

AMIRAJ

COLLEGE OF ENGINEERING & TECHNOLOGY

CHAPTER -9 UNIX/LINUX OPERATING SYSTEM

```
[dope@dreckskind:~]$ ls -ld *(/*)
.dope/      .gnupg/      .ncftp/      .tags/      .xmms/      gnus-5.10.6/
.emacs.d/   .irssi/      .opera/      .terminfo/  .xvpics/    homepage/
.eshell/    .luna/      .qt/         .translate/ .zsh/       nslrn/
.fluxbox/   .media/     .screen/    .vim/       MuttMail/   ripples/
.fonts/     .filter/    .sigs/      .w3m/       Pics/        scripts/
.fvwm/      .slang/     .xchat/     Trash/      shots/
.getmail/   .src/       .src/       .src/
.gimp-1.2/  .src/       .src/       .src/
[dope@dreckskind:~]$ status
Date.: 2016-05-19 09:49
Shell.: bash (PID = 18167, 4 nests)
Term.: /dev/pts/0 (xterm), 38400 bauds, 80 x 24 cars
Login.: root (root) on dreckskind
System: GNU/Linux 3.0a
Uptime: 04:09:49 up 5 days, 49 min,  5 users,  load average: 0.30, 0.19, 0.12
[dope@dreckskind:~]$
```



Linux Shell Scripting
Text Processing Tools

```
override@Atul-HP: ~$ ls -l
total 212
drwxrwxr-x  5 override override 4096 May 19 03:45 acadenv
drwxrwxr-x  4 override override 4096 May 27 18:20 acadview_demo
drwxrwxr-x 12 override override 4096 May  3 15:14 anaconda3
drwxr-xr-x  6 override override 4096 May 31 16:49 Desktop
drwxr-xr-x  2 override override 4096 Oct 21  2016 Documents
drwxr-xr-x  7 override override 40960 Jun  1 13:09 Downloads
-rw-r--r--  1 override override 8980 Aug  8  2016 examples.desktop
-rw-rw-r--  1 override override 45005 May 28 01:40 hs_err_pid1971.log
-rw-rw-r--  1 override override 45147 Jun  1 03:24 hs_err_pid2006.log
drwxr-xr-x  2 override override 4096 Mar  2 18:22 Music
drwxrwxr-x 21 override override 4096 Dec 25 00:13 Mydata
drwxrwxr-x  2 override override 4096 Sep 20  2016 newbin
drwxrwxr-x  5 override override 4096 Dec 20 22:44 nltk_data
drwxr-xr-x  4 override override 4096 May 31 20:46 Pictures
drwxr-xr-x  2 override override 4096 Aug  8  2016 Public
drwxrwxr-x  2 override override 4096 May 31 19:49 scripts
drwxr-xr-x  2 override override 4096 Aug  8  2016 Templates
drwxrwxr-x  2 override override 4096 Feb 14 11:22 test
drwxr-xr-x  2 override override 4096 Mar 11 13:27 Videos
drwxrwxr-x  2 override override 4096 Sep  1  2016 xdm-helper
override@Atul-HP: ~$
```

Subject:- OS
Code:-3140702

Prepared by:
Asst.Prof.Foram Patel
(Computer Department,ACET)

AMIRAJ
COLLEGE OF ENGINEERING & TECHNOLOGY

SHELL SCRIPT

- A shell script is a computer program designed to be run by the Unix/Linux shell which could be one of the following:
- The Bourne Shell, The C Shell
- A shell script is a file containing a series of commands.
- SHELL SCRIPTING is writing a series of commands for the shell to execute.
- It can combine lengthy and repetitive sequences of commands into a single and simple script, which can be stored and executed anytime.
- To successfully write a shell script, you have to do three things:
- Write a script
- Give the shell permission to execute it
- Put it somewhere the shell can find it

BASIC LINUX COMMANDS

bc :

- bc command is used for command line calculator.
- The Syntax is
- bc [options]

cal :

- cal command is used to display the calendar.
- The Syntax is
- cal [options] [month] [year]

cd :

- cd command is used to change the directory.

BASIC LINUX COMMANDS

- **clear :**
- This command clears the terminal screen.
- **date :**
- date command prints the date and time.
- The Syntax is
- `date [options] [+format] [date]`
- **echo:**
- echo command prints the given input string to standard output.
- The Syntax is
- `echo [options..] [string]`

BASIC LINUX COMMANDS

•cp :

- cp command copy files from one location to another.
- If the destination is an existing file, then the file is overwritten;
- if the destination is an existing directory, the file is copied into the directory

•grep:

- grep command selects and prints the lines from a file which matches a given string or pattern.

•hostname :

- hostname specifies the name of the host

•head:

- head command is used to display the first ten lines of a file, and also specifies how many lines to display.

BASIC LINUX COMMANDS

- **last:**

- last command is used to display the last logged in users list.

- **ls:**

- ls command lists the files and directories under current working directory.

- **man:**

- man command which is short for manual, provides in depth information about the requested command (or) allows users to search for commands related to a particular keyword.

- **mkdir:**

- mkdir command is used to create one or more directories.

BASIC LINUX COMMANDS

- **pwd**

- Show current working directory.

- **touch file**

- Create or update file

- **rm file**

- Deleting the file

- **rm -r dir**

- Deleting the directory

- **rm -f file**

- Force to remove the file

BASIC LINUX COMMANDS

- **cp file1 file2**

- Copy the contents of file1 to file2

- **cp -r dir1 dir2**

- Copy dir1 to dir2; create dir2 if not present

- **mv file1 file2**

- Rename or move file1 to file2, if file2 is an existing directory

- **ps**

- To display the currently working processes

- **top**

- Display all running process

BASIC LINUX COMMANDS

- **kill pid**

- Kill the process with given pid

- **grep**

- pattern file Search for pattern in file

- **Uptime**

- Show current uptime

- **whoami**

- Who you are logged in as

- **finger user**

- Display information about user

BASIC LINUX COMMANDS

- **exit**

- Logout the current session

- **tty**

- Displays current terminal.

- **clear**

- This command clears the screen.

- **cat**

- The 'cat' command is actually a concatenator but can be used to view the contents of a file.

- **wc**

- This command counts lines, words and letters of the input given to it.

WHILE LOOP SYNTAX IN UNIX

- while [condition]
do
 command1
 command2 Command N
Done

- Example:
while [\$n -le 5]
do
 echo "Welcome \$n times."
done

FOR LOOP SYNTAX IN UNIX

- for
((*variable_initialization*; *loop_condition*; *variable_update*))
do
 statement 1
 statement 2
 statement 3
 ...
 statement n
done

Example:

```
for (( c=1; c<=5; c++ ))  
do  
echo "Welcome $c times"  
done
```

IF CONDITION SYNTAX IN *UNIX*

- if [expression 1]
- Then
- Statement(s) to be executed if expression 1 is true
- elif [expression 2]
- then
- Statement(s) to be executed if expression 2 is true
- elif [expression 3
- then
- Statement(s) to be executed if expression 3 is true
- else
- Statement(s) to be executed if no expression is true
- fi

CASE SYNTAX IN UNIX

- case word in
- pattern1)
- Statement(s) to be executed if pattern1 matches ;;
- pattern2)
- Statement(s) to be executed if pattern2 matches ;;
- pattern3)
- Statement(s) to be executed if pattern3 matches ;;
- *)
- Default condition to be executed ;;
- esac