[Linux Free Bootcamp](https://youtu.be/mEQT-j-6j8A?feature=shared)

Process Management in Linux

* What is a Process?

In Linux, a process is just a program that’s running on your computer.

Process States:

1.Running(R): The process is actively using the CPU.

2.Sleeping(S): The process is waiting for something, like input from the user or a response from the disk.

3.Stopped(T): The process is paused. You can pause a process using Ctrl+Z.

4.Zombie(Z): The process has finished, but it’s still in the system because its parent process hasn’t checked its results yet.

Creating Processes:

* In Linux, processes are created using a system call called fork(). When a process calls fork(), it creates a copy of itself called the child process.
* If the child process needs to run a different program, it uses the exec() command to replace its program with the new one.

Commands in Linux to manage processes:

ps: Shows a list of processes running on the system.

ps aux: shows all the processes.

top: Shows live information about processes, like how much CPU and memory they are using.

kill: Stops a process. For example, kill 1234 stops the process with PID 1234.

nice: Changes a process's priority. If a process has a lower priority, it gets less CPU time. For example, nice -n 10 my\_program runs a program with lower priority.

fg and bg: These are used to manage processes in the foreground or background. If you start a long-running task and want to free up the terminal, you can use bg to run it in the background.

Signals:

A signal is a message sent to a process, asking it to do something. Some common signals are:

SIGINT: Interrupts a process (usually Ctrl+C in the terminal).

SIGTERM: Asks a process to stop politely.

SIGKILL: Forcefully stops a process immediately (this can't be ignored).

Scheduling and Priorities:

* Linux uses a scheduler to decide which process gets to use the CPU and for how long. Processes are given priority based on their nice value:
* The nice value ranges from -20 (high priority) to 19 (low priority).
* A process with a low nice value gets more CPU time than one with a high nice value.

Note:

* When a process is done, the system clears it from memory and reclaims the resources it was using (like CPU time and memory).
* If a process finishes normally, the system removes it.
* If a process is killed, the system also removes it.
* If a child process finishes, the parent process has to check the child’s exit status. If not, the child becomes a zombie.

Thank You !