UNIX COMMANDS

File Commands

1) cat
2) cp
3) mv
4) rm

cat command:

1) Creating the new file
2) Display content of the file
3) Concatenating more than one file
4) Appending data to the existing file
5) Copying many files in to one single file.

Ex:

1) Creating the new file:

```
$cat >paypal.txt
--------
--------
--------
ctrl+d
```

2) Dispay data in the file:

```
$cat < paypal.txt
or
$cat paypal.txt
```

3) Concatenation of files:

```
$cat paypal.txt google.txt
```

4) Copying data from many file to single file:

```
$cat paypal.txt google.txt >amazon.txt
```

ls : Used for listing /display all the files in the curreny directory.

cp command:

1) copy contents from a file to another file(source->destination)
2)copy files from one directory to another directory.

Example:

```
$ cp paypal.txt google.txt
```

mv command:

1) Used for renaming a file(changing name of the file)
2) Renaming a directory
3) Moves files from one directory to another directory

Example:

```
$mv paypal.txt funpal.txt --> renaming a file
$mv chennai hyd --> renaming a directory
$mv google.txt hyd -->here the file google moves to hyd
directory
```

** Note: If we use cp command duplicate files will be crated
**If we use mv command file will be moved to target location(no duplicates)

rm command: used for delete/removing a file/directory.

Example:

```
$ rm google.txt --> it deletes file directly without
confirmation
$ rm -i google.txt -->deletes file with confirmation
$rm -r hyd
```

** Here this commands removes hyd directory even thou its having sub directories & files.

* But rmdir command removes directory when it should be empty.
* -r is recursive

Unix directory commands

1. Mkdir
2. cd
3. rmdir

mkdir: Creating a directory which has set of files & sub directories.

Example:

```
$mkdir hyd
$mkdir chennai
```

cd: Used for changing/closing directory

Example:

```
$cd chennai
$chennai>
```

If we want to come back to my previous directory we can use like

```
$chennai>cd ..
$ cd                  --> come back to root directory
```

If we want to come back more than one directory then

```
$chennai>places>cd ../..      --> this command is back
to 2 directories
```

rmdir: Used for removes/deletes directory only if is empty.

Example:

```
$rmdir chennai
```

Unix Miscellaneous commands

- ls
- pwd
- ln
- head
- tail
- cal
- ps
- kill
- who
- whoami
- uptime
- ut

ls : This command displays files and directories in columnar format.

Example:

```
$ ls
$ ls -S
```

Arrange the files based on the size(S is upper letter)

```
$ ls -l
```

long listing the files

```
$ ls -a
```

Displays hidden files

```
$ ls -i
```

Displays inodes for each file

```
$ ls -R
```

Displays all directories along with subdirectories in current working directory.

creating a hidden file:

```
$ cat >.employee.txt
```

Wild card characters using with ls command.

?	Represents single character
*	Represents gruop of characters
[ ]	Represents searching pattern

Examples:

$ ls ?	Displays files with one letter
$ ls x*	Displays files which are starting with 'x'
$ ls ???	Displays files with 3 letters
$ ls *.out	Displays all the files with extension 'out'
$ ls [a-z]	Displays single character files which are from a to z
$ ls [a-z]*	Displays files starting with a to z
$ rm ?	Removes the files with single character
$ rm *.c	Removes the files with extension 'c'
$ cp ? chennai	Single digit files will be copied into directory 'chennai'

pwd : This command shows current working directory in unix

```
$ pwd
```

head : Used to display First lines of the file

Example:

```
$ head  paypal.txt
```

This command displays default 10 lines of the file paypal.txt

```
$ head -n15  paypal.txt
```

This command displays 15 lines of the file paypal.txt

```
$ head -n3  paypal.txt
```

This command displays 3 lines of the file paypal.txt

tail : Used to display last lines of the file.

Ex:

```
$ tail paypal.txt
```

This command displays default last 9 lines.

```
$ tail -n5 paypal.txt
```

This command displays last 4 lines of the file.

```
$ tail -n15 paypal.txt
```

This command displays last 14 lines of the file.

cal command: Used for display the calender.

Ex:

```
$ cal 2011                      Displays 2011 calender
$ cal  2   2011               Displays Feb month in 2011 year
```

ln :

1.This will create link between 2 files.(2nd file should new one)

2. If one file modified another one affected.

Ex:

```
$ln paypal.txt    funpal.txt
```

ps : Knowing backgroud process running /stopped

```
$ ps pid
```

kill : Used for terminating the process

Ex:

```
$ kill  pid
```

who : Used for to display who are working in the system

Ex:

```
$ who
```

whoami: Display my user name/accout

Ex:

```
$ whoami
```

uptime: This command used for display load in the server.

Unix Filter Commands

1) grep
2) sort
3) more
4) cut
5) wc
6) uniq

1) grep: (Global Regular Expression Pattern)

This command is used for searching a required patern in a file.

Syntax:
$ grep [- optopn] "search patern" Filename [redirection symbol newfilename]

Options:

-i Ignores case sensitiveness in searching patern
-n displays line numbers for those lines which gets matched and un matched with the patern
-c counts number of times a searching patern exists and does not exists
-v (verbose) Displays those lines that does not match with the patern

Example:

```
$ cat > paypal.txt
welcome to unix
paypal welcomes you
unix multi user os
WELCOME to the world of unix

$ grep "welcome" paypal.txt
$ grep -i "welcome" paypal.txt
$ grep -i -n "welcome" paypal.txt
$ grep -i -c "welcome" paypal.txt
$ grep -i -v "welcome" paypal.txt
$ grep -i -v -n "welcome" paypal.txt > funpal.txt
```

Sort : Used to arrange numbers/text in ascending/descending order.

* by default it arranges ascending order.

Syntax:

```
$ sort [-option] [redirection synbol ] filename
[redirection synbol] [new filename]
```

options:

-r   Arrage data in reverse or descending order
-n   Arrage data in ascending or descending order by considering whole number.
* if n is not used then numbers gets arranged in order based on 1st digit.

Ex1:

```
$ sort > paypal.txt
6
2
9
1
5
3

$ cat paypal.txt      Ascending order
$ sort -r paypal.txt > funpal.txt     → Descending order
$cat funpal.txt
```

Ex2:

```
$ sort >google.txt
176
2165
8
93
-----
------ [ctrl+d]
```

```
$cat google.txt      It considered first digit
$sort -n google.txt > yahoo.txt    It considered whole
number.
```

more: This filter command used to display information from multiple files based n page wise.

It gives an identification of end of file for first file and begining of next file.

Syn:

$more [-option] file1,file2,file3, etc...

options:

-p   clears the screen and displays next file in the list of files

*note:

Enterkey retrives next file based on %s
spacebar retrieves complete data from next file

```
Ex: $more - p paypal.txt   funpal.txt
```

cut: Used to cut the required text from a file.It can cut the data on the columns and feilds.

Syntax:

$ cut [-option] filename [redirection symbol new filename]

-c   To cut the data in columns

-f   To cut the data in feilds that is that data which is separated by tab.

Ex1:

```
$ cat>paypal.txt
```

```
Hyderabad
Secunderabad
Andhra [ctrl+d]

$ cut -c1 paypal.txt    [Enter]
H
S
A

$cut -c3 paypal.txt     [Enter]

d
c
d

$cut -c1 -3 paypal.txt [Enter]

Hyd
Sec
And
```

Ex2:

```
$cat > funpal.txt [Enter]
India    Delhi
Andhra    Hyderabad
Peers    Net
[ctrl+d]

$ cut   -f1   funpal.txt     [Enter]
India
Andhra
Peers

$ cut   -f2   funpal.txt    [Enter]

Delhi
Hyderabad
Net
```

wc: It will count number of lines,worlds,characters in a file.

Syntax:

$ wc [-option] filename

options:

-l count number of lines
-w count number of words
-c count number of characters

Ex:

```
$ wc -l paypal.txt
$ wc -w paypal.txt
$ wc -c paypal.txt
```

uniq: This filter is used to get the uniq or duplicate lines from a file.Data should be in order.

Syntax: $ unix [-option] filename

options:

-d Display duplicate lines
-u Display uniq lines
-c Counts number of times each word has occured in a file

Ex:

```
$ cat > city.txt
ameerpet
ameerpet
peers
bhel
hyderabad
ameerpet
```

```
peers
secunderabad   [ctrl+d]

$ sort city.txt> city1.txt

$cat city1.txt      Ascending order of data displaying

$ uniq -u city1.txt

$ uniq -d city1.txt

$ uniq -c city1.txt
```

Unix File compare commands

1. cmp
2. diff
3.comm

cmp: It comapares 2 files.If files are same it returns promp or else it returns the message where the difference encountered.

Syn: $ cmp file1 file2

```
Ex: $ cmp  paypal.txt   funpal.txt
```

diff: This command compares 2files like cmp.If any difference found in 2 files it displays those lines.

Ex:

```
$ diff  paypal.txt   funpal.txt
```

comm : Used to compare 2 sorted files.It provides out put in 3 columns.

* In first column displays uniq lines of first file.
* In second column displays uniq lines of second file
* In third column displays common lines in 2 files.

Ex:-

Step1 : Create two files with some data

```
$ cat> paypal.txt
risk
payments
ebay
uv
norkom
```

```
$ cat> funpal.txt
foodball
cricket
crems
ebay
payments
```

Step2: Sort above files and store the data into another 2 different files

```
$ sort  paypal.txt   paypal1.txt   [Enter]

$ sort  funpal.txt   funpal1.txt   [Enter]
```

Step3: Use comm command

```
$ comm paypal1.txt   funpal1.txt  [Enter]
```

** output shows in 3 different columns.

Unix File Access Permissions – Chmod

chmod: It provides permissions over a file in 3 categories.

1) owners
2) groups

3) others

Permissions which can be granted are read,write and execute

1) read (r)
2) write (w)
3) execute (e)

These permissions are represented with numeric values

```
r -  4
w -  2
e -  1
-----------
     7
-----------
```

Owners are users whose files gets referred from their respective accounts.
Groups are users whose accounts are dependent on the other accounts.
Others are users who can access the files of other users.

Chmod command is used to change the permissions for a file or directory.

Syntax:

```
$ chmod  FAP  Filename
```

* FAP is file access permissions

Examples:

```
$ chmod ooo  paypal.txt
```

No permissions to owners,groups and others

```
$ chmod 777  paypal.txt
```

All permissions to owners,groups and others

```
$ chmod 444  paypal.txt
```

Read permission to owners,groups and others (4 - read)

```
$ chmod 600  paypal.txt
```

Read(4),write(2) permissions to owners, no permissions to groups and others

```
$ chmod 664  paypal.txt
```

Read,write permissions to owners,groups and read permission to others

```
$ chmode 111 paypal.txt
```

Execute permission to owners,groups and others

Change permissions using name of the permission:

Examples:

```
$ chmod  u-w  g-w  o-r  paypal.txt
```

* write permission cancelled from owner
* write permission cancelled from groups
* read permission cancelled from others

```
$ chmod  u+rwx  g+rwx  o+r  paypal.txt
```

* read,write,execute permissions added to owners
* read,write,execute permissions added to groups
* read permission added to others

** Note: +   used for giving permissions.

- is used for removing permissions.

Interview FAQ’S on Unix Commands

1) How to find hidden files in current directory?

```
$ ls -lrta
```

2) How to find current running processes in unix server?

```
$ ps -ef
```

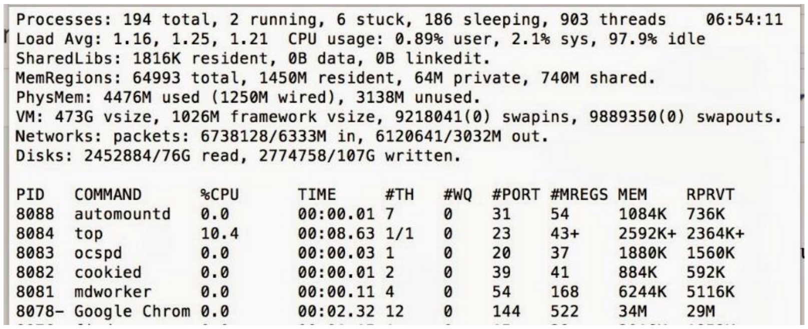
and if we want to find specific process we can use 'grep' with pipe

```
$ ps -ef | grep -i 'application'
```

3) How to find process which is taking maximum memory in server?

```
$ top
```

top command tell us about cpu usage , process id and other details. below is output of top command



```
Processes: 194 total, 2 running, 6 stuck, 186 sleeping, 903 threads    06:54:11
Load Avg: 1.16, 1.25, 1.21  CPU usage: 0.89% user, 2.1% sys, 97.9% idle
SharedLibs: 1816K resident, 0B data, 0B linkedit.
MemRegions: 64993 total, 1450M resident, 64M private, 740M shared.
PhysMem: 4476M used (1250M wired), 3138M unused.
VM: 473G vsize, 1026M framework vsize, 9218041(0) swapins, 9889350(0) swapouts.
Networks: packets: 6738128/6333M in, 6120641/3032M out.
Disks: 2452884/76G read, 2774758/107G written.

PID   COMMAND      %CPU      TIME     #TH    #WQ  #PORT #MREGS MEM     RPRVT
8088  automountd   0.0       00:00.01 7      0    31    54     1084K   736K
8084  top          10.4      00:08.63 1/1    0    23    43+    2592K+ 2364K+
8083  ocspd        0.0       00:00.03 1      0    20    37     1880K   1560K
8082  cookied      0.0       00:00.01 2      0    39    41     884K    592K
8081  mdworker     0.0       00:00.11 4      0    54    168    6244K   5116K
8078- Google Chrom 0.0       00:02.32 12     0    144   522    34M     29M
```

4) How to find Exception in log files available in current directory and how to find number of occurrence?

```
$ grep 'Exception' log1.txt | wc -l
```

5) find all files in current and subdirectories which contains 'log' name?

```
$ find . -name 'log'
```

6) How do you access command line arguments from within a shell script?

Arguments passed from the command line to a shell script can be accessed within the shell script by using a $ (dollar sign) immediately followed with the argument's numeric position on the command line.

7) How to tails last 200 lines of any log fine?

```
$ tail -200f filename.txt
```

8) How to find remaining disk space in unix\linux server?

```
$ df -kl
```

```
df -kl
Filesystem   1024-blocks      Used Available Capacity  iused    ifree %iused  Mounted on

/dev/disk0s2   244277768 153679844  90341924    63% 38483959 22585481   63%   /
```

9) How to make any script file executable?

```
$chmod 755 *.sh
```

10) How to kill process in unix server?

```
$ kill -9 #pid
```

these #pid can be found using ps -ef command.