**UNIX LAB**

**ASSIGNMENT 3**

Name: Sreekesh Iyer Class/Roll No. D10A/24

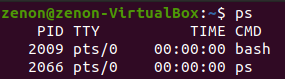
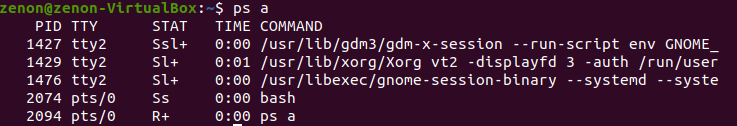
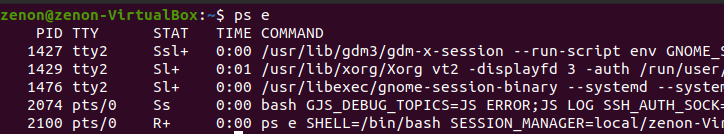
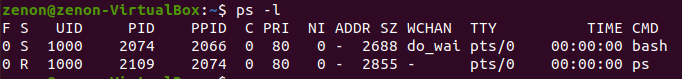
**Aim:** To understand process management UNIX commands in Linux shell.

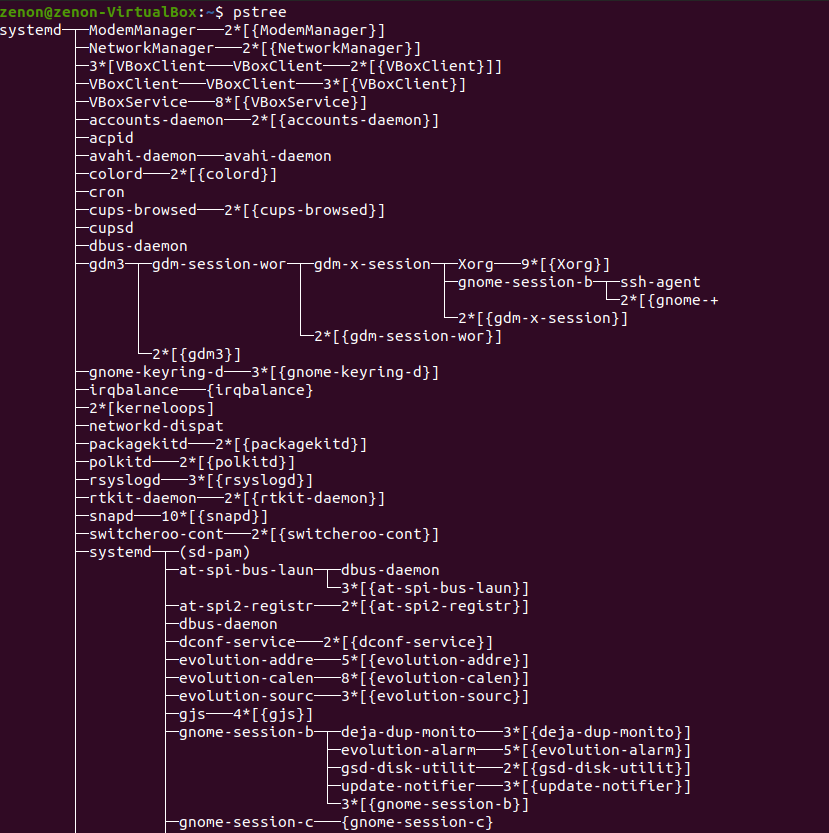
**ps:** Linux provides us with a utility called ps for viewing information related to the processes on a system which stands as an abbreviation for “Process Status”. ps command is used to list the currently running processes and their PIDs along with some other information depends on different options. It reads the process information from the virtual files in the /proc file system. /proc contains virtual files, this is the reason it’s referred to as a virtual file system.  
Options:

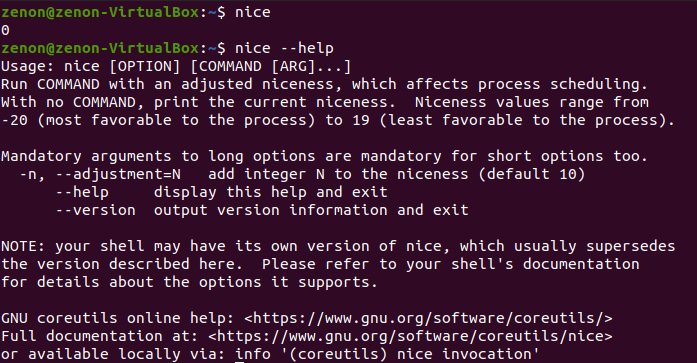
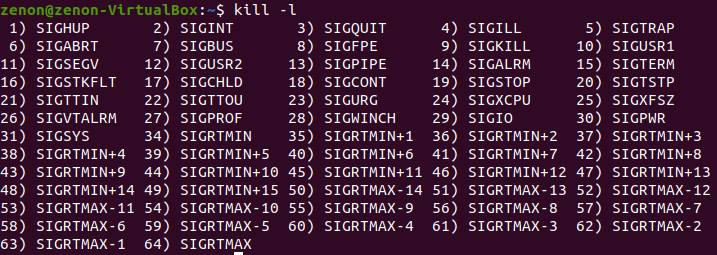
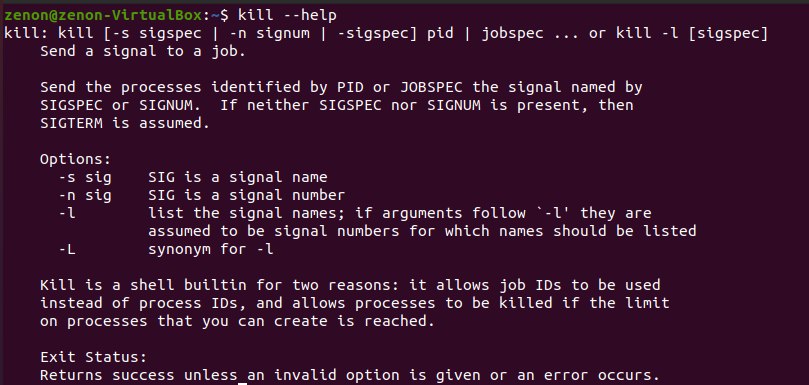
u (username) - displays processes running at the moment

t (terminal name) - displays processes running on the mentioned terminal

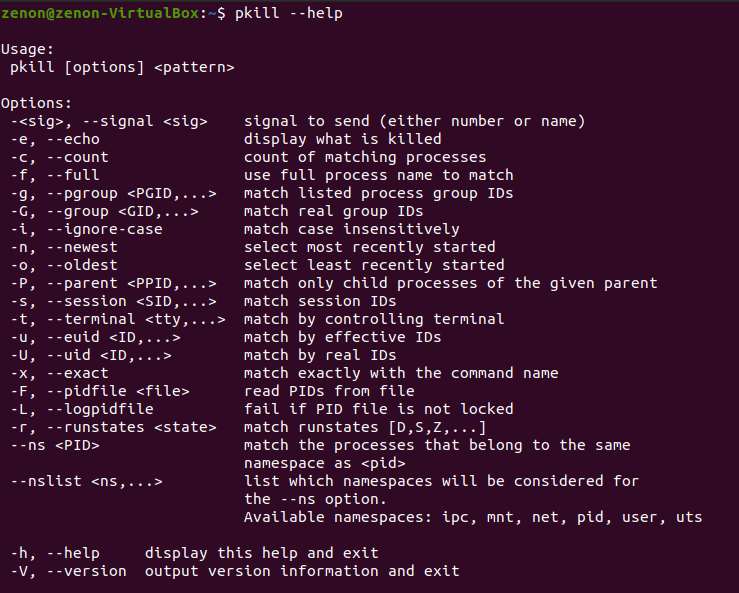
l - displays running processes in long format with memory related information

**pstree**: It is similar to *ps*, but instead of listing the running processes, it shows them in a tree. The tree-like format is a more convenient way to display the processes hierarchy and makes the output more visually appealing.  


**nice**: nice command in Linux helps in the execution of a program/process with modified scheduling priority. It launches a process with a user-defined scheduling priority. In this, if we give a process a higher priority, then Kernel will allocate more CPU time to that process.  
  
  
  
  
**kill**: Once in a while, some applications may start behaving erratically and become unresponsive or start consuming a lot of system resources. Unresponsive applications cannot be restarted because the original application process never shuts down completely. The only solution is to either restart the system or kill the application process.  
  
The kill command sends a signal to specified processes or process groups, causing them to act according to the signal. When the signal is not specified, it defaults to -15 (-TERM), which terminates the process.  
  


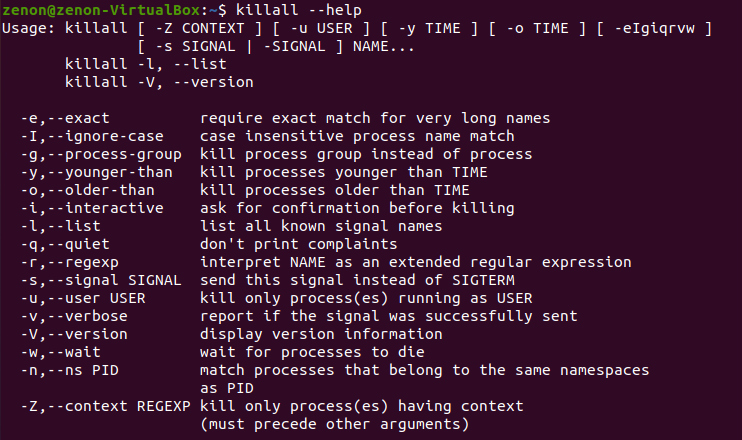
**pkill**: pkill is a command-line utility that sends signals to the processes of a running program based on given criteria. The processes can be specified by their full or partial names, a user running the process, or other attributes.

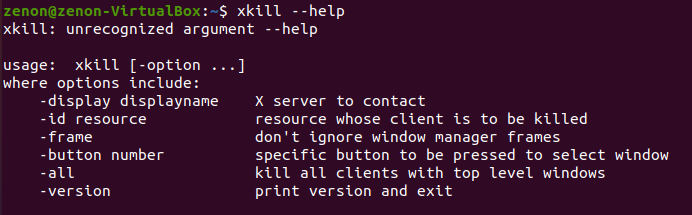
The pkill command is a part of the procps (or procps-ng) package, which is pre-installed on nearly all Linux distributions. pkill is basically a wrapper around the pgrep program that only prints a list of matching processes.  
  


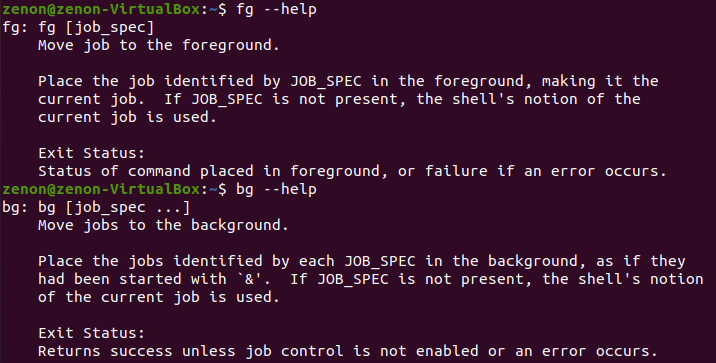
**killall**: killall sends a signal to all processes running any of the specified commands. If no signal name is specified, SIGTERM is sent.

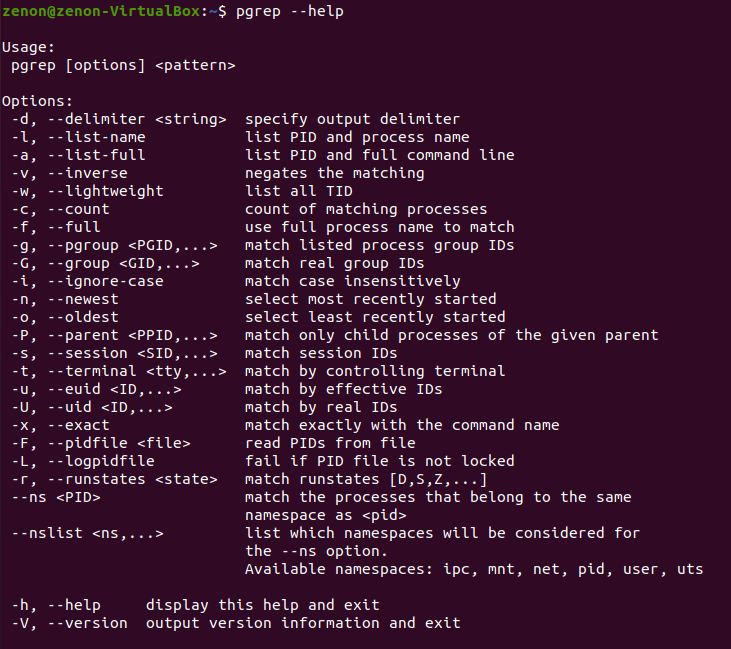
Signals can be specified either by name (e.g., -HUP or -SIGHUP) or by number (e.g., -1) or by option -s.

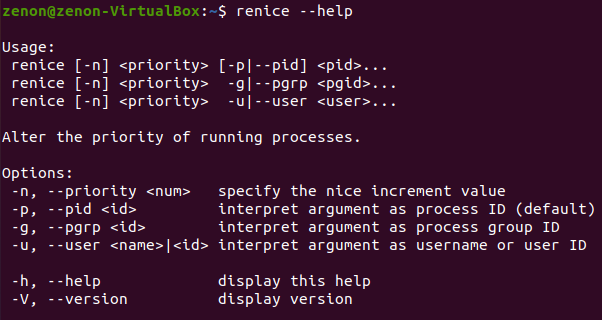
killall returns a zero return code if at least one process was killed for each listed command, or no commands were listed and at least one process matched the -u and -Z search criteria. killall returns non-zero otherwise.

A killall process never kills itself but may kill other killall processes.  
  
  
  
**xkill**: Command xkill is used to kill a process on the X server without passing the process name or PID. It forces the X server to close the communication with its clients, which ultimately kills its clients by its X resource. In short, xkill instructs the X server to terminate the client.

  
  
**fg**:fg command in Linux used to put a background job in foreground.  
**bg**: bg command in Linux is used to resume execution of a suspended process as if they had been started with &. Use the bg command to restart a stopped background process.

  
  
**pgrep**: pgrep is a command-line utility that allows you to find the process IDs of a running program based on given criteria. It can be a full or partial process name, a user running the process, or other attributes.

  
  
**renice**: the renice command allows you to change and modify the scheduling priority of an already running process. Linux Kernel schedules the process and allocates CPU time accordingly for each of them.



**free**: It shows the total amount of free and used physical memory in the system as well as the buffers used by the kernel.   
  
