

Lab 5.1 – Linux – Sample Solutions

- **Work through the – *Lab 5.0 Preparation Linux Tutorial - Processes* before attempting this lab sheet.**

1. A process is a unit of work on a Linux system. Each program you run represents one or more processes, and Linux provides commands for viewing and manipulating them. Every process is identified by a numeric process ID or PID.

There are a number of ways of viewing processes in Linux. Give a one line explanation of the commands below and **include a screen shot** of the output of these commands.

Sample Answer: look up each command using the **man** command

- a) **ps**
 - b) **ps -l** (minus lowercase-L)
 - c) **ps -aux**
 - d) **uptime**
 - e) **w**
 - f) **top**
 - g) **free**
2. Linux allows users to run processes either in the foreground or the background. Define **foreground process** and **background process** and explain why you might run a process in the background.

Sample Answer:

By default any process started on the bash command line terminal is run in the foreground which consumes your command prompt. The command prompt remains unresponsive until the command has completed. If the command prompt is unresponsive, you have two options - either terminate the process by the CTRL+C key combination or stop (suspend) the process using CTRL+Z. By pressing CTRL+C, the process will be terminated. By pressing CTRL+Z, the process will be suspended AND the command prompt will be available and responsive to the user.

The process may later be set to run in the background – in which case the process will continue to run AND the command prompt will be available and responsive to the user.

3. Explain the **sleep** command.

Sample Answer:

The sleep command causes a delay for a specified amount of time. The time is specified as a command line parameter, which may be expressed as seconds, minutes, hours or days.

4. Why would you need to use the **sleep** command?

Sample Answer:

It may be used in a script to temporarily pause the script so as to allow another process to complete before continuing the script.

5. What is the difference between the commands **sleep 10** and **sleep 10 &**

Sample Answer:

The sleep 10 command will run as a foreground process.

The sleep 10 & command will run as a background process.

6. Run the **sleep** command with t = 1000 in the foreground. Write down the command you used. **sleep 1000**

- To suspend this process type ctrl-Z
- Type **ps -l** and record the value at the State column (the second column labelled S) for this sleep process. **T**
- To re-start the process but run it in the background type **bg**.
- What is the value of this (sleep) process's State column? **S**

7. The **kill** command is used to terminate unwanted processes. It's syntax is:

Kill -9 PID

- Terminate the sleep 1000 command.
- Type **ps** again to see if it has been removed from the list.

8. Run the **top** command and then suspend it (use ctrl Z to suspend)

- Using the command **ps -l**, what tells you it is a suspended process?

The State column value is T.

- b. Using the command **jobs**, what tells you it is a suspended process?
The status of the job is Stopped (which means suspended).
 - c. Restart it (use **fg** for “Foreground”).
 - d. To quit the **top** process type **q**
9. Create a file called listings.txt in your home directory that contains the list (in long format) of all files in one of your directories.
ls -l > listing.txt
10. Using a Linux command, how would you append your name to the end of the file listings.txt?
echo Martin >> listing.txt
11. What is the command to remove the write permission from the listings.txt file, for the owner of the file?
chmod u-w listing.txt
12. Having completed step 11. What happens if you try to append your name to the listings.txt file in the normal way?
The following error: bash: listings.txt: Permission denied
13. Rewrite the command to redirect the output of the command you wrote in the previous step to an error file.
echo Martin 2> error.txt >> listing.txt

Note: if you type the following it would not work because the order matters.

```
echo Martin >> listing.txt 2> error.txt
```

Remember the command line terminal processes a command from left to right and from top to bottom.