**Experiment No :** 04

**Experiment Name** : Process Handling in Linux

**Objective :**

Our aim is to learn about Process handling systems , how the work and what are the advantages of using Linux terminal and understand it’s natural Phenomenon . .

**Working Procedure** :

Step 1 : First we need to open up our Linux and Search in Search bar opotion to find Terminal .

Step 2 : Now write the different Commands and after finishing it clear it ..

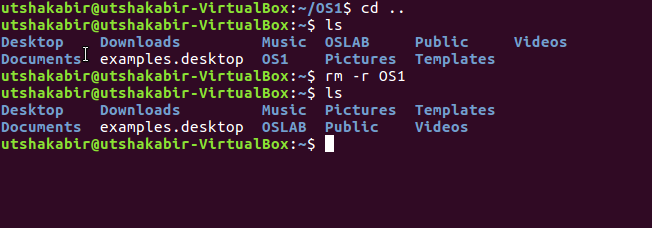
Some Specific Commands are given below –

**cd**-> Used to navigate the directories.  You can move to any location by path.

1. **cd**This will move you back to your home, same as cd ~
2. **cd ..**This will take you back exactly one directory.  Starting in /home/justin/Desktop, cd .. will put me into /home/justin.  This can be expanded upon, cd ../../ from the Desktop location instead will move me 2 back, from my Desktop to /home.

**cd foldername/**This will move you forward to the given folder *in your current folder.*  Take note of the missing prefix / it is an important omission.  if I am in /home/justin and I want to get to Desktop, I must type cd Desktop/ without the / before Desktop.  Typing / before it places us in the root of file system, which is incorrect.

1. **cd /some/other/path**This will take you to the specified folder path, supposing it exists as typed exactly.  Don't forget your tab completion!

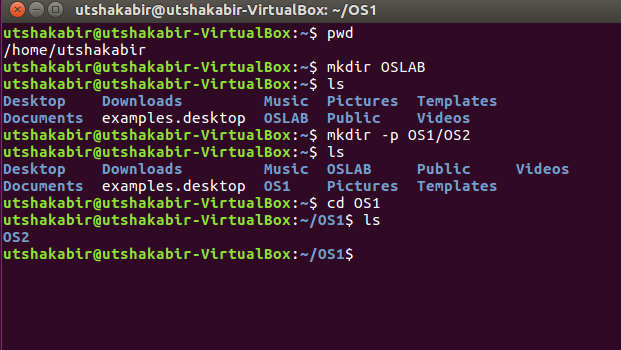


**mkdir**-> Create directories

1. **mkdir folder\_name**Creates the folder with the specified name
2. **mkdir -p /path/to/folder/name**Creates each folder as necessary.  To create folder /home/justin/newfolder/2ndfolder, and only /home/justin exists, using **mkdir -p**will make both directories newfolder and 2ndfolder.

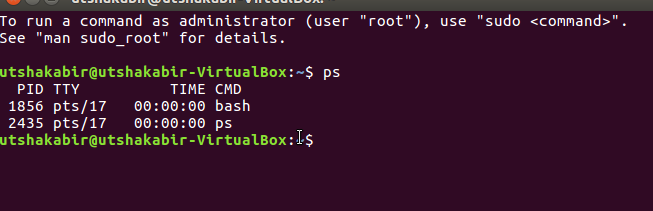
**rm** -> Remove files

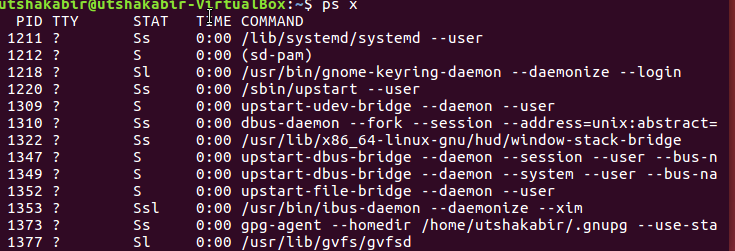
1. For all intents and purposes, removing files via **rm** is permanent.  It does not use the Trash bin.  Use with caution and make sure you are deleting explicitly what you want, not what you think you want.  If you decide to get fancy with your delete commands, it's probably going to come back to bite you.
2. **rm file** Remove the specified file from the system**.**
3. **rm -r folder**  Remove the specified folder from the system
4. **rm -rf folder**Removes the specified folder **forcefully** from the system.  This command can severely break your configuration if used incorrectly as it will not prompt you if something critical is being deleted.  If you have to use this, chances are something more is broken or there was a mistake made.  **This should only be used as an absolute last resort method and is not recommended.**



**ps** -> List processes

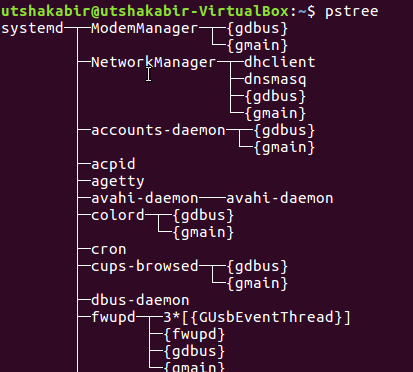
1. **ps aux/ps x->**List all processes in detail running on the system, including user, Process ID (PID), and name of process.  Using this, one can view their process list and if necessary, kill unnecessary or stalled processes.





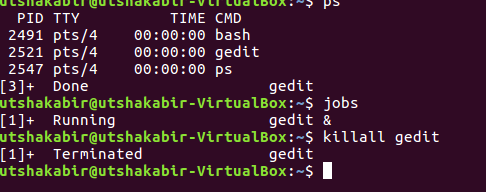
**tee**-> Send output to both a file and the terminal

1. **tee** is used in conjunction with a ' | ' in order to take the command output and send it elsewhere.  This is useful if there are errors which fly by the screen before you can read them, this way whatever goes on the screen is also captured to a file.
2. Example: **dmesg | tee boot.txt** would run the command **dmesg** which shows the initial boot info, and the ' **|**' sends the output of **dmesg**to **tee**, which then does its job by sending it to the terminal and to the log file **boot.txt**.



**kill / killall / xkill** -> Kill offending processes.

1. **kill PID**PID is a number referencing the offending process.  One should obtain the PID from a command like **ps aux**.  If a process refuses to die, one can alternatively specify **kill -9 PID** which should terminate the process by any means, even uncleanly or if it will mess up the system.
2. **killall program**  Killall kills \*by name\* all instances of said program.  If there are for example 3 firefox sessions open, **killall firefox**will do just that; kill all firefox sessions.  **kill** would simply take the specified PID of the offending firefox process you wish to kill, and kill that one only.
3. **xkill** is a GUI way to click and kill windows.  Typing in **xkill** should present a skull-and-crossbones icon, and the next window clicked on will be killed



**Discussion :**

In these Lab ,We find out Some commands to work on process Handling.Process handling is very efficient for us .It gives Us permission to access all of these Works by not using any memory.And we can use the Commands from Command prompt and then Execute it for the Further Process..