**Module :2- Linux server - Operate running systems**

**20. View Running Processes with ps**

* The ps (process status) command displays information about currently running processes.
* Basic usage:
* ps
  + Shows processes running in the current shell.
* To view **all processes** on the system:
* ps -e

or

ps aux

* + a: show processes for all users.
  + u: show user-oriented format (user, CPU%, memory%).
  + x: include processes not attached to a terminal.
* Example output of ps aux:
* USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND
* root 1 0.0 0.1 22580 1268 ? Ss 10:00 0:01 /sbin/init
* user 2345 0.1 1.5 131200 15432 pts/0 Ss 10:30 0:03 /usr/bin/bash
* You can also use:
* ps -ef
  + Full-format listing of all processes.

**21. Terminate Processes with kill**

* The kill command sends signals to processes, usually to terminate them.
* Basic usage:
* kill PID
  + Sends SIGTERM (signal 15), a graceful termination request.
* To force kill (immediate termination):
* kill -9 PID
  + Sends SIGKILL (signal 9), which forcefully stops the process.
* Example:
* kill 2345
* To find a process's PID before killing:
* ps aux | grep process\_name
* Alternative commands:
  + pkill process\_name — kill process by name.
  + killall process\_name — kill all processes matching the name.

**22. Use top or htop to Monitor System Resources and Processes**

* **top**:
  + Default system monitoring tool in Linux.
  + Run:
  + top
  + Displays dynamic real-time view of system processes, CPU, memory, and swap usage.
  + Interactive commands inside top:
    - P: sort by CPU usage.
    - M: sort by memory usage.
    - q: quit.
* **htop**:
  + Enhanced, user-friendly version of top.
  + Needs to be installed (sudo apt install htop or yum install htop).
  + Run:
  + htop
  + Provides colored, graphical display.
  + Allows mouse interaction, easier process management (kill, renice).
  + Shows CPU cores separately, memory bars, and process tree.

**23. Configure a Linux Computer to Boot to CLI Using systemd and Reboot to Confirm**

* **Background:**  
  Modern Linux systems use systemd to manage services and boot targets.
* To configure the system to boot to CLI (Command Line Interface) instead of GUI (Graphical User Interface), you need to set the default systemd target to multi-user.target (CLI), instead of graphical.target (GUI).

**Steps:**

1. **Check current default target:**
2. systemctl get-default
   * Usually returns graphical.target if booting to GUI.
3. **Set default target to CLI:**
4. sudo systemctl set-default multi-user.target
5. **Reboot the system:**
6. sudo reboot
7. **After reboot, the system boots into CLI mode** (text-based login prompt).

| **Task** | **Command/Action** |
| --- | --- |
| View running processes | ps aux or ps -ef |
| Terminate process by PID | kill PID or kill -9 PID |
| Monitor resources with top | top |
| Monitor resources with htop | htop (install if needed) |
| Set boot to CLI (no GUI) | sudo systemctl set-default multi-user.target and reboot |
| Set boot back to GUI | sudo systemctl set-default graphical.target and reboot |