

Linux Commands Lab Work:

Shell is an application program which provides CUI/CLI in Linux System through which user can interact with system by means of entering commands in a text format.

+ Command name: "pwd" - print/present working directory

- this command displays absolute path (full path) of the present working directory
- pwd command internally refers the value of shell variable by the name "PWD".

+ Command name: "mkdir" - create directory/directories

- create a new directory in a home directory named as -- "os/linux_commands"

```
$mkdir -p os/linux_commands
```

+ Command name: "cd" - change directory - for navigation throughout filesystem.

```
$cd dirpath      : change dir to dirpath
$cd ~            : change dir to the home dir
$cd /            : change dir to root (/) dir
$cd -            : change dir to previously accessed dir
$cd .            : change dir to current dir (.) single dot
$cd ..          : change dir to parent dir (..) two dots
```

+ Command name: "ls" - lists directory contents

```
$ls : by default displays contents of current dir columnwise in a
alphanumerically sorted manner.
```

```
$ls dirpath : displays contents of dirpath, whereas dirpath is any absolute path
or relative path.
```

```
$ls -l      : displays dir contents listwise
$ls -1      : display dir contents one entry per line
$ls -a      : display all contents of the dir (including hidden files)
$ls -A      : display almost all contents of the dir (excluding . & .. )
$ls -li -1   : display inode numbers of each file
$ls -s      : display no. of data blocks allocated for each file
$ls -ls      : display no. of data blocks allocated for each file in a dir
```

options/flags/arguments can be used with ls command or information about any other command can be checked with the help of man command.

+ Command name: "man"

- "man" command is a system manual pager i.e. online help of linux system which is used to display information about commands/library functions/system calls etc..

For example:

`$man ls` : to display information about "ls" command - by default it displays information of user commands from section-1 of manual pages.

`$man 1 mkdir` : displays information user commands from section-1 of manual pages.

`$man 2 _exit` : to display information about `_exit()` system call

`$man 3 printf` : to display information about `printf()` library functions

- to man command we can pass first arg as a **section number** of a manual pages

1: to display information about user/system commands binaries

2: to display information about system calls

3: to display information about library functions

- explore man command with the following command

`$man man`

+ Command name: "touch" - change the file timestamps

- `$touch india.txt` --> timestamps of a file "india.txt" gets changes if it exists, if it

file does not exist, then new file named as "india.txt" gets created having size 0.

+ Command name: "cp" - copy files and directories

+ Command name: "mv" - move file/files from one location to another location

`$mv <filepath> <dest_dir_path>`

- also can be used to rename the file if filepath exists in a dest_dir_path itself.

Exercise:

- open terminal (shell):

press - `ctrl+alt+t`

- enter `pwd` command to display current working dir/present working directory

`$pwd`

- create a new dir in a home directory

`$mkdir os`

- change directory to os

`$cd os`

- create a new directory in a os directory by the name `linux_commands`

`$mkdir linux_commands`

- change directory to `linux_commands`

`$cd linux_commands`

- create dir's by the name one, two and three in a current dir
`$mkdir one two three`

- check dir's got created or not, give ls command with following options
`$ls`
`$ls -l`
`$ls -s`
`$ls -a`

- goto the dir one
`$cd one`

- goto to the previously accessed directory
`$cd -`

- make sure you are in a linux_commands dir by using cd command, create a new dir by the name four, inside dir four create sub dir five, inside dir five create sub dir six with only one mkdir command.
`$mkdir -p four/five/six`

- check dir four and sub dir's inside it got created or not by using command
`$ls -R` : display contents of dir and its sub dir's recursively

- goto the dir one
`$cd one`

- create a new file by the name "file1.txt" and enter your name personal information into it:

```
$cat > file1.txt
sachin pawar
sunbeam karad
sunbeam pune
```

- press (cntrl+d) to stop writing the contents into the file (i.e. to enter end of file character into the file.

- to display contents of the file
`$cat file1.txt`

- to display contents of the file in a reverse order
`$tac file1.txt`- to append data into the file "file.txt"
`$cat >> file1.txt`
algorithms and data structures
operating system concepts and linux programming
(cntrl+d)

- create two more files "file2.txt" and "file3.txt" by using cat command in the same dir one

file2.txt contents are:

c programming language
java programming lanaguage
c++ programming language
advanced web programming

file3.txt contents are:

android programming
iphone programming
mobile computing technologies

- enter the following commands and observe the output:

\$cat file1.txt file2.txt file3.txt : cat command is used to concatenate contents of the file/s and display on the terminal.

+ Command name: "rmdir" - this command is used to remove empty directory only.

\$rmdir <dirpath> : to remove empty directory only

\$rm -r <dirpath> : to remove the contents of the dir recursively and remove dir

- delete all the directories and its sub directories contents:

- goto to the directory linux_commands from current dir by using relative path or

absolute path as:

\$cd /home/sunbeam/os/linux_commands

\$rm -r : delete all the contents of current dir recursively.

Exercise:

- create a dir by the name one in a cur dir and change dir to one

\$mkdir one

\$ cd one

- create 3 text files inside dir one

\$touch 1.txt 11.txt 111.txt- create dir by the name two inside one and go inside it

\$mkdir two

\$cd two

- create 3 text files inside dir two

\$touch 2.txt 22.txt 222.txt

- create dir by the name three inside two and go inside it

\$mkdir three

\$cd three

- create 3 text files inside dir three
\$touch 3.txt 33.txt 333.txt
- create dir by the name four inside three and go inside it
\$mkdir four
\$cd four
- create 3 text files inside dir four
\$touch 4.txt 44.txt 444.txt
- create dir by the name five inside four and go inside it
\$mkdir five
\$cd five
- create 3 text files inside dir five
\$touch 5.txt 55.txt 555.txt
- goto home directory:
\$cd ~ OR \$cd /home/sunbeam

After creating above directory structure do following operations:

1. list directory contents of the dir five from cur directory (i.e. from home dir)
 2. write your name in a file named 444.txt of dir four from current dir.
 3. remove the file named 555.txt from current directory.
 4. change directory to five, use relative path to go dir five
 5. write your course name in file named 333.txt which resides in dir three from cur directory (i.e. from five).
 6. list the contents of dir two from the cur directory (i.e. from five)
 7. remove file named 222.txt which belongs to dir two from cur dir.
 8. change directory to one
 9. remove all the files only from dir named five from current directory (i.e. from one).
 10. remove directory named five from the current directory (i.e. from one).
 11. remove whole directory four from current directory (i.e. from one).
 12. change directory to home directory
 13. change the time stamps of the file named 333.txt which resides in a directory by the name three from the current directory.
 14. add contents into the file 11.txt by using cat command, and display again from current directory (i.e. from home directory).
 15. append content inside the same file again from the same location.
 16. goto dir one, create a file by the name .india.txt and append data into it.
- After creating file give \$ls command & \$ls -a command and check the difference.

+ Command name: cal

\$cal

:

displays calender of current month

\$cal <year> :

displays calender of year mentioned

\$cal <month> <year> : displays calender of month of year

\$cal -3

:

display previous, current & next month calender

```
$cal -1
```

:

displays calender of current month only.

+ Command name: date

```
$date
```

:

displays current date

```
$date +"%d/%m/%Y"
```

```
$date +"%d/%m/%y"
```

SunBeam