows the full path of the command, cmd.

### who

Shows who, is logged on.

### whoami

Shows the userid that is currently logged on to the session.

### xargs

xargs takes the standard input from a pipe and uses that input as the command line argument(s) for another command.

echo /home | ls

Now try the following:

$ echo "/home" | xargs ls