

Experiment No. 4

Aim: Execute UNIX commands for system administrative tasks such as process management and memory management.

- a) Execution of Process Management Commands like ps, pstree, nice, kill, pkill, killall, xkill, fg, bg, pgrep, renice, etc.
- b) Execution of Memory Management Commands like free, /proc/meminfo, top, htop, df, du, vmstat, demidecode, sar, pagesize, etc

Objective: To understand process management and memory management commands in UNIX.

Outcome: LO4: Execute Unix commands for system administrative tasks such as process management and memory management.

What is a Process?

An instance of a program is called a Process. In simple terms, any command that you give to your Linux machine starts a new process.

Having multiple processes for the same program is possible.

Types of Processes:

- Foreground Processes: They run on the screen and need input from the user. For example Office Programs
- Background Processes: They run in the background and usually do not need user input. For example Antivirus.
- fg jobname

Top

This utility tells the user about all the running processes on the Linux machine.

Field	Description	Example 1	Example 2
PID	The process ID of each task	1525	961
User	The username of task owner	Home	Root
PR	Priority Can be 20(highest) or -20(lowest)	20	20
NI	The nice value of a task	0	0
VIRT	Virtual memory used (kb)	1775	75972
RES	Physical memory used (kb)	100	51
SHR	Shared memory used (kb)	28	7952

PS

This command stands for 'Process Status'.

This command is similar to 'top' command but the information displayed is different.

To check all the processes running under a user, use the command

```
Ps ux
```

You can also check the process status of a single process, use the syntax –

```
ps PID
```

Kill

This command **terminates running processes** on a Linux machine.

To use these utilities you need to know the PID (process id) of the process you want to kill

Syntax -

```
kill PID
```

To find the PID of a process simply type

```
pidof Process name
```

NICE

Linux can run a lot of processes at a time, which can slow down the speed of some high priority processes and result in poor performance.

To avoid this, you can tell your machine to prioritize processes as per your requirements.

This priority is called Niceness in Linux, and it has a value between -20 to 19. The lower the Niceness index, the higher would be a priority given to that task.

The default value of all the processes is 0.

To start a process with a niceness value other than the default value use the following syntax

```
nice -n 'Nice value' process name
```

If there is some process already running on the system, then you can 'Renice' its value using syntax.

```
renice 'nice value' -p 'PID'
```

B) Memory Management Commands

DF

This utility reports the free disk space(Hard Disk) on all the file systems. If you want the above information in a readable format, then use the command

```
'df -h'
```

Free

This command shows the free and used memory (RAM) on the Linux system.

htop

Every Linux administrator and users have must come across the situation where the system has reached its saturation point and it starts consuming all of the resources allocated to it. At this particular time, we wholeheartedly want to identify the processes responsible for consuming such high resource usage. To identify such processes, we have many tools and commands available nowadays, but the best among all is **htop**.

htop allows you to monitor processes running on the system along with their full command lines. We can perform tasks related to processes (killing, renicing) without entering their PIDs. Htop allows us to sort the processes on the basis of CPU, Memory and Time Interval. It also allows us to sort all the processes owned by a particular user.

vmstat

vmstat command in Linux/Unix is a **performance monitoring command** of the system as it gives the information about processes, memory, paging, block IO, disk and CPU scheduling. All these functionalities makes the command *vmstat* also known as **virtual memory statistic reporter**

vmstat [options][delay [count]]

```
anisha@anisha:~$ history
 1  ps
 2  man ps
 3  ps -ef
 4  ps -ux
 5  top
 6  ps -al
 7  jobs
 8  fg
 9  bg
10  h:~$ top
11  kill 3786
12  nice
13  df
14  df -h
15  man df
16  free
17  gedit fork1.c
18  gcc fork1.c
19  ./a.out
20  gedit fork1.c
21  gcc fork1.c
22  ./a.out
23  gedit fork1.c
24  gcc fork1.c
25  ./a.out
26  ps
27  ps -ef
28  vmsat
29  vmstat
30  vmstat -f
31  vmstat -m
32  vmstat -s
33  vmstat -d
34  vmstat -t 3 5
35  history
```

Conclusion:- Memory management and process management commands executed successfully.