Government Girls' Polytechnic, Bilaspur

Name of the Lab: **Operating system Lab**

Practical: **Linux Lab**

Class: **5th sem(CS), 4th sem(IT)**

Teachers Assessment:20 End Semester Examination:50

EXPERIMENT NO:-1

1. **OBJECTIVE :-** Practice on stty command.

2. **HARDWARE & SYSTEM SOFTWARE REQUIRED :-**
   Intel P-III Processor and above
   64 MB RAM and above
   HDD 4.3 GB Hard Disc and above

3. **SOFTWARE REQUIRED :-** Linux Operating System(Red Hat)

4. **THEORY: -**
   In this command we have change and print terminal line settings.

5. **PROGRAM INPUTS & OUTPUT :-**
   I/P: stty[-f DEVICE:- -- file=DEVICE][-a:--all]
   O/P: Print all current setting in human readable from.

6. **OBSERVATION:** Studied above all commands and run successfully.

EXPERIMENT NO:-2

1. **OBJECTIVE: -** Study of password command.

2. **HARDWARE & SYSTEM SOFTWARE REQUIRED :-**
   Intel P-III Processor and above
   64 MB RAM and above
   HDD 4.3 GB Hard Disc and above

3. **SOFTWARE REQUIRED :-** Linux Operating System(Red Hat)

4. **THEORY :-**
   In this command we have change user's password or password attribute.
   Syntax: $passwd [option]

5. **PROGRAM INPUTS & OUTPUT :-**
   O/P: Lock user's account
   I/P: passwd –l --lock
   O/P: Unlock user's account
   I/P: passwd –u –unlock
   O/P: User can't change password
   I/P: passwd –c –cannot-change

6. **OBSERVATION:** Studied above all commands and run successfully.

EXPERIMENT NO:-3

1. **OBJECTIVE: -** Study of who, who am i, tty, date and cal commands.

2. **HARDWARE & SYSTEM SOFTWARE REQUIRED :-**
   Intel P-III Processor and above
   64 MB RAM and above
   HDD 4.3 GB Hard Disc and above

3. **SOFTWARE REQUIRED :-** Linux Operating System(Red Hat)

4. **THEORY :-**

   Who: It display the list of currently logged in users.
   Syntax: $who [option]
   Whoami: It displays the current user name.
   Syntax: $whoami [option]
   TTY: It prints the file name of the terminal connected to standard input.
   Syntax: $tty [option]
   Date: This command display current system date and time.
   Syntax: $date [option] format
   Cal: It displays the current month calendar.
   Syntax: $cal [month] [year]

5. **PROGRAM INPUTS & OUTPUT :-**

   1. $who
   Abhi tty1 jan1 04:00
   Anu tty2 jan2 04:00

   2. $whoami
   Abhi tty1 jan1 04:00

   3. $tty
   /dev/console

   4. $date
   Mon jan1 04:00 IST 2011

   5. $cal 8 2002

   August 2002

S	M	Tu	W	Th	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

6. **OBSERVATION:** Studied above all commands and run successfully.

EXPERIMENT NO:-4

1. **OBJECTIVE: -** Executing commands in background.

2. **HARDWARE & SYSTEM SOFTWARE REQUIRED:-**
   Intel P-III Processor and above
   64 MB RAM and above
   HDD 4.3 GB Hard Disc and above

3. **SOFTWARE REQUIRED: -** Linux Operating System (Red Hat)

4. **THEORY:-**
   This will display the character or word followed this command in the banner like format, banner prints character in a sort of ASCII art poster, for example to print wait in big letters. I will type banner wait at UNIX command line or in my script. This is how it will look.

   Syntax: $banner [character/string]

5. **PROGRAM INPUTS & OUTPUT: -**
   $banner wait

   Wait

6. **OBSERVATION:** Studied above all commands and run successfully.

EXPERIMENT NO:-5

1. **OBJECTIVE: -** Study of ps, kill commands.

2. **HARDWARE & SYSTEM SOFTWARE REQUIRED:-**
   Intel P-III Processor and above
   64 MB RAM and above
   HDD 4.3 GB Hard Disc and above

3. **SOFTWARE REQUIRED: -** Linux Operating System (Red Hat)

4. **THEORY: -**
   Ps: Basically Linux operating system is process based. Every process has unique id. We can display currently running process.

   Syntax: $ps [option]

   Kill: This command is called canceling process. Using this command we can terminate specified process.

   Syntax: $kill [signal/pid]

5. **PROGRAM INPUTS & OUTPUT: -**

1. $ps

PID	TTY	TIME	CMD
999	pts/0	00:00	bash
1638	pts/0	00:00	ps

2. $ps -e

PID	TTY	TIME	CMD
1	?	00:04	

3. $kill 1638

It terminate the currently process 1638.

6. **OBSERVATION:** Studied above all commands and run successfully.

EXPERIMENT NO:-6

1. **OBJECTIVE: -** Listing the files in a directory using all options to ls.

2. **HARDWARE & SYSTEM SOFTWARE REQUIRED:-**
   Intel P-III Processor and above
   64 MB RAM and above
   HDD 4.3 GB Hard Disc and above

3. **SOFTWARE REQUIRED: -** Linux Operating System (Red Hat)

4. **THEORY: -**
   Linux operating system support to display all files and directories in different format. They are the following commands:

   $ls: it displays all file and directory names.
   Syntax: $ls [option][file name]
   $ls –l: it displays long listing to file or directory.
   $ls –a: it display all above hidden file.
   $ls –i: it displays all files and directories with a specified inode number.
   $ls –r: it displays all files and directories in reverse order of the basis of the file.
   $ls –t: it displays all files and directories on the basis of created time.
   $ls –F: it display all files, directories, executable file and symbolic files.

6. **PROGRAM INPUTS & OUTPUT: -**

   $ls -F
   NIT\ - Directory file
   NEO* - Executable file
   UNIX@ - Linked file

6. **OBSERVATION:** Studied above all commands and run successfully.

EXPERIMENT NO:-7

1. **OBJECTIVE: -** Creating sub-directories.

2. **HARDWARE & SYSTEM SOFTWARE REQUIRED:-**
   Intel P-III Processor and above
   64 MB RAM and above
   HDD 4.3 GB Hard Disc and above

3. **SOFTWARE REQUIRED: -** Linux Operating System (Red Hat)

4. **THEORY: -**
   Directory is a area where we are storing all file and sub directories.
   Mkdir: Using this command we are able to create directories.
   Syntax: $mkdir [option] [dir name]
   Rmdir: Using this command we are able to remove directories.
   Syntax: $rmdir [option] [dir name]

5. **PROGRAM INPUTS & OUTPUT:-**
   1. $mkdir NEO
   This command creates a directory with name NEO.
   2. $rmdir NEO
   This command deletes a directory with name NEO.

6. **OBSERVATION:** Studied above all commands and run successfully.

EXPERIMENT NO:-8

1. **OBJECTIVE: -** Changing the mode of a file/directory.

2. **HARDWARE & SYSTEM SOFTWARE REQUIRED:-**
   Intel P-III Processor and above
   64 MB RAM and above
   HDD 4.3 GB Hard Disc and above

3. **SOFTWARE REQUIRED: -** Linux Operating System (Red Hat)

4. **THEORY: -**
   Linux operating system support different user and different permission. In Linux each and every file may have 3 permission. They are read[r], write[w], and execute[x]. We can change this permission for a directory or file. Here changing means adding or removing particular or set of permissions.
   If you want, you can use chmod to change the protection on your home directory so that other cannot read or execute your home directory.
   Chmod: Its stands for change mode. Using this command we are able to add the permission, remove the permission and assigning the permission.
   Syntax: $chmod [option] [file name]

5. **PROGRAM INPUTS & OUTPUT:-**
   1. Write a command adds execute permission to the owner of the file.
   Ans: $chmod u+x nit
   2. Write a command adds write permission to group remove read permission to others.
   Ans: $chmod g+w, o-r nit

6. **OBSERVATION:** Studied above all commands and run successfully.

EXPERIMENT NO:-9

1. **OBJECTIVE: -** Changing the owner of a file/directory.

2. **HARDWARE & SYSTEM SOFTWARE REQUIRED:-**
   Intel P-III Processor and above
   64 MB RAM and above
   HDD 4.3 GB Hard Disc and above

3. **SOFTWARE REQUIRED: -** Linux Operating System (Red Hat)

4. **THEORY:-**
   Linux operating system support different user and different permission. In Linux each and every file may have 3 permission. They are read[r], write[w], and execute[x]. We can change this permission for a directory or file. Here changing means adding or removing particular or set of permissions.
   If you want, you can use chmod to change the protection on your home directory so that other cannot read or execute your home directory.
   Chown: Its stands for change owner. Using this command we are able to change the existing owner of the file.
   Syntax: #chown [new owner name] [file name]

5. **PROGRAM INPUTS & OUTPUT:-**
   #chown nit kit
   This command is change the existing file nit to kit.

6. **OBSERVATION:** Studied above all commands and run successfully.

EXPERIMENT NO:-10

1. **OBJECTIVE: -** Study of file processing commands.

2. **HARDWARE & SYSTEM SOFTWARE REQUIRED:-**
   Intel P-III Processor and above
   64 MB RAM and above
   HDD 4.3 GB Hard Disc and above

3. **SOFTWARE REQUIRED: -** Linux Operating System (Red Hat)

4. **THEORY:-**
   Some more file processing command in Linux o/s:
   More: it displays a specified file information page wise.
   Syntax: $more [option] [file name]
   Less: using this command we are able to display one screen full page at a time.
   Syntax: $less [option] [file name]
   Head: using this command we are able to display the requested lines from a specified files. By default head command display 10 lines.
   Syntax: $head [option] [file name]
   Tail: using this command we are able to display the requested lines from a specified files. By default tail command display last 10 lines.
   Syntax: $tail [option] [file name]

5. **OBSERVATION:** Studied above all commands and run successfully.

EXPERIMENT NO:-11

1. **OBJECTIVE: -** Display date using various formats.

2. **HARDWARE & SYSTEM SOFTWARE REQUIRED:-**
   Intel P-III Processor and above
   64 MB RAM and above
   HDD 4.3 GB Hard Disc and above

3. **SOFTWARE REQUIRED: -** Linux Operating System (Red Hat)

4. **THEORY: -**
   Date: it is display current system date and time.
   Syntax: $date [option] [format]

5. **PROGRAM INPUTS & OUTPUT:-**
   $date +%d: it displays the current system date.
   $date +%m: it display the current system month
   $date +%y: it display the current system year in short form.
   $date +%H: it display the current system time in number of hours.
   $date +%s: it display the current system time in number of second.

6. **OBSERVATION:** Studied above all commands and run successfully.

EXPERIMENT NO:-12

1. **OBJECTIVE: -** User to user communication using communication commands.

2. **HARDWARE & SYSTEM SOFTWARE REQUIRED:-**
   Intel P-III Processor and above
   64 MB RAM and above
   HDD 4.3 GB Hard Disc and above

3. **SOFTWARE REQUIRED: -** Linux Operating System (Red Hat)

4. **THEORY:-**
   Linux operating system two types of communication they are
   1. On-line communication 2. Off-line communication

   1. On-line communication: it is possible between currently login users. Using write command we are able to communicate with a specified are or users.
   Syntax: $write [option] [user name]

   2. Off-line communication: in this communication any one should logged in. Using mail command we are able to sends the mail and receive the mail.
   Syntax: $mail [option] [user name]

5. **PROGRAM INPUTS & OUTPUT: -**
   $write nit
   Hi how are you?
   (Then the typed message will send to nit)
   $mail nit
   Subject:
   Type the mail
   ……
   …….
   ……..
   Ctrl+d

6. **OBSERVATION:** Studied above all commands and run successfully.

EXPERIMENT NO:-13

1. **OBJECTIVE: -** Study of VI editor.

2. **HARDWARE & SYSTEM SOFTWARE REQUIRED:-**
   Intel P-III Processor and above
   64 MB RAM and above
   HDD 4.3 GB Hard Disc and above

3. **SOFTWARE REQUIRED: -** Linux Operating System (Red Hat)

4. **THEORY:-**
   A Linux operating system we have set of editors. All this editors are ASCII text base.
   VI: it has all the features of Ed and ex. It is powerful tools to develop different shell programming. It has the following modes:
   1. Command mode: it is by default mode whenever we hit a key from keyboard the corresponds action to display on the screen.
   2. Insert mode: which convert to command mode to insert mode. When the editor is insert mode we are able to inter the required data as a input.
   3. Ex command mode: (:) it is prompt of ex mode. In this command using, w it means save the current file, and: WD it means save the current file and quit.

6. **OBSERVATION:** Studied above all commands and run successfully.

EXPERIMENT NO:-14

1. **OBJECTIVE: -** Modes of VI.

2. **HARDWARE & SYSTEM SOFTWARE REQUIRED:-**
   Intel P-III Processor and above
   64 MB RAM and above
   HDD 4.3 GB Hard Disc and above

3. **SOFTWARE REQUIRED: -** Linux Operating System (Red Hat)

4. **THEORY:-**

VI editor have tree modes:

1. Command mode: it is by default mode whenever we hit a key from keyboard the corresponds action to display on the screen.

2. Insert mode: which convert to command mode to insert mode. When the editor is insert mode we are able to inter the required data as an input.

3. Ex command mode: (:) it is prompt of ex mode. In this command using, w it means save the current file, and: WD it means save the current file and quit.

6. **OBSERVATION:** Studied above all commands and run successfully.

EXPERIMENT NO:-15

1. **OBJECTIVE: -** Creating and saving files using VI.

2. **HARDWARE & SYSTEM SOFTWARE REQUIRED:-**
   Intel P-III Processor and above
   64 MB RAM and above
   HDD 4.3 GB Hard Disc and above

3. **SOFTWARE REQUIRED: -** Linux Operating System (Red Hat)

4. **THEORY: -**
   Creating file: There are many way to load editor in the computer memory but the general syntax to start vi is
   $vi [option] [file name]
   Saving file: during entering of the text, always remember to save your file otherwise if power goes off, you will lose all the currently edited text. For saving the file for the first time, give the following keystroke command at the command mode:
   : w poly
   The text file will be saved with name poly and you can still continue editing the text.
   If you don't want to save the file and want to quit vi give following command:
   : q!

5. **PROGRAM INPUTS & OUTPUT:-**
1. $vi –r poly
Creating vi with text file and option. Recover the crashed file with option –r .
2. :W poly
Saves the file with existing name and resumes editing.

6. **OBSERVATION:** Studied above all commands and run successfully.

EXPERIMENT NO:-16

1. **OBJECTIVE: -** Cursor movement commands.

2. **HARDWARE & SYSTEM SOFTWARE REQUIRED:-**
   Intel P-III Processor and above
   64 MB RAM and above
   HDD 4.3 GB Hard Disc and above

3. **SOFTWARE REQUIRED: -** Linux Operating System (Red Hat)

4. **THEORY:-**
   In Linux h j k l are cursor movement command.
   The symbol ^ before a letter means that <ctrl> key should be held down while the letter key is pressed.

5. **PROGRAM INPUTS & OUTPUT:-**
   * j or<return>
     [Or down-arrow] move cursor down one line.
   * k [or up-arrow] move cursor up one line.
   * h or<backspace>
     [Or left-arrow] move cursor left one character.
   * l or<space>
     [Or right-arrow] move cursor right one character.

6. **OBSERVATION:** Studied above all commands and run successfully.

EXPERIMENT NO:-17

1. **OBJECTIVE: -** Cut and paste commands.

2. **HARDWARE & SYSTEM SOFTWARE REQUIRED:-**
   Intel P-III Processor and above
   64 MB RAM and above
   HDD 4.3 GB Hard Disc and above

3. **SOFTWARE REQUIRED: -** Linux Operating System (Red Hat)

4. **THEORY:-**
   Cut: it brinks a specified field are character and delimiter. Here –f for field, -c for character, -d for delimiter etc. using this filter we are able to retrieve specified Colum data's from a specified table.
   Syntax: $cut [option] [file name] or [table name]
   Paste: using this command we can display specified file information side by side.
   Syntax: $paste [option] [file 1] [file 2]

5. **PROGRAM INPUTS & OUTPUT:-**

Example of cut command:

$cat> stud

s.n	stu.name	marks
1	Abhi	10
2	Ravi	10

$cut –f3 stud

Marks
10
10

Example of paste command:

$cat>one
HI, HOW ARE YOU?
Ctrl+d
$cat>two
I'M FINE.
Ctrl+d
$paste one two
HI, HOW ARE YOU? I'M FINE

6. **OBSERVATION:** Studied above all commands and run successfully.

EXPERIMENT NO:-18

1. **OBJECTIVE: -** Find and replace commands.

2. **HARDWARE & SYSTEM SOFTWARE REQUIRED:-**
   Intel P-III Processor and above
   64 MB RAM and above
   HDD 4.3 GB Hard Disc and above

3. **SOFTWARE REQUIRED: -** Linux Operating System (Red Hat)

4. **THEORY:-**
   grep: using this command we are able to find particularly in a specified file or files. This command supports all regular expressions. Grep stands for globally search for regular expressions.
   Syntax: $grep [option] [string] [file names]
   mv: using this command we are able to rename a specified file or directory.
   Syntax: $mv [option] [source file name] [target file name]

5. **PROGRAM INPUTS & OUTPUT:-**
   1. $grep java nit
   It is searching java string nit file.
   2. $mv nit1 nit2
   It is replacing nit1 file to nit2.

6. **OBSERVATION:** Studied above all commands and run successfully.