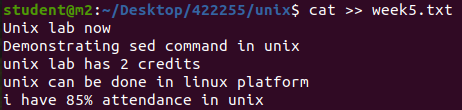
**WEEK-05: (ASSIGNMENT-05):**

Implement shell scripts using sed, ps, kill, lp, lprm, lpstat commands

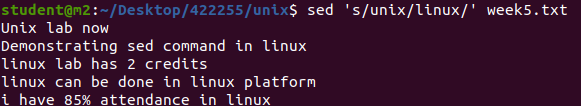
1. Command : sed

SED command in UNIX stands for stream editor and it can perform lots of functions on file like searching, find and replace, insertion or deletion. Though most common use of SED command in UNIX is for substitution or for find and replace. By using SED you can edit files even without opening them, which is much quicker way to find and replace something in file, than first opening that file in VI Editor and then changing it.

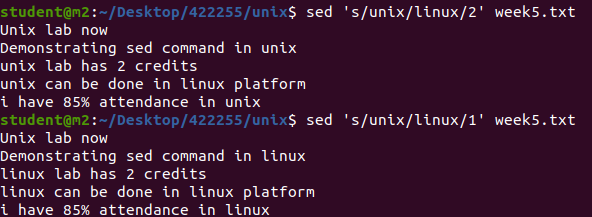
* SED is a powerful text stream editor. Can do insertion, deletion, search and replace(substitution).
* SED command in unix supports regular expression which allows it perform complex pattern matching.****

**1. Replacing or substituting string :** Sed command is mostly used to replace the text in a file. The below simple sed command replaces the word “unix” with “linux” in the file.

**$sed 's/unix/linux/' filename**

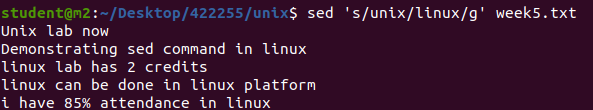
****

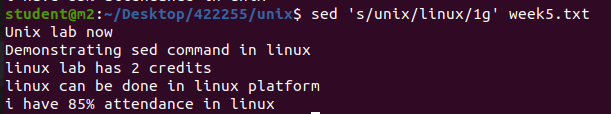
**2. Replacing the nth occurrence of a pattern in a line :** Use the /1, /2 etc flags to replace the first, second occurrence of a pattern in a line. The below command replaces the second occurrence of the word “unix” with “linux” in a line.



**3. Replacing all the occurrence of the pattern in a line :** The substitute flag /g (global replacement) specifies the sed command to replace all the occurrences of the string in the line.

**$sed 's/unix/linux/g' filename.txt**

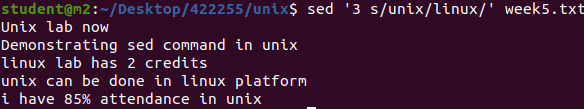


**4. Replacing from nth occurrence to all occurrences in a line :** Use the combination of /1, /2 etc and /g to replace all the patterns from the nth occurrence of a pattern in a line. The following sed command replaces the third, fourth, fifth… “unix” word with “linux” word in a line. 

**5. Parenthesize first character of each word :** This sed example prints the first character of every word in parenthesis.

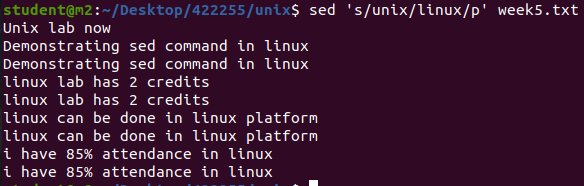


**6. Replacing string on a specific line number :** You can restrict the sed command to replace the string on a specific line number.

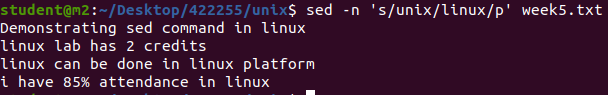


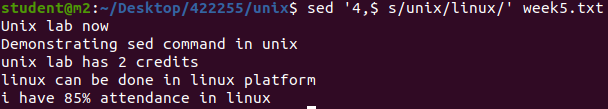
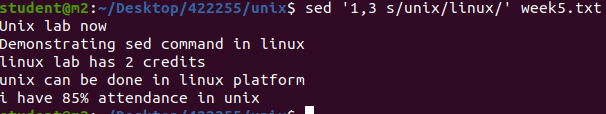
**7. Duplicating the replaced line with /p flag :** The /p print flag prints the replaced line twice on the terminal. If a line does not have the search pattern and is not replaced, then the /p prints that line only once.

**$sed 's/unix/linux/p' filename.txt**

8. **Printing only the replaced lines :** Use the -n option along with the /p print flag to display only the replaced lines. Here the -n option suppresses the duplicate rows generated by the /p flag and prints the replaced lines only one time.

**$sed -n 's/unix/linux/p' filename.txt**

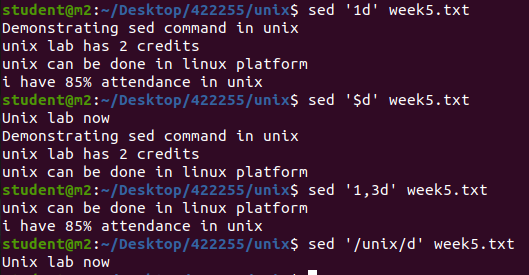
9. **Replacing string on a range of lines :** You can specify a range of line numbers to the sed command for replacing a string.

**$sed '1,3 s/unix/linux/' file.txt**

Here $ indicates the last line in the file. So the sed command replaces the text from second line to last line in the file.

**10. Deleting lines from a particular file :** SED command can also be used for deleting lines from a particular file. SED command is used for performing deletion operation without even opening the file

**sed 'nd' filename.txt**



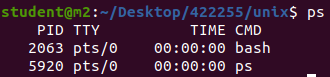
1. Command : ps

The `**ps`** command, which stands for “process status,” is like a computer tool that helps you see what’s happening inside your Linux computer. Imagine your computer is doing several things simultaneously, like running different programs or apps. These are the processes and the `**ps`** command lets you take a quick look at them. When you use it without any special instructions, it shows you the processes that are connected to the window or screen you are currently using. But here’s where it gets interesting: you can make the ps command show you exactly what you want to know by giving it special instructions, called options. These options let you customize the information you see, like finding out which programs are using the most computer power or checking what a specific user is doing.

**1) Simple process selection :** Shows the processes for the current shell – Where,

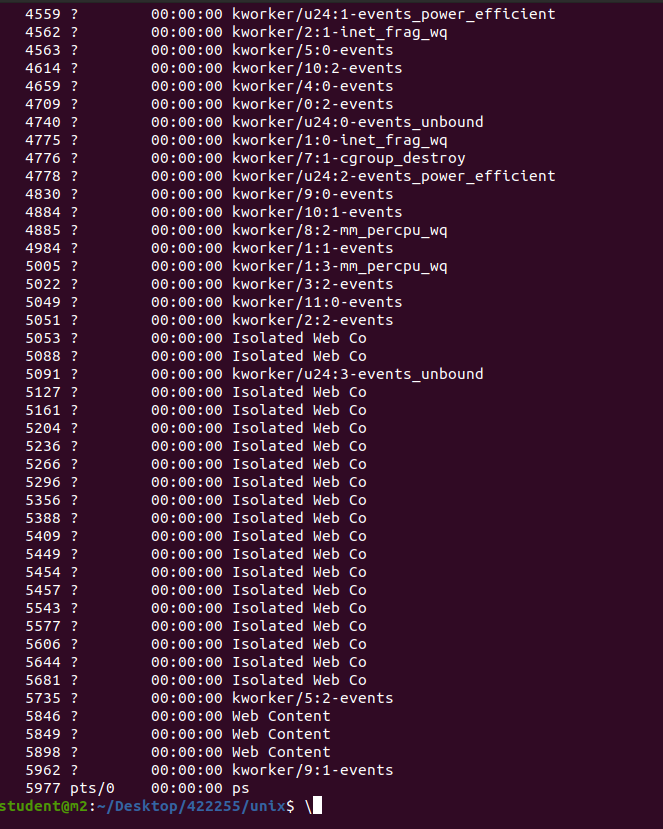
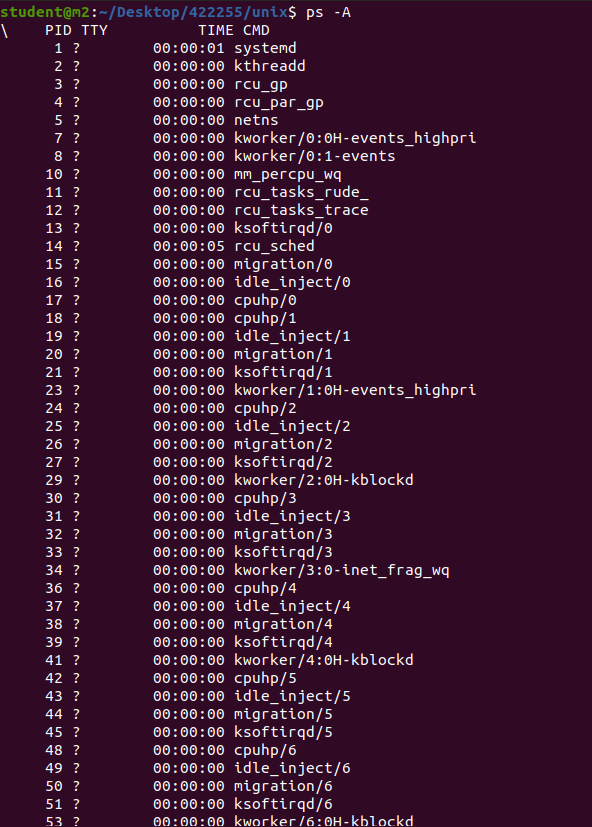
* **PID –** the unique process ID
* **TTY –** terminal type that the user is logged into
* **TIME –** amount of CPU in minutes and seconds that the process has been running
* **CMD –** name of the command that launched the process.

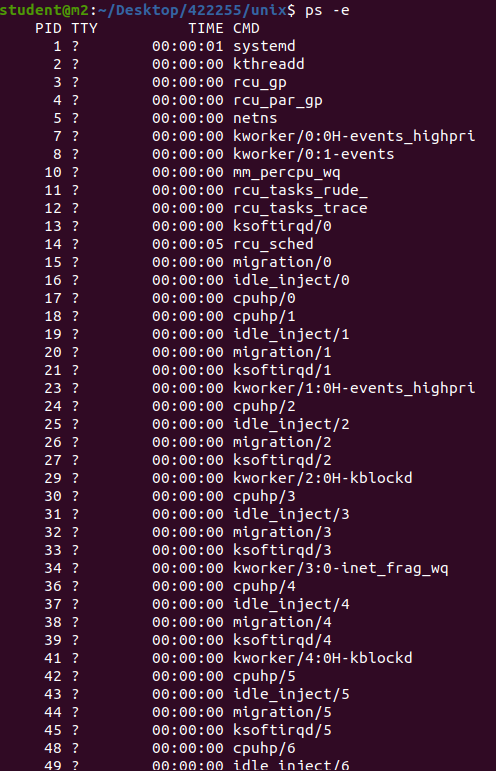
**Note –** Sometimes when we execute **ps** command, it shows TIME as 00:00:00. It is nothing but the total accumulated CPU utilization time for any process and 00:00:00 indicates no CPU time has been given by the kernel till now. In above example we found that, for bash no CPU time has been given. This is because bash is just a parent process for different processes which needs bash for their execution and bash itself is not utilizing any CPU time till now.



**2) View All Running Processes in Linux.**

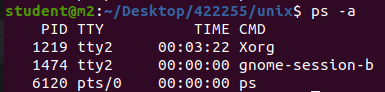
To view all running processes, use either of the following options with the `**ps`** command:



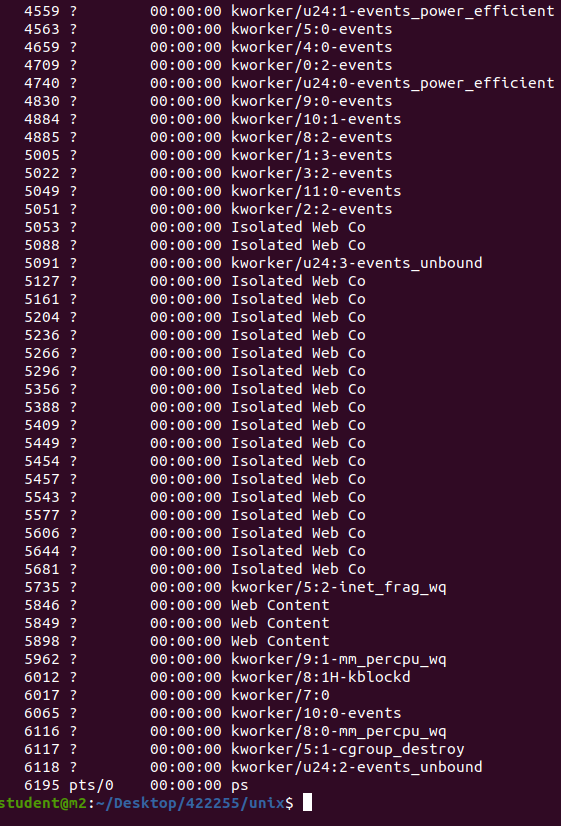
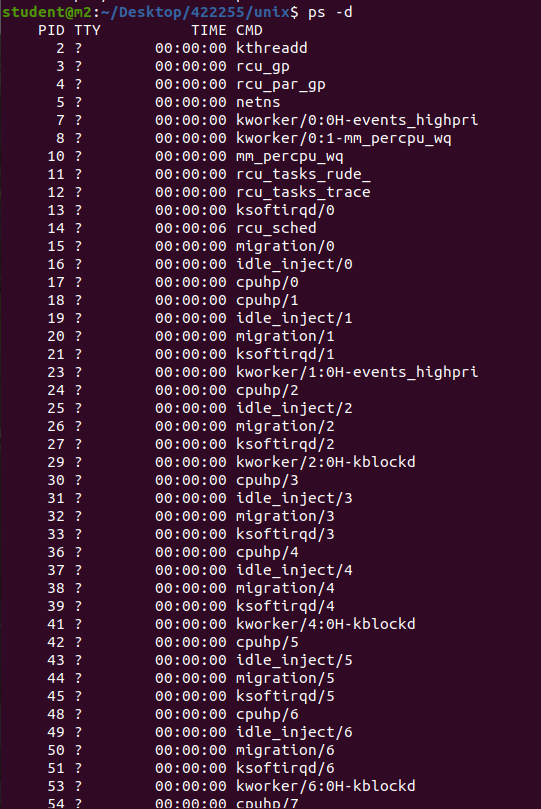


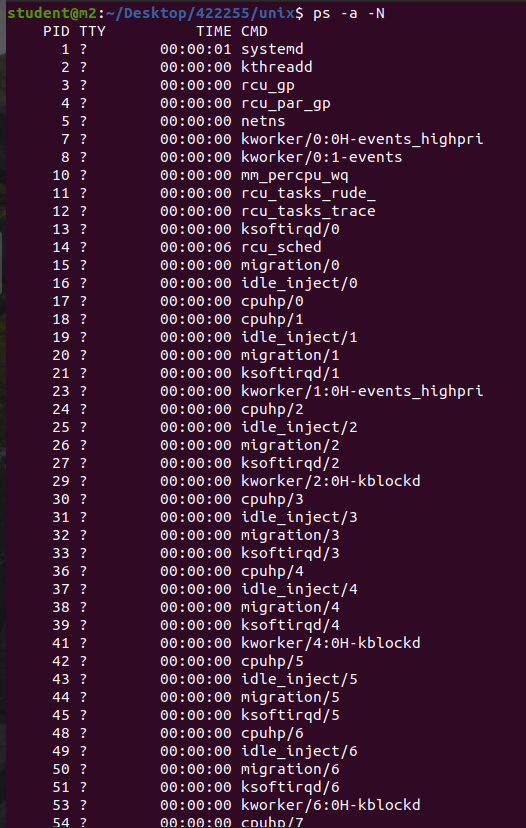
**3) List Processes Not associated with a Terminal in Linux**

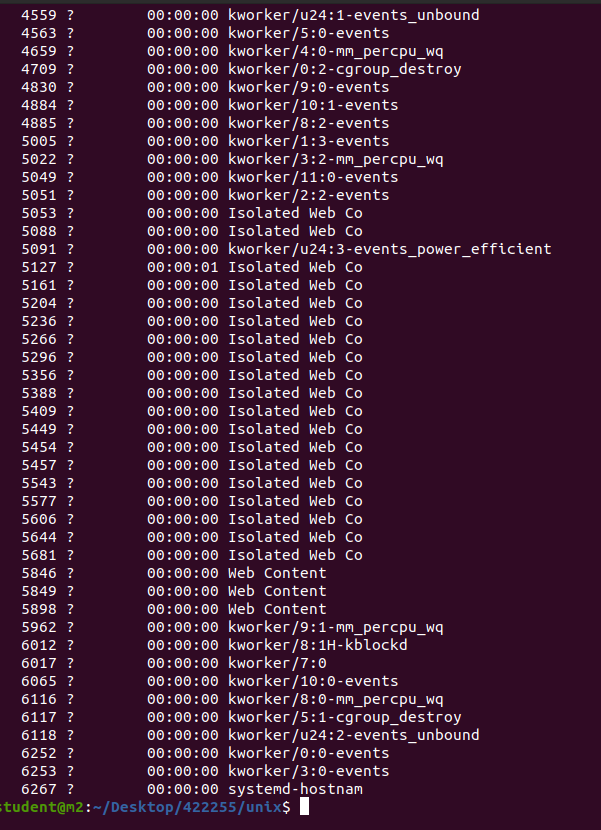
View all processes except both session leaders and processes not associated with a terminal.



**4) List All The Processes Except Session Leaders in Linux**

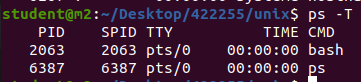
A session leader is a process that initiates other processes. View processes except session leaders:

**5) List All Processes Except those that Fulfill the Specified Conditions (negates the selection)**

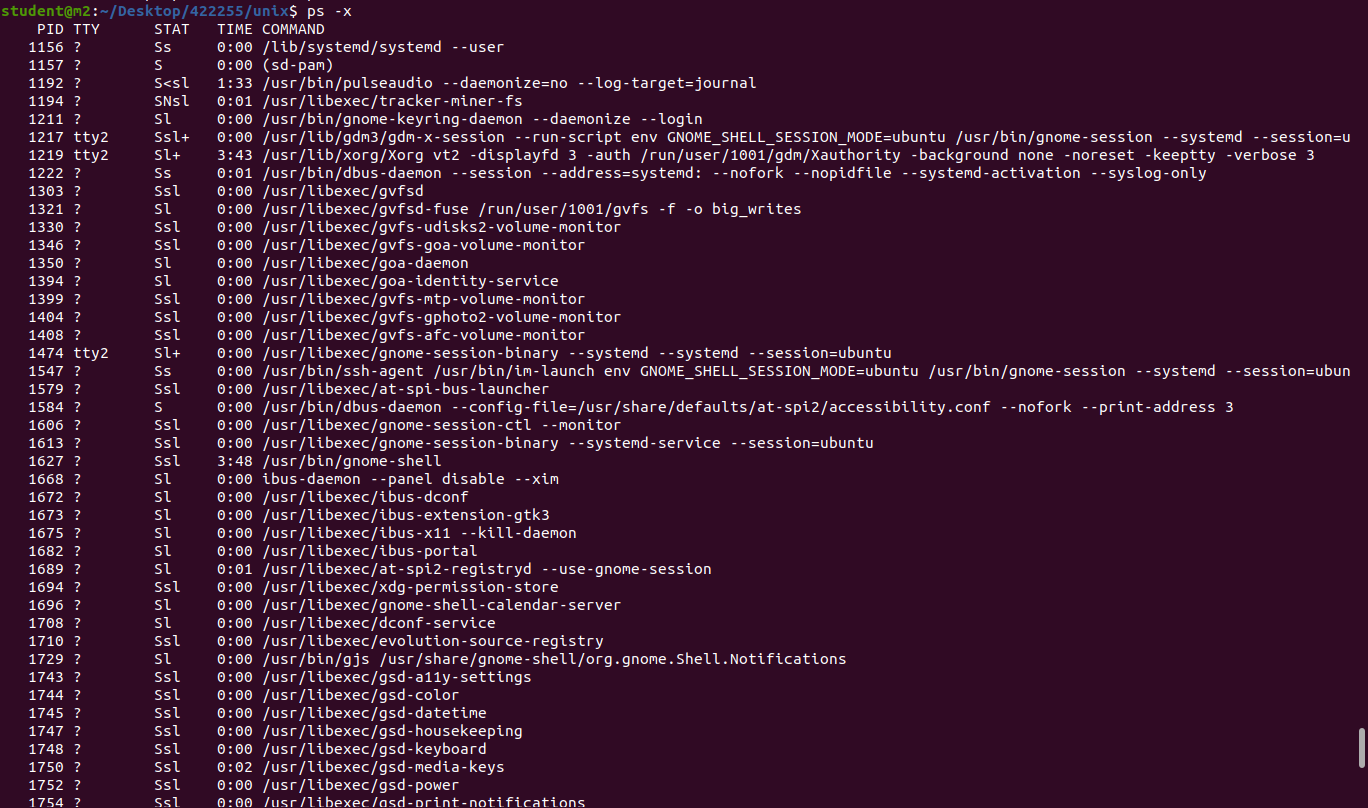


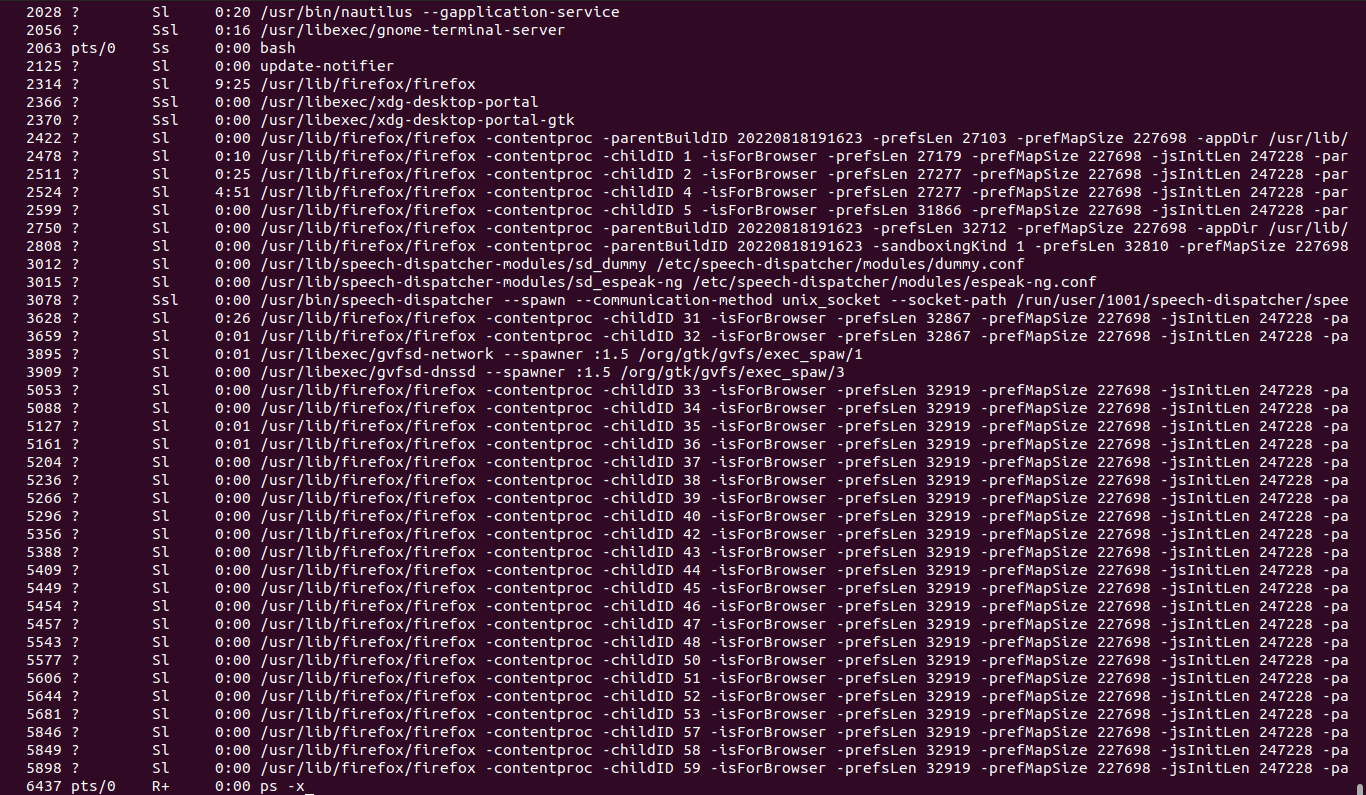
**6) List All Processes Associated with this Terminal in Linux**

ps -T

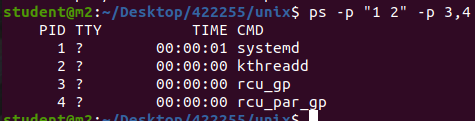


**7) View All Processes Owned By You**

Processes i.e same EUID as ps which means runner of the ps command, root in this case –

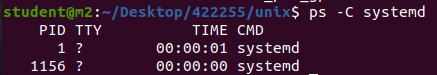
**8) Process selection by list**

Here we will discuss how to get the specific processes list with the help of ps command. These options accept a single argument in the form of a blank-separated or comma-separated list. They can be used multiple times.

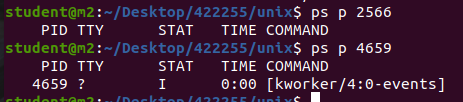
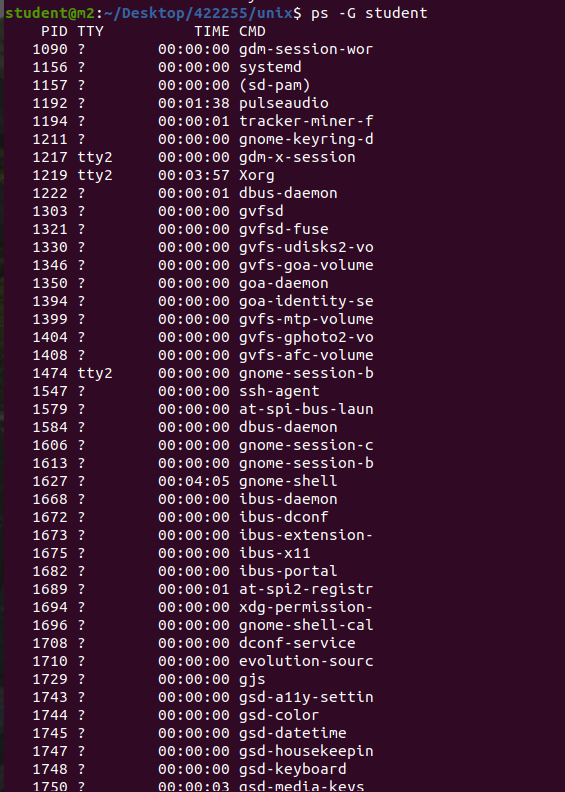
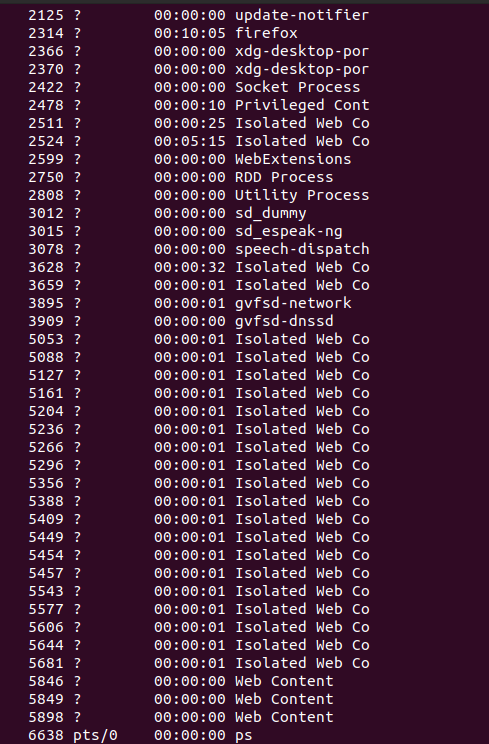


**9) Select Processes by Command Name**

This selects the processes whose executable name is given in cmdlist. There may be a chance you won’t know the process ID and with this command it is easier to search.



**10) Select by group ID or name**

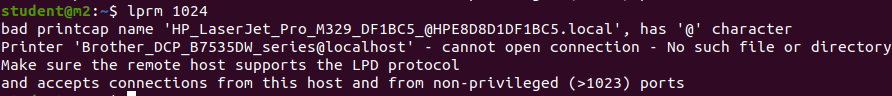
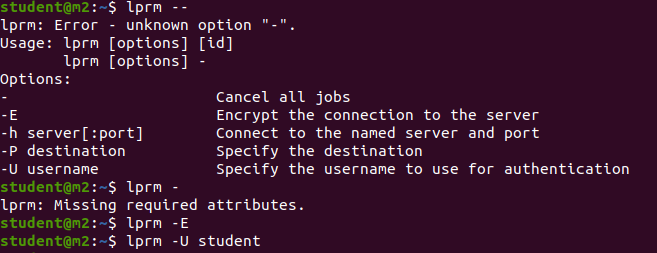
The group ID identifies the group of the user who created the process.

1. Command : lp

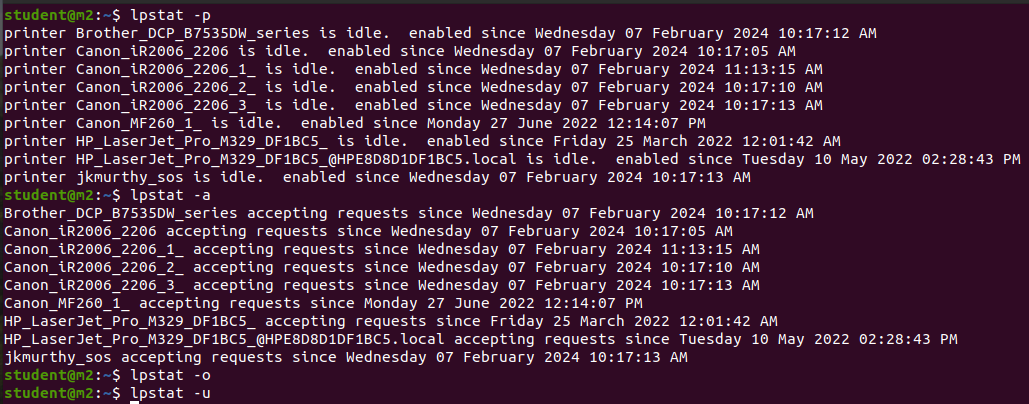
**lp** command arranges for the files specified by the Files parameter and their associated information (called a request) to be printed by a line printer. If you do not specify a value for the Files parameter, the lp command accepts standard input. The file name - (dash) represents standard input and can be specified on the command line in addition to files. The lp command sends the requests in the order specified. If the job is submitted to a local print queue, the lp command displays the following to standard output: Job number is: nnn where nnn is the assigned job number. To suppress the job number use the -s flag.

1. Command : lprm

Removes jobs from the line printer spooling queue.



1. Command : lpstat



1. Command : kill

