Lab No: 15 **Date:** 2082/

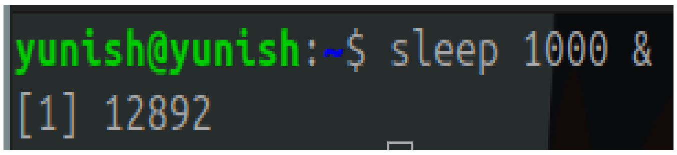
Title: Prepare a lab report to create and terminate process.

In Linux, a process refers to any program or task currently executing on the system, ranging from simple background operations to sophisticated applications. Each process is assigned a unique Process ID (PID), enabling users to monitor, control, and manage it effectively. Processes can be initiated using commands like sleep, and unwanted processes can be stopped by referencing their PID. Mastering these fundamental operations is key to efficient process management in Linux.

Step-by-Step Theory:

1. Starting a Process with sleep 1000 &

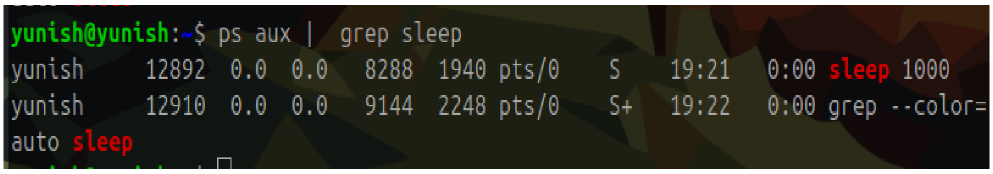
- The sleep 1000 command instructs the system to pause execution for 1000 seconds.
- Adding & at the end runs the command in the background, freeing up the terminal.
- This action spawns a new background process.
- The system assigns a unique PID to this process for identification.



```
yunish@yunish:~$ sleep 1000 &
[1] 12892
```

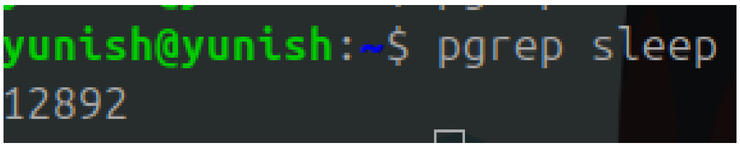
2.Viewing Processes with ps aux | grep sleep

- The ps aux command displays a detailed list of all active processes on the system.
- Piping the output to grep sleep filters the results to show only processes related to the sleep command.
- Note that the grep sleep command itself may appear in the output due to the search operation.


```
yunish@yunish:~$ ps aux |  grep sleep
yunish     12892  0.0  0.0   8288  1940 pts/0    S    19:21   0:00 sleep 1000
yunish     12910  0.0  0.0   9144  2248 pts/0    S+   19:22   0:00 grep --color=
auto sleep
```

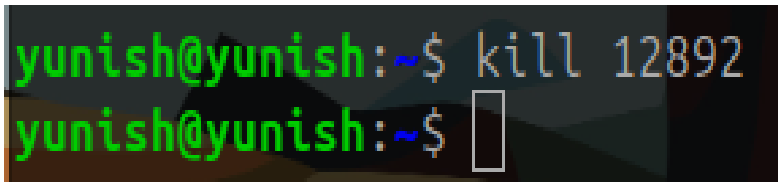
3. Finding the PID **with** pgrep sleep

- The pgrep sleep command searches for processes by name and returns only the PID of the sleep process.
- This method is more precise and cleaner compared to using grep with ps.
- It avoids including irrelevant processes in the output.


```
yunish@yunish:~$ pgrep sleep
12892
```

4. Terminating a Process with kill <PID>

- The kill <PID> command sends a SIGTERM signal to the specified process, requesting it to terminate gracefully.
- This is the preferred approach for stopping a process safely.
- Replace <PID> with the actual process ID obtained from pgrep or ps.


```
yunish@yunish:~$ kill 12892
yunish@yunish:~$ 
```

Conclusion

Linux provides powerful tools for managing processes effectively. Commands like sleep allow you to create processes, while ps and pgrep help monitor them. The kill command enables you to stop processes cleanly. By mastering these commands, you can optimize system resource usage and maintain better control over running applications.