



RIPHAH  
INTERNATIONAL  
UNIVERSITY

## **Operating System**

### **Lab # 12**

**Name: Samreen Bibi**

**Sap Id: 46484**

**Batch: BSCS 5<sup>th</sup> semester**

**Submitted to: Mam Kausar**

## Lab Tasks:

**Q1:** Which command would you use to find the process ID (PID) of a process named OSLab without running it. After obtaining the PID, which command would you use to kill the process?

### Answer:

#### Finding the Process ID (PID) of "OSLab" Without Running It

To find the PID of a process named "OSLab" without actually starting it, you can use the `pgrep` command. This command searches for all running processes that match the given name and returns their Process ID(s).

#### Killing the Process After Finding the PID

After obtaining the PID of the "OSLab" process, you can use the `kill` command to terminate it. The `kill` command sends a termination signal to the process, asking it to stop.

**Syntax:** `kill <PID>`

**Q2:** How would you write a script that uses a signal trap to handle specific signals, and what is the purpose of a signal trap in such a script?

### Answer:

#### Signal Trap in a Script

A **signal trap** in a script is a way to catch certain signals (like interruptions or requests to stop) and handle them in a custom way, instead of letting the script exit or behave unexpectedly.

```
#!/bin/bash
trap 'echo "SIGINT recieved. Cleaning up..."; exit' SIGINT
trap 'echo SIGTERM recieved. Exiting Gracefully.."; exit' SIGTERM
echo "Script is running. press ctrl+c to send SIGINT or use kill to SIGTERM."
while true
do
sleep 1
done
~
~
~
```

```
~  
"task.sh" [New] 8L, 243B written  
[root@localhost ~]# chmod 777 task.sh  
[root@localhost ~]# ./task.sh  
Script is running. press ctrl+c to send SIGINT or use kill to SIGTERM.  
^CSIGINT recieved. Cleaning up...  
[root@localhost ~]#
```

### Purpose of a Signal Trap:

- **Custom Signal Handling:** Allows you to specify how to handle different signals, such as cleaning up resources, saving work, or gracefully terminating a process.
- **Prevent Unintended Termination:** For example, if a process receives the `SIGINT` (interrupt) signal, a script can handle it to prevent the process from just exiting unexpectedly.
- **Graceful Shutdown:** You can trap signals like `SIGTERM` to clean up resources (like closing files, saving state, etc.) before the process ends.

**THE END**