Basic Commmands

#ls (List the contents of any directory)
#ls -l (Long listing - detailed report of available files)
#ls -la (Long listing including hidden files)
#pwd (Shows your present working directory)
#cd .. (change directory - moves you to one level up)
#cd /var/tmp (change directory to the specified path)
#cd (Bring back to your home directory)
Ctrl + L (Clear Screen)
Shift + Pg up Key (scroll your screen upwards)
Shift + Pg down Key (Scroll your screen downwards)
#cd - (Switches u between your current & previous directory)
#touch file1 (creates an empty file)
#mkdir dir1 (Creates new directory)
#cd dir1 (change directory to dir1)
#pwd
#cat > file2 (Create new file with data)
Enter the data and press control + d to save the contents
#cat file2 (check the data inside the given file)
#cat >> file2 (Appends the new data in previously created file)
Enter the data, press enter and ctrl + d
#cat file2

@copy@

#cp file2 file3 (make a copy of file2 with new name file3 including the contents of file)
#cp file2 /tmp (make a copy of file2 in /tmp directory)
#cp file2 /tmp/backup_file2 (make a copy of file2 in /tmp directory with new name)
#cp -r dir1 dir2 (-r switch is used to copy a directory)
#cp -r dir1 /tmp/backup_dir1 (copy directory & rename)

@move@

#mv dir1 /var/tmp/ (Moves the dir1 to specified path)
#mv file1 /tmp/backup_file1 (Move and rename)

@rename@

#mv old_name new_name (same mv command is used for renaming)

@remove/delete@

#rm file1 (delete regular file)
#rm -r dir1 (deletes the directory)
#rm -rf dir2 (deletes all data in directory forcefully, use carefully)

ITE

Basic Directory Structure

Directory Architecture

Directories inside the /
/boot - (as name suggests, file under this directory are used for booting purpose)
/home - (Contains home directory of normal users)
/root - (Home directory user root)
/etc - (All sort of configurations files are stored into this directory)
/bin - (bin refers to binary files, binary files are executables similar to exe files in windows)
/sbin - (Binary related with user root are stored into this directory, normal user cannot access the sensative commands in linux system)
/dev - (In linux every thing is file, Files related with connected hardware devices in your system, are stored into this directory[Block special & character special files])
/lib - (Contains Library files)
/media - (Used to access CD & Pendrives)
/mnt - (It is used for temporary mounting)
/proc - (Live information about hardware i.e. RAM,CPU etc)
/usr - (Usr is treated as program files folder in windows)
/var - (Contains the information related to servers ie. web,ftp etc)

Basic VIM editor

Vi is powerfull editor in linux used to edit file.
VIM is upgraded version of vi we use currently to edit files.

#vim filename

There are 3 modes in vim editor :

1)Command Mode
2)Insert Mode
3)Exit Mode

Command Mode :-

This mode is used to run command in vim editor like copy, paste, search, delete, undo etc.
Search in Vim -
/<text> - search the <text> downwards in file

n - continue search in same direction
N - continue search in opposite direction
?<text> - search upwards in file
n - continue search in same direction
---
u - used to undo the last change's
---

yank(copy) | cut (delete)
---

word - yw	dw
line - yy	dd
5line - 5yy	5dd
25line- 25yy	25dd

put (paste)
p - used to paste the data below cursor for line oriented data
P - used to paste the data above cursor

Insert Mode :-
It is Used to enter or edit text and for cursor movement within file
i - insert
a - ?
o - ?
I - ?
O - ?
A - ?

<Esc> is used to exit from insert mode to command mode

Exit Mode :-
This mode is used to save and exit the file
:q - quit
:wq - write & quit
:w - write only
:q! - exit forcefully and abandon changes
:wq! - save & exit forcefully

--------End of Topic----------
Class work after 2 times practice of vim editor :-
1) Search how to remove the highlights on text we are serching in file.
2) Insert the line number before every line
3) open two files in vim editor at the same time using split window inside the vim editor and work on both of them.

Find Command in Linux

Find is used to search the data anywhere in system :

#find / -name passwd {Command will search file named passwd in / , remember this command will exactly match the case of search value, you have entered}

#find / -name *.jpg {command will search all the files in your system having extension .jpg}

#find / -iname passwd {Command will search the value "passwd" in capital or small letters in whole system, including all mounted partitions}

#find / -user username {command will search all the data belongs to username user in whole system }

#find / -size +10M {Command will search all the files in system those are greater then 10 MB in size in whole system, you can search in K,M,G}

#find / -size -10M {Command will search all the files in system those are smaller then 10 MB in size, This command is practically not usable as maximum files in system is below 10M}

#find / -size 10M {Command will search the files of exact 10 MB}

#find /home -ctime +3 {Command will search the data "created" before then last 3 days}

#find /home -atime -3 {Command will search the data "accessed" within last 3 days}

#find /home -mtime 3 {Command will search the data "modified" exactly 3 days before}

Redirection

#ls -l / {Command will return Standard Output on Monitor}

Redirection Standard Output for any other use :-

#ls -l / > stdout [1> and > is same] {This command will redirect to output of ls -l Command to the file stdout, you will not able to see anything on screen after command.

#cat stdout { Now this command will show the output of previous command as the output of ls -l / command is stored permanently in stdout file}

Appending the Standard Output :-

#ifconfig > stdout
#ls -l / >> stdout {This command will append the output of ls -l command to file stdout.

Redirecting Standard Error

Assuming we are adding the user named motorola and his primary is google, but google group doesn't exist in machine, then useradd command will return the error, and we will store the standard error generated by command.

#useradd -g google motorola {This command will generate the error}
#useradd -g google motorola 2> error {by this command any output generated by useradd command is stored into file "error".

Redirect Standard Output & Error seperatly in single command

$find / -iname passwd > stdout1 2> stderr1 {Command will store the output in file stdout1 and error in stderr1 file seperately}

$find / -iname passwd &> stdout_err {Command will store the output and error in single file}

Using Pipe Operator

| (pipe) Operator is used to send the output on one command to another command. For example if we run
#ls -l /etc

Above command will generate a very long output and we are not able to see the output in one screen, then using | operator, we will redirector the outpur of ls command to more command :

#ls -l /etc/ | more
This will show the output of /etc directory in page wise manner.

Another example :
#cat /etc/passwd {this will show the output of /etc/passwd file in small alphabets}

#cat /etc/passwd | tr 'a-z' 'A-Z' {Command will convert the small letters into capital letters as the output of /etc/passwd is truncated into capital letters using tr command}

#cat /etc/passwd | tr 'a-z' 'A-Z' | more {Command will send the output of cat command to tr and tr command's output will be sended to more command}

#cat /etc/passwd | tr 'a-z' 'A-Z' > capspasswd {Finally command will store the output in capspasswd file in capital letters.

Using grep command :-

#cat /etc/passwd | grep -i root {Grep command is used to grep the required output from the long files, in this command we are trying to capture the lines where word "root" is used. -i switch is used to ignore the case senstivity.

#cat /etc/httpd/conf/httpd | grep virtualhost
#cat /etc/httpd/conf/httpd | grep -i virtualhost { see the difference in above last two commands}

---
diff

Create two similar files of 3 or 4 lines and check the differences between both the files
#diff file1 file2 >> track_differences

---

tee Command

#ifconfig | tee test12 {this command will store the output of ifconfig command as well display the output of ifconfig on screen }

Standard Input

echo "redhat" | passwd --stdin student

Input & Output Diagram and then 2>&1

2>&1 is used to channelize the data via pipe i.e. stderr is redirected via pipe to stdout and then the data is stored in a file.

Redirection

#ls -l / {Command will return Standard Output on Monitor}

Redirection Standard Output for any other use :-

#ls -l / > stdout [1> and > is same] {This command will redirect to output of ls -l Command to the file stdout, you will not able to see anything on screen after command.

#cat stdout { Now this command will show the output of previous command as the output of ls -l / command is stored permanently in stdout file}

Appending the Standard Output :-

#ifconfig > stdout
#ls -l / >> stdout {This command will append the output of ls -l command to file stdout.

Redirecting Standard Error

Assuming we are adding the user named motorola and his primary is google, but google group doesn't exist in machine, then useradd command will return the error, and we will store the standard error generated by command.

#useradd -g google motorola {This command will generate the error}
#useradd -g google motorola 2> error {by this command any output generated by useradd command is stored into file "error".

Redirect Standard Output & Error seperatly in single command

$find / -iname passwd > stdout1 2> stderr1 {Command will store the output in file stdout1 and error in stderr1 file seperately}

$find / -iname passwd &> stdout_err {Command will store the output and error in single file}

Using Pipe Operator

| (pipe) Operator is used to send the output on one command to another command. For example if we run
#ls -l /etc
Above command will generate a very long output and we are not able to see the output in one screen, then using | operator, we will redirector the outpur of ls command to more command :

#ls -l /etc/ | more
This will show the output of /etc directory in page wise manner.

Another example :
#cat /etc/passwd {this will show the output of /etc/passwd file in small alphabets}

#cat /etc/passwd | tr 'a-z' 'A-Z' {Command will convert the small letters into capital letters as the output of /etc/passwd is truncated into capital letters using tr command}

#cat /etc/passwd | tr 'a-z' 'A-Z' | more {Command will send the output of cat command to tr and tr command's output will be sended to more command}

#cat /etc/passwd | tr 'a-z' 'A-Z' > capspasswd {Finally command will store the output in capspasswd file in capital letters.

Using grep command

#cat /etc/passwd | grep -i root {Grep command is used to grep the required output from the long files, in this command we are trying to capture the lines where word "root" is used. -i switch is used to ignore the case senstivity.

#cat /etc/httpd/conf/httpd | grep virtualhost
#cat /etc/httpd/conf/httpd | grep -i virtualhost { see the difference in above last two commands}

---

diff

Create two similar files of 3 or 4 lines and check the differences between both the files
#diff file1 file2 >> track_differences

head & tail

#head -n 10 /etc/passwd
#tail -n 10 /etc/passwd
ls -lt | head -n 10 {This command will show last 10 modified files in /etc directory}

---

tee Command

#ifconfig | tee test12 {this command will store the output of ifconfig command as well display the output of ifconfig on screen }

Standard Input

echo "redhat" | passwd --stdin student

Input & Output Diagram and then 2>&1

2>&1 is used to channelize the data via pipe i.e. stderr is redirected via pipe to stdout and then the data is stored in a file.