

OS Assignment 1

Name : Ubaid Akhtar Ghante

Reg no. : 2019BC5043

Roll no. : A22

Various Commands Used On Unix

- * Process related commands

→ fg - to continue a program which was stopped & bring it to foreground.

Ex :- 1) Launch any process

2) Stop it with 'ctrl + z'

3) Now bring it again using 'fg process-name'.

→ top - shows all running processes

Ex :- Just enter 'top' and press enter & 'q' to move out of process display

→ ps - it is similar to 'top' and but it is user specific or process specific.

Ex:

'ps	root'	OR	'ps	1268'
				↑ PID

→ kill - to terminate running processes.

Syntax : KIM PID

→ `pidof` - to find the PID of any process you want.

Ex: pidof Photoshop.exe

→ nice / renice - used to assign priorities to the process running or about to start.

Ex: 'niche -n 19 banshee'

This will keep bangshee at lowest priority

as in Linux it ranges from -20 to 19. Lower the niceness index, higher the priority.

'renice -20 -p 3293'

Here priority is being changed of PID 3293 to the highest.

Default priority index is 0.

→ df - shows free disk space on all file systems.
use '-h' for more readable format
Ex: 'df -h'

→ free - shows free & used ram on your system.
use '-m' & '-g' to display in MB & GB.
Ex: 'free -m'

* Scheduling related commands

→ crontab - used to run some scheduled task at specific time or date repeatedly.

Syntax: 'crontab -e'

Ex: Open cronjobs using above command & write the job you want to schedule in the format:

" * * * * * command/script "

Day of week (0-7)
Month (1-12)
Day (1-31)
Hr (0-23)
Min. (0-59)

Note: Use ';' for multiple tasks after every task.

→ **at** - used for scheduling task which are run only once.

Ex: 'at 5:00PM' ↵

now 'at' prompts you for the command to be run.

'at > echo hello >> thisfile'

this will write hello in thisfile at 5 pm.

Now press 'ctrl + D' to complete the task list.

Note:- You can view list of 'at' jobs given using 'atq' command.

* Memory subsystem related commands

→ **ipcs** - used to look at shared memory on a linux system

Ex:- 'ipcs -a'

here, '-a' gives some additional info.

→ **vmstat** - this gives a lot of info which contains swap & free memory numbers.

Ex:- 'vmstat'

→ **dmidecode** - used to find out hardware info about the installed RAM.

Ex:- sudo dmidecode -t 17

* File system related commands

→ **touch** - create a new file or update its timestamp.

Ex:- Create empty file called 'file1' & 'file2'

\$ touch file1 file2

→ **cat** - concatenate files & print to stdout.

Ex:- Create file with hello written in it.


```
$ cat > file1  
Hello  
^D
```

→ cp - simply copies the files.

Ex: copy contents from file1 to file2
\$ cp file1 file2

→ mv - moves files or rename files.

Ex: \$ mv file1 file2

→ rm - removes files & directories

Ex: \$ rm file1

→ mkdir / rmdir - make & remove directory respectively.

Ex: \$ mkdir dir1

\$ rmdir dir1

→ cd - change the directory you are currently in.

Ex: \$ cd dir1

→ pwd - prints out the path of present working directory.

Ex: \$ pwd

→ ls - lists all the files in present working directory.

Ex: \$ ls

* I/O related commands

→ > - the output intended for stdout can be directed to a file or follow up command using this notation. (Output redirection)

Ex:- \$ echo hello > file1

This will clear file1 & write hello or create file1 & write hello.

→ >> - Similar to '>' this notation appends the output rather than rewriting it.

Ex:- \$ echo hello again >> file1

→ < - used to redirect input of a command.

Ex:- \$ wc -L < users

→ << - used to read input until it finds a line containing the specified delimiter.

Syntax: command << delimiter
document
delimiter.

Ex:- \$ wc -L << EOF

hello

hello again

EOF

Output :- 2

→ | - takes output from one program, or process, and sends it to another.

Note :- STDIN → 0

STDERR → 2

STDOUT → 1

* Hardware related commands

→ `lscpu` - reports info. about cpu & processing unit.
Ex: `$ lscpu`

→ `lshw` - extract info. from different /proc files & shows multiple hardware units.
Ex: `$ sudo lshw -short`

→ `lsusb` - list all usb buses & devices connected to the USB controllers

→ `fdisk` - modify partitions on hard drives & list out partition info. using this command
Ex: `$ sudo fdisk -l`

→ `mount` - used to mount/unmount & view mounted file system

Ex: `$ mount | column -t | grep ext`

This will return a column of mounted fs where there is ext.