1. Count multiple occurrences

**grep -o "your\_name" filename | wc -w**

2. Retrieve those dates where time is between 19:10:00 to 19:20:00

**awk '{if (($4 ~ "19:1[0-9]") || ($4 ~ "19:20")) {print}}' your\_file**

3. Top 10 CPU usage list

**ps -eo pcpu, pid, user, args |** sort -k 1 -r |head -10

4. Explanation: Find all files (excluding directories) from the

current directory which have been accessed no more(-) than 10 minutes ago.

**find -type f -amin -10**

**5. Pipe** is used to connect between two command

**aargs** redirect the output of second command as input to the first one

**find . -type f -mtime -1 -print | xargs ls**

**6.** To access B from A server

**rsh 10.0.0.10 -l root -n "ps -eaf | grep -i httpd"**

-D is used for enabling Debug socket and see local user

-n is used to redirect input from special device

-N Generate Separate Standard Output

-l username: specifies the remote user

7. Current Executing Process :

**ps -elf or ps -ef**

8. Last 15 min users

**last** or **who -s |grep '2017-06-06'** or **who -hu** or **who -u**

9. File type search

**find /path/to -regex ".\*\.\(jpg\|gif\|png\|jpeg\)" > log**

**find /path/to/ -iname '\*.gif' -o -iname '\*.jpg' filename.java -- for only this file**

10. CSV file read

cat xyz.csv

fo open("xyz", "r")

print xyz.[1]

cat test | while read a b c d; do echo "$a:$b:$c:$d"; done

Only first row : **awk -F "," 'NR==1 {print $1}' \*.csv**

**awk -F "," '{print $1}' \*.csv**

11. Mobile no verification

echo "enter a phone number"

while["$phone\_no" ! = [0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]]

do echo

12. Print 10 line default : tail /path/to/file

tail -n <number\_of\_lines> /path/to/file <file1> <file2> <file3>

**head -n 2 /etc/passwd**

**head -2 myownfile | tail -1**

head -c5 example.txt --- 5 bit chr

13. The ps -aef command will list all the running processes. **alias pg='ps -aef | ls -l'**

14. **awk** 'BEGIN {start\_action} {action} END {stop\_action}' filename

awk 'BEGIN {sum=0} {sum=sum+$5} END {print sum}' input\_file

awk 'BEGIN { for(i=1;i<=5;i++) print "square of", i, "is",i\*i; }' --- square of 1 is 1

awk 'BEGIN {FS=":"} {print $2}' input\_file OR awk -F: '{print $2}' input\_file --- center 0

awk 'BEGIN {OFS=":"} {print $4,$5}' input\_file -- center:0

awk '{print NF}' input\_file --display the number of columns in each row.

awk '{print NR}' input\_file --display the line numbers from 1

15. String function can be used with awk

split(string, array, delimiter)

index(string,search)

length(string)

split(string,array,separator)

substr(string,position)

substr(string,position,max)

tolower(string)

toupper(string)

16. basename : to remove suffix and prefix

basename /usr/bin/perlscript --remove the prefix, /usr/bin/, and print string 'perlscript'

basename perlscript script -- print perl

basename /usr/bin/perlscript script -- print perl

17. Grep Grep command

whole word check : grep -w "world" file.txt

regular exp check: grep "^[0-9].\*" file.txt

insensitive search: grep -i "UNix" file.txt

simple: grep "string" file1 file2

display 2 line before match : grep -B 2 "Error" file.txt

display 2 line after match : grep -A 2 "Error" file.txt

Displaying the lines around the match: grep -C 5 "Error" file.txt

string in all files recursively : grep -r "string" \*

invert the match: grep -v "string" file.txt

start with a string: grep "^start" file.txt

Matching the lines that end with a string : grep "end$" file.txt

position of chr: grep -o -b "string" file.txt

display line no : grep -n "string" file.txt

file display not match : grep -L "string" \*

file display that match: grep -l "string" \*

count of match : grep -c "sting" file.txt

remove emty line : grep -v "^$" file.txt

18.

find -type d -name ".\*" -- c. Finding hidden directories

find . -type f -name ".\*" -- b Finding hidden regular file

find . -type d -- a. Finding directories

find . -size 10M : exactly size

find . -size +10M : based on more size

find . -size -10M : based on less size

find . -perm 777 : based on permission

find . -cmin -120 : change in 2 hr before

find . -atime -1 : Access in 1 days

find -name "\*java\*" | xargs ls -l : find permission

19 DATE

format : date '+%m-%d-%Y' -- 01-23-2012

set : date -s "01/01/2000 12:12:12"

date -u --- Mon Jan 23 09:40:21 UTC 2012

date '+%H-%M-%S' -- only time part 01-48-45

date '+%m-%d-%Y %H-%M-%S' --- 01-23-2012 01-49-59

date '+%s' -- unix epoch 1327312228

20. SED : Replacing or substituting string

sed 's/unix/linux/' file.txt --- unix to linux

sed 's/unix/linux/2' file.txt -- 2 time replace

sed 's/unix/linux/g' file.txt \_\_ gloabl replace

>sed -n 's/unix/linux/p' file.txt -- replaced print file

>sed '2 d' file.txt -- line delete

>sed '5,$ d' file.txt -- range delete

21. cp sum.pl tmp/

cp log.dat bad.dat tmp/

cp /usr/local/bin/multiply.sh . -- copy a file from a different directory to the current directory.

cp docs/\* tmp/ :: all file

cp docs/\* scripts/\* tmp/

cp -r docs tmp/ -- recursively copy

cp -f force\_file.txt /var/tmp/ -- force the cp command to copy

22. .hidden\_file : Current dir : ls -a

Other than current and parent : ls - A

Permission : ls -l

Column wise : ls -x

23 cut

cut -c4,6 file.txt -- 4th and 6th postion of all lines

cut -c4 file.txt - deffault 4th chr of all lines

cut -c4-7 file.txt : 4 se 7 chr

cut -c-6 file.txt : 1 se 6

cut -c10- file.txt : 10th to end

cut -c- file.txt : strt to end

24. rm and rmdir [options] directories

rmdir -p docs/entertainment/movies/

rm doc\*

rmdir -ifr docs/entertainment/movies/ -- f (no user) i (incensitive) r (recur)

25. mv [options] oldname newname

mv -i log.dat bad.dat : promote to ovverwitre

26. join [options] file1 file2

-o list : displays only the specified fields from both the files

27. split -l2 -d textfile -- break at 2nd and 3rd -- xaa and xab files

28. crontab -e or crontab -l : \* \* \* \* \* /usr/local/bin/list\_unix\_versions.sh

MI HH DOM MON DOW command

MI : Minutes from 0 to 59

HH : Hours from 0 to 23

DOM : Day of month from 0 to 31

MON : Months from 1 to 12

DOW : Day of week from 0 to 7 (0 or 7 represents Sunday)

crontab -u username -l

crontab -r -- remove

crontab -e -- edit

crontab cron\_backup.dat -- RESTORE

29 . SCP

scp filename user@remotehost:/remote/directory/

scp Unix-storage.dat /var/tmp/

scp -r directory user@remotehost:/var/tmp/

scp source\_user@source\_host:/usr/bin/mysql\_backup.sh target\_user@target\_host:/var/tmp/

30. wc [options] filenames

You can use the following options with the wc command.

-l : Prints the number of lines in a file.

-w : prints the number of words in a file.

-c : Displays the count of bytes in a file.

-m : prints the count of characters from a file.

-L : prints only the length of the longest line in a file.

31. ssh -l username remote-server

remote-server:[~]>exit

localhost:[~]> ssh username@remote-server

ssh -V

ssh -v user@remote-host

32. kill [-s signal] pid

kill -l --- ALL

ps -aef

Number Signal Name Description

0 SIGNULL Used to check access to the process id

1 SIGHUP Hup signal. Terminates the process.

2 SIGNINT Interrupt signal. Terminating the process

3 SIGQUIT Quit signal. Terminate process with core dump

9 SIGKILL Forcibly killing a process

24 SIGSTOP Pausing the process

26 SIGCONT Runs a stopped process

33. VS (concurrent Version Control system) is a version controlling system

cvs [option] filename

cvs add mysql.bat

cvs commit msql.bat

34. FTP file transfer prog

ftp remote-server-name :: - u -p

>open remote-server-name

ftp> get windows-cleveland.bat

ftp>mget \*.png -- remote to local

ftp>put linux-virtual-server.rpm --local to remote

ftp>delete linux-dedicated-server.dat

ftp>quit

ftp>help

35 TAR : .tar [options] [Archive file] [files list]

The options of tar command are:

c : creates a tar file.

v : verbose. Displays the files information.

f : Specify the tar file name.

r : **updates** the tar file with new files.

x : **Extracts** files from the archive (tar file).

t : view contents of tar file.

z : Specify the tar command to create a tar file using gzip in unix.

j : uses bzip2 to create the tar file.

36. The syntax of **sort** command is: sort [options] filename

The options are:

-b : Ignores leading spaces in each line

-d : Uses dictionary sort order. Considers only spaces and alphanumeric characters in sorting

-f : Uses case insensitive sorting.

-M : Sorts based on months. Considers only first 3 letters as month. Eg: JAN, FEB

-n : Uses numeric sorting

-R : Sorts the input file **randomly**.

-r : Reverse order sorting

-k : Sorts file based on the data in the specified field positions.

-u : Suppresses duplicate lines

-t : input field separator

37. **zip** [options] zipfile files\_list

-d : Removes the file from the zip archive

-u : Updates the file in the zip archive

-m : Deletes the original files after zipping.

-r : Recursively zips the files in a directory

-x : Exclude the files in creating the zip

-v : verbose mode

-1 : Compresses the files faster

-9 : Compresses the files better

-f : freshen only changed files.

zipfile : creates the zip file with name as zipfile.zip

files\_list : list of files to be zipped.

38. echo "Mail body" | mail -s "Mail subject" to@example.com

echo "Mail body" | mail -s "Mail subject" to@example.com

cat body.txt | mail -s "Mail subject" to@example.com

mail -s "Mail subject" "user1@example.com,user2@example.com" < body.txt

mail -f /var/spool/mail/user

39 . uniq [option] filename

The options of uniq command are:

c : Count of occurrence of each line.

d : Prints only duplicate lines.

D : Print all duplicate lines

f : Avoid comparing first N fields.

i : Ignore case when comparing.

s : Avoid comparing first N characters.

u : Prints only unique lines.

w : Compare no more than N characters in lines

40. Translate :echo "linux dedicated server" | tr "[a-z]" "[A-Z]"

41. **Permissions : chmod**

$ ls -l sample.sh

-rwx-rw-r-- 1 matt deploy 94 Oct 4 03:12 sample.sh

chmod [options] mode filename

-R : recursively change the permissions of a directory.

-v : Verbose

Read, write and execute: **ugo** : 4+2+1=7 : **rwx**

chmod 777 sample.sh or chmod ugo+rwx sample.sh

+ : Additional permissions. Selected permissions are added.

- : Revoke the permissions. Selected permissions are revoked.

= : Specific permissions. Only selected permissions are assigned.

$ chmod g-w sample.sh $ chmod o+x sample.sh $ chmod u=w sample.sh

$ chmod 421 sample.sh - Give read permission to user, write permission to group and execute permission to others.

42 **expr [expression]**

$ expr 5 + 3

$ expr 1 \< 2

echo "2+5" | bc

echo "3\*8" | bc

echo "2/3" | bc

echo "10^2" | bc

> echo "6%4" | bc

43.**paste [options] files-list**

The options of paste command are:

-d : Specify of a list of delimiters.

-s : Paste one file at a time instead of in parallel.

--version : version information

--help : Help about the paste command.

paste file1 file2

44. **top [options]**

The options are:

-b : Starts top command in batch mode. Useful for sending top output to other programs or file.

-d : specify the delay time between the screen updates.

-n : Number of iterations, the top should produce before ending.

-u : Monitor only the specified user processes.

-p : Monitor only the specified processes. Specify the process ID

45. **hostname**

IP: hostname -i

Domain: hostname -d

Socket: hostname -s