**IT 5423: Lab 4 – Process Management in Linux**

**Total Points: 100**

**Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Prelab:**

**What is a Process?**

An instance of a program is called a Process. In simple terms, any command that you give to your Linux machine starts a new process.



Having multiple processes for the same program is possible.

Types of Processes:

* Foreground Processes: They run on the screen and need input from the user. For example Office Programs
* Background Processes: They run in the background and usually do not need user input. For example Antivirus.

**Running a Foreground Process**

To start a foreground process, you can either run it from the dashboard, or you can run it from the terminal.

When using the Terminal, you will have to wait, until the foreground process runs.



**Running a Background process**

If you start a foreground program/process from the terminal, then you cannot work on the terminal, till the program is up and running.

Particular, data-intensive tasks take lots of processing power and may even take hours to complete. You do not want your terminal to be held up for such a long time.

To avoid such a situation, you can run the program and send it to the background so that terminal remains available to you. Let's learn how to do this -



**Fg**

You can use the command "fg" to continue a program which was stopped and bring it to the foreground.

The simple syntax for this utility is:

fg jobname

Example

1. Launch 'banshee' music player
2. Stop it with the 'ctrl +z' command
3. Continue it with the 'fg' utility.



Let's look at other important commands to manage processes -

**Top**

This utility tells the user about all the running processes on the Linux machine.



Press 'q' on the keyboard to move out of the process display.

The terminology follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Description** | **Example 1** | **Example 2** |
| PID | The process ID of each task | 1525 | 961 |
| User | The username of task owner | Home | Root |
| PR | Priority Can be 20(highest) or -20(lowest) | 20 | 20 |
| NI | The nice value of a task | 0 | 0 |
| VIRT | Virtual memory used (kb) | 1775 | 75972 |
| RES | Physical memory used (kb) | 100 | 51 |
| SHR | Shared memory used (kb) | 28 | 7952 |
| S | StatusThere are five types:          'D' = uninterruptible sleep          'R' = running          'S' = sleeping          'T' = traced or stopped          'Z' = zombie | S | R |
| %CPU | % of CPU time | 1.7 | 1.0 |
| %MEM | Physical memory used | 10 | 5.1 |
| TIME+ | Total CPU time | 5:05.34 | 2:23.42 |
| Command | Command name | Photoshop.exe | Xorg |

**PS**

This command stands for 'Process Status'. It is similar to the "Task Manager" that pop-ups in a Windows Machine when we use Cntrl+Alt+Del. This command is similar to 'top' command but the information displayed is different.

To check all the processes running under a user, use the command -

ps ux



You can also check the process status of a single process, use the syntax -

ps PID



**Kill**

This command **terminates running processes** on a Linux machine.

To use these utilities you need to know the PID (process id) of the process you want to kill

Syntax -

kill PID

To find the PID of a process simply type

pidof Process name

Let us try it with an example.



**NICE**

Linux can run a lot of processes at a time, which can slow down the speed of some high priority processes and result in poor performance.

To avoid this, you can tell your machine to prioritize processes as per your requirements.

This priority is called Niceness in Linux, and it has a value between -20 to 19. The lower the Niceness index, the higher would be a priority given to that task.

The default value of all the processes is 0.

To start a process with a niceness value other than the default value use the following syntax

nice -n 'Nice value' process name



If there is some process already running on the system, then you can 'Renice' its value using syntax.

renice 'nice value' -p 'PID'

To change Niceness, you can use the 'top' command to determine the PID (process id) and its Nice value. Later use the renice command to change the value.

Let us understand this by an example.



**DF**

This utility reports the free disk space(Hard Disk) on all the file systems.



If you want the above information in a readable format, then use the command

'df -h'



**Free**

This command shows the free and used memory (RAM) on the Linux system.



You can use the arguments

free -m to display output in MB

free -g to display output in GB

**Summary:**

* Any running program or a command given to a Linux system is called a process
* A process could run in foreground or background
* The priority index of a process is called Nice in Linux. Its default value is 0, and it can vary between 20 to -19
* The lower the Niceness index, the higher would be priority given to that task

|  |  |
| --- | --- |
| **Command** | **Description** |
| **bg** | To send a process to the background |
| **fg** | To run a stopped process in the foreground |
| **top** | Details on all Active Processes |
| **ps** | Give the status of processes running for a user |
| **ps PID** | Gives the status of a particular process |
| **pidof** | Gives the Process ID (PID) of a process |
| **kill PID** | Kills a process |
| **nice** | Starts a process with a given priority |
| **renice** | Changes priority of an already running process |
| **df** | Gives free hard disk space on your system |
| **free** | Gives free RAM on your system |

**Lab: Now answer the questions below and provide screenshots (multiple screenshots as needed).**

**1) Manage processes [4\*5= 20 points]**

Use the “ps” command to get a snapshot of current running processes.

i) Use “ps -e” to get all processes

ii) Use “ps -f” to get more details

iii) What is the PID of your "bash" process?

iv) Use pgrep or “| grep” to search for “bash”

[screenshots here]

**2) Use “pstree” command to show process in a hierarchy [2\*5= 10 points]**

i) Where is the “bash” shell process? What are its sub-processes?

ii) Use “kill [PID]” to kill a process you like, for example: kill 1002

[screenshot here]

**3. Use the “top” command to see running processes. Keep the terminal open and move it aside. [10 points]**

[screenshot here]

**4) Control jobs for multi-tasking**

**a) Create some jobs [4\*5= 20 points]**

i) Type “cat” command in the terminal, and press ctrl-c to terminate it. Type “jobs” command and you will see nothing.

ii) Type “cat” command in the terminal, and press ctrl-z to stop it (suspend it). Use “jobs” command and what do you see?

iii) Execute some command in the background using the &, for example: nano &

iv) Execute another command (for example, “man nano”) and press ctrl-z to stop it.

[Screenshot]

**b) Use the “jobs” command to see all jobs. [5 points]**

[Screenshot]

**c) Switch between jobs [5 points]**

Use the “fg” command to bring a stopped job to foreground. Or we can specifically use “fg [job-number]” to bring any jobs to foreground. For example, “fg 3”. See the job number from the “jobs” command output.

[Screenshot]

**d) Use ctrl-z to stop a job and send it to background [3\*5 = 15 points]**

Practice switching between the “nano” job and the “man nano” job.

i) Kill the “cat” job (terminate it)

ii) kill %1

iii) Verify using the “jobs” command. Take a screenshot of the terminal which clearly shows all jobs

[screenshot]

**5) Shell startup configuration**

a) Edit the ~/.profile file to add a piece of script to display current date; save the file.

b) Type login in an open terminal to login with the current user and see the effect.

c) Edit the ~/.bashrc file to add a piece of script to display current date; save the file.

d) Open a terminal from the taskbar in GUI and see the effect. [15 points]

[screenshot]