Practical – 1 Linux General Utility Commands & Basic File command

- man command

  man command in Linux is used to display the user manual of any command that we can run on the terminal. It includes NAME, SYNOPSIS, DESCRIPTION, OPTIONS, EXIT STATUS, RETURN VALUES

  ```
  $ man [COMMAND NAME]
  ```

  e.g man echo // echo is the command name . command will provide a help of echo command

  press q key to exit from man manual

- Who command

  The who command displays the following information for each user currently logged in to the system

  Syntax : who [options]

  It displays
  Login name of the users
  Terminal line numbers
  Login time of the users in to system

  Option s

  -H : display headings
  -u: list users logged in

```
student@mcastaff:~$ who
student  :0         2021-03-20 12:55 (:0)

 student@mcastaff:~$ who -u
student  :0         2021-03-20 12:55   ?        1109 (:0)

student@mcastaff:~$ who -uH
NAME     LINE        TIME          IDLE        PID     COMMENT
student  :0         2021-03-20 12:55   ?        1109 (:0)
```

- ## Whoami command

  It displays the username of the current user.

  e.g student@mcastaff:~$ whoami
  student

- ## Pwd command

  print name of current/working directory(including path form /)

  e.g student@mcastaff:~$ pwd
  /home/student

- ## Cal command

  Display the calendar. By default, display current month calendar.

  Syntax : cal [option]

  Options :
  -3: Display the previous, current and next month surrounding today.
  -y: Display a calender for specified year
  -j: Display julian date
  -m month number: calendar for particular month

  Examples
  $ cal
  $ cal -m 6
  $ cal -y 2020
  $ cal -j
  $ cal -jm 4
  $ cal -3

- ## Date Command

  Print or set system date and time.

  Syntax : date [option] [+FORMAT]

  Different format options:
  %a locale's abbreviated weekday name (e.g., Sun)
  %A locale's full weekday name (e.g., Sunday)
  %b locale's abbreviated month name (e.g., Jan)
  %B locale's full month name (e.g., January)
  %H hour (00..23)
  %I hour (01..12)
  %H hour (00..23)

%I hour (01..12)
%m month (01..12)
%M minute (00..59)
%y last two digits of year (00..99)
%Y year

Example

$date

$ date +%a

Sun

$ date +%A%y

Sunday21

$ date +"%a %b %y %H"

Sun Mar 21 17

- mkdir command

create a directory

syntax : mkdir [option] directoryname

-p : make parent directories as needed, error if exist

Example

$mkdir test

$ ls // ls is for listing out all the files/directory inside current directory

Desktop Downloads Music Public **test**

$ mkdir test1 test2

$ ls

Desktop Downloads Music Public test test1 test2

$ mkdir test/sub1 //create sub1 directory under the test directory

$ mkdir new/sub1

mkdir: cannot create directory 'new/sub1': No such file or directory

$ mkdir -p new/sub2 //create new directory and inside that sub2

- cd command

change directory

Example

$cd /home/student/test/sub1 // absolute way to go inside sub1 directory. Path always start from root(/)

Relative way

Single dot (.) is current directory

Double dot(..) is one level-up to go to the parent directory from current directory

home$cd test

home/test $ cd ..

home$

- rmdir command

  remove the empty directory

  Syntax rmdir [options] directory names

  Option :

  -v : gives verbose to deleted directory

  -p : remove directory and its parent directory also

Example

$rmdir -v test1

rmdir: removing directory, 'test1'

$ rmdir test/sub1

$ mkdir -p test/sub1 test/sub2 test/sub1/sub11

$ rmdir -p test/sub2 test/sub1/sub11 // remove subdirectories first then remove parent directory

- cat command

  concatenate files and display the content on standard output. Create a new file.

  syntax : cat [option] files

  options

  -b: number nonempty output lines, overrides -n

  -E: display $ at end of each line

  -n: number all output lines

  -s: suppress repeated empty output lines

  Example

  create a file inside test2 directory

  test2$ cat > file1

  this is the first file

  this is the second line of the file

  emppty lines in between

test2$ cat > file2
this is second file.

entering data into second file.

abc
efg
testing of cat command.

$cat file1
$ cat file1
$ cat -b file1
$ cat -n file1
$ cat file1 file2 // concatenate the both file's content and print on standard output
$ cat -s file1

- cp command

cp command copies the file or group of file. It can also create the exact image of file with other name.

Syntax : cp [options] source destination

Options :

-i : - prompt whether to overwrite existing file
-R :- copy whole directory structure
-v : verbos

student@mcastaff:~/test2$ cp file2 second.txt // copy file2 as second.txt
student@mcastaff:~/test2$ cp file2 second.txt sub/ //copy file2 and second.txt inside subdirectory called "sub"

student@mcastaff:~/test2$ cat > two.txt
new file
two.txt
testing for the cp command
student@mcastaff:~/test2$ cp -v two.txt two
'two.txt' -> 'two'
student@mcastaff:~/test2$ cp -i two.txt new.txt
cp: overwrite new.txt? n
student@mcastaff:~/test2$ cp -R sub sub1 // copy all content of sub directory to sub1 inside test2 directory

- mv command

move or rename the file

Syntax : mv [options] source destination

-i : interactive prompt before overwrite

Example

student@mcastaff:~/test2$ mv two sub/ // move two file inside sub/

student@mcastaff:~/test2$ mv sub/two ../two

// move two file inside sub is to parent directory of "test2" (.. parent directory)

student@mcastaff:~/test2$ cd ..

student@mcastaff:~$ ls

Desktop Downloads Music new.txt Public test2 Videos

Documents examples.desktop new Pictures Templates two

Rename file

student@mcastaff:~/test2$ ls

file1 file2 new.txt one.txt second.txt sub sub1 two1

student@mcastaff:~/test2$ mv second.txt newsecond.txt

student@mcastaff:~/test2$ ls

file1 file2 newsecond.txt new.txt one.txt sub sub1 two1

- rm command

rm commad is used to delete a file from a directory.

Syntax : rm [option] file/files name

Options

- i : prompt for conformation

- r : recursively remove directory tree structure

student@mcastaff:~/test2$ rm one.txt new.txt //remove the one.txt and new.txt

student@mcastaff:~/test2$ ls

file1 file2 newsecond.txt sub sub1 two1

student@mcastaff:~/test2$ rm -i file1

rm: remove regular file 'file1'? n

student@mcastaff:~/test2$ rm -r sub // delete sub directory and all files inside sub directory

student@mcastaff:~/test2$ ls

file1 file2 newsecond.txt sub1 two1

Implement following commands and write their syntax, description, options (if any) and examples with output.

1. Cal
2. date
3. pwd
4. mkdir
5. rmdir
6. cd
7. cat
8. cp
9. mv

Exercise

10. create a directory as 2MAxx . (e.g 2MA01) xx is your roll no
11. create a directory OSLP inside 2MAXX, create subdirectory LE under OSLP .
12. create subdirectory newdir under LE .
13. create sub directory into newdir called sub1,sub2 and test
14. create a file one.txt in newdir.
15. create a file two.txt in newdir.
16. create a file f1 f2 in newdir.
17. display content of one.txt and two.txt with line number.
18. create a file one.txt into sub1.
19. create a file two.txt into sub2
20. go to sub2 directory and come back to LE directory
21. delete test and sub2 directory.
22. create mydir/new1 and mydir/new2 under LE directory.
23. copy all the file form new dir to sub2 and sub1
24. move all file from sub2 to mydir/new1 directory.
25. delete the file f1 from newdir.
26. Rename the file f2 with f2.txt in newdir
27. Display the date in format **2022 feb 14 18 10** . (note 18 hrs 10 minute)