Types of process

- What is Process?
- Foreground processes (also referred to as interactive processes)
- Background processes (also referred to as non-interactive processes)
- Parent and Child process
- Daemons
- Init Process
- Zombie and Orphan process

ps command

- It displays information about the active processes running on the system.
- It also provides a snapshot of the current state of the system and its processes, including the process ID (PID), process state, parent process ID (PPID), CPU usage, memory usage, and other details.
- > SYNTAX:
- \$ ps [options]

Output of ps command

- PID unique process ID
- **TTY –** terminal type
- **TIME –** amount of CPU time
- **CMD** name of the command associated with process

Output of \$ps -e or A

Shows information about all processes running on the system

Output of \$ps -f

Displays a full-format listing of process

Fields described by ps –f:

- UID: User ID that this process belongs
- PID: Process ID
- PPID: Parent process ID
- C: CPU utilization of process
- STIME: Process start time
- TTY: Terminal type associated with the process
- TIME: CPU time taken by the process
- CMD: The command associated with process

Output of \$ps -u

Shows information about processes associated with a specific user

Output of \$ps –C

Display the process associated with command name

Output of \$ps -o

Display the output of the ps command by specifying the columns names.



- It displays all the currently running processes in a hierarchical tree structure, starting from the init process (usually with a process ID of 1) as the root of the tree.
- SYNTAX:
- \$ pstree [options] [pid or username]



To display process tree with process id



To see who is the owner/user of a process

To display process tree specific to a user

top command

- It displays information about the processes running on a system in real-time. It provides a dynamic, real-time view of the system's performance, showing the resource usage statistics for all active processes.
- SYNTAX:
- \$ top [options]

top command

- **PID:** unique process id.
- **PR:** process's priority means lower number higher the priority.
- VIRT: Virtual memory used by the task.
- USER: User name of owner of task.
- %CPU: CPU usage.
- TIME+: CPU Time
- SHR: Shared Memory size (kb)
- NI: Nice Value of task. A Negative nice value means higher priority, and positive Nice value means lower priority.
- %MEM: Memory usage of task.
- RES: Physical RAM(kilobytes)
- COMMAND: command associated with process

Shortcut keys while using top command

- c- for command absolute path
- n- for to change the number of task displayed
- k- to kill process by PID
- d or s– to change the refresh interval
- r- renice the value of process
- u- to display the process specific to user
- f- customized the field as required
- H- show threads

htop command

It also displays information about the processes running on a system in realtime. It provides a dynamic, real-time view of the system's performance, showing the resource usage statistics for all active processes. It is newer program compared to top command. It offer more functionality than top command.

SYNTAX:

\$ htop [options]

atop command

- It also displays information about the processes running on a system in realtime. It provides more detailed information about processes than top and htop command but its complex to use than other.
- It is more powerful tool for monitoring system performance.
- SYNTAX:
- \$ atop [options]

nice and renice command

- It allows you to run a command with a modified scheduling priority.
- ► The scheduling priority is a value that determines the order in which processes are scheduled to run on the system's CPU.
- The nice command takes a numeric argument, which ranges from -20 (highest priority) to 19 (lowest priority). By default, the nice value is set to 0
- SYNTAX:
- \$ nice -n <nice_value> <command>
- \$ renice <nice_value> -p <process_id>

How To Calculate PR or PRI Values

- Total number of priorities = 140
- Real time priority range(PR or PRI): 0 to 99
- User space priority range: 100 to 139
- Nice value range (NI): -20 to 19

- PR = 20 + NI
- PR = 20 + (-20 to + 19)
- PR = 20 + -20 to 20 + 19
- PR = 0 to 39 which is same as 100 to 139

fg, bg and jobs commands

- > You can use the **jobs** command to list the jobs that are currently running or suspended in the background.
- > You can use the **fg** command to bring a background job to the foreground.
- You can use the 'Control+Z keys and bg command to return a job to the background. The Control+Z keys suspend the job, and place it in the background as a stopped job. The bg command runs the job in the background.
- **SYNTAX:**
- \$ bg
- \$ fg
- \$ jobs

kill command

▶ It is used to terminate or stop a running process.

SYNTAX:

\$ kill [signal or option] PID(s)

You can kill the process in two ways

• Process name

• Process ID

kill command

The most common kill signals are:

Single Value
18
2
9
15
19
20

Effect

Continue the process Interrupt from keyboard Kill signal Termination signal Stop the process Stop the process