UNIT--7

# COMMUNICATION IN LINUX

Scanned by CamScanner

## Background Processes :-

A Background Process in a computer science that runs "behind the scenes" (i.e. in the background) & without user intervention. Typical tasks for these processes includes logging, system monitoring, scheduling & user notification.

On a Unix system, a background process or job can be further identified as one whose group ID differs from it's terminal group ID. This type of process is unable to receive keyboard signals from & typically will not send output to it's parent terminal.

Although Background processes are typically used for purposes requiring few resources, any processes can be run in the background, & even though the process is running in the background. To run a process in the background we add ampersand (&) symbol at the end of process.

It does not display the result.

Scanned by CamScanner

Example:- sort abc > xyz & ↵
(It display an id. -> 1213)

It display the no. which is the process id of that process using this we can see the status of a process or kill a process.

## * Advantages of Background Process :-

1) We can use console to execute one command.
2) If a process is running in Background then the same process will continue to run on the Unix server without any communication with the front console. The console can be re-used.
3) The most common reason to put a process in the background is to allow you to do something else interactively without waiting for the process to complete. At the end of the command you add the special background symbol (&). This symbol tells your shell to execute the given command in the background.

Scanned by CamScanner

## Disadvantages of Background Process :-

1) It does not report the success of process to the user.
2) It must require a re-direction of output.
3) All Background Process are terminated when we log-out to the system.
4) We cannot use more process in the Background because it digrade the system performance.

## Foreground Process :-

Foreground Process has access to the terminal standard I/O's. Background process typically run with little or no user interaction at all, they interact with the system.

In a foreground job, all of your input is directed to the process stdin (including the SIGINT) that is generated when we press Ctrl - C.

Scanned by CamScanner

## Process Related Command :-

1) ps :- (process state)

It display the all process which are running current-
-ly on a terminal.

Syntax :- ps ↵

pid time command.

- Some option :-

i) ps - a :-

It show all the process running on a terminal which are connected to the system. It is run by system administrator.

Syntax :- ps - a (all) ↵

pid terminals Time command.

ii) ps - u (for specific user)

It display the process run by any specific user.

Example :- ps - u Jyoti (user name) ↵

iii) ps - e (terminal) :-

It display the all process run by user & it also display the system process.

Scanned by CamScanner

iv) ps -t (terminal) ans)

It display the process run at any specific terminal.

Example:- ps -t 3a ↵
↳ Terminal name.

## Changing Process Priority ans)

The process having the higher-priority executed earlier than the process having lower-priority in the process queue. The priority of process decided by a no. which is associated with each process. The no. is called Nice value. It can be range from 0-39. The default nice value of the process is 20 (Twenty). The higher the nice value of a process lower is the priority. Thus a process with a nice value 25 executes after the nice value of 20 of a process.

ps -l ans)

It display the nice value of the process which is executed by user.

Scanned by CamScanner

ps- L ↵

1 - sort

2 - grep

A system administrator can change the priority (higher priority) of a process. But a general user change the priority (lower priority). We can change the priority of a process using Nice command.

$ nice -25 abc ↵ (low priority).

# nice -5 abc ↵ (high-priority).

* kill :-

We can terminate a process during it's execution using kill command. When we execute kill command then process is terminated from process queue.

Example:-

① kill ↵ (All running process are Terminated).

② kill 1214 ↵ (Terminate the process whose id is 1214)

Scanned by CamScanner

**nohup** :-

If we execute that process we have executed should not terminate when we log-out. Then we use the process pre-fixed with nohup command. Using this command we can send a process running in the background, log-out & come next time/day to see the output.

**Example:-** nohup sort abc > pqr ↵

**at** :-

Using this command we can execute a process according to the specific time given by the user.

**Example:-** $ at 9:30 am ↵

at > Date

at > clear

at > echo "Good Morng"

ctrl + D ↵

now + 50 sec.

now + 1 min.



Scanned by CamScanner

now + 2 hours.

" + 2 days.

" + 3 months

" + 2 years ↵

* **batch** :-

Using this command Linux executes our command when the processor are in idle state or min. workload.

Example:- $ batch ↵

> sort abc > pqr.

* **man** :-

Using this command we can display the details about a command.



Scanned by CamScanner