MODULE 1

Basic commands

1) Commands are actually files containing programs, often written in C. How will you find out in which directory does the file corresponding to the man command resides?

```
balamurugan@balamurugan-Lenovo-E41-25:~$ which man
/usr/bin/man
```

2) How will you find out what is the use of the ps command.

By using man command we can find description of a command.

```
balamurugan@balamurugan-Lenovo-E41-25:~$ man ps
```

```
PS(1)                                   User Commands                                   PS(1)

NAME
       ps - report a snapshot of the current processes.

SYNOPSIS
       ps [options]

DESCRIPTION
       ps displays information about a selection of the active processes.  If you want a repetitive update of the selection and the displayed
       information, use top instead.

       This version of ps accepts several kinds of options:

       1   UNIX options, which may be grouped and must be preceded by a dash.
       2   BSD options, which may be grouped and must not be used with a dash.
       3   GNU long options, which are preceded by two dashes.

       Options of different types may be freely mixed, but conflicts can appear.  There are some synonymous options, which are functionally
       identical, due to the many standards and ps implementations that this ps is compatible with.

       Note that ps -aux is distinct from ps aux.  The POSIX and UNIX standards require that ps -aux print all processes owned by a user named x,
       as well as printing all processes that would be selected by the -a option.  If the user named x does not exist, this ps may interpret the
       command as ps aux instead and print a warning.  This behavior is intended to aid in transitioning old scripts and habits.  It is fragile,
       subject to change, and thus should not be relied upon.

       By default, ps selects all processes with the same effective user ID (euid=EUID) as the current user and associated with the same terminal
       as the invoker.  It displays the process ID (pid=PID), the terminal associated with the process (tname=TTY), the cumulated CPU time in
       [DD-]hh:mm:ss format (time=TIME), and the executable name (ucmd=CMD).  Output is unsorted by default.

       The use of BSD-style options will add process state (stat=STAT) to the default display and show the command args (args=COMMAND) instead of
       the executable name.  You can override this with the PS_FORMAT environment variable.  The use of BSD-style options will also change the
       process selection to include processes on other terminals (TTYs) that are owned by you; alternately, this may be described as setting the
       selection to be the set of all processes filtered to exclude processes owned by other users or not on a terminal.  These effects are not
       considered when options are described as being "identical" below, so -M will be considered identical to Z and so on.

       Except as described below, process selection options are additive.  The default selection is discarded, and then the selected processes are
```

General Purpose Utilities in Linux

1)Display the calender for the month of March 2012

```
balamurugan@balamurugan-Lenovo-E41-25:~$ cal 3 12
      March 12
Su Mo Tu We Th Fr Sa
        1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31
```

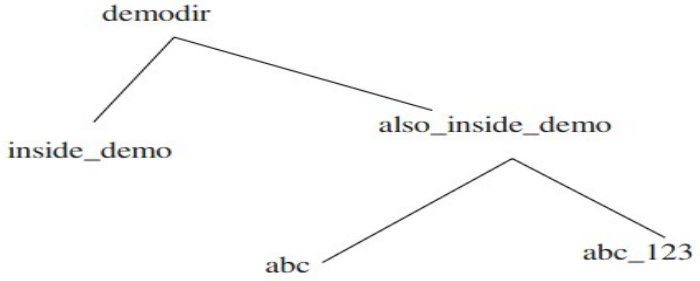
2)List all the files and directories of the directory /usr/lib on the terminal. Now put the same information in a file named results.

Display the contents of the file results now.

```
balamurugan@balamurugan-Lenovo-E41-25:~$ ls
bala.txt  Desktop  Documents  Downloads  Music  personalfiles  Pictures  Public  snap  Templates  Videos  week1.odt
balamurugan@balamurugan-Lenovo-E41-25:~$ pwd
/home/balamurugan
balamurugan@balamurugan-Lenovo-E41-25:~$ ls > results
balamurugan@balamurugan-Lenovo-E41-25:~$ ls
bala.txt  Desktop  Documents  Downloads  Music  personalfiles  Pictures  Public  results  snap  Templates  Videos  week1.odt
balamurugan@balamurugan-Lenovo-E41-25:~$ cat results
bala.txt
Desktop
Documents
Downloads
Music
personalfiles
Pictures
Public
results
snap
Templates
Videos
week1.odt
```

File Systems

1)Make a directory structure like this in your home directory



```
balamurugan@balamurugan-Lenovo-E41-25:~$ mkdir demodir
balamurugan@balamurugan-Lenovo-E41-25:~$ ls
bala.txt  demodir  Desktop  Documents  Downloads  Music  personalfiles  Pictures  Public  results  snap  Templates  Videos  week1.odt
balamurugan@balamurugan-Lenovo-E41-25:~$ cd demodir
balamurugan@balamurugan-Lenovo-E41-25:~/demodir$ mkdir inside_demo also_inside_demo
balamurugan@balamurugan-Lenovo-E41-25:~/demodir$ ls
also_inside_demo  inside_demo
balamurugan@balamurugan-Lenovo-E41-25:~/demodir$ cd also_inside_demo
balamurugan@balamurugan-Lenovo-E41-25:~/demodir/also_inside_demo$ mkdir abc abc_123
balamurugan@balamurugan-Lenovo-E41-25:~/demodir/also_inside_demo$ ls
abc  abc_123
```

2) Remove the also_inside_demo directory

```
balamurugan@balamurugan-Lenovo-E41-25:~$ cd demodir
balamurugan@balamurugan-Lenovo-E41-25:~/demodir$ ls
also_inside_demo  inside_demo
balamurugan@balamurugan-Lenovo-E41-25:~/demodir$ rm -rf also_inside_demo
balamurugan@balamurugan-Lenovo-E41-25:~/demodir$ ls
inside_demo
```

Redirection of Pipes

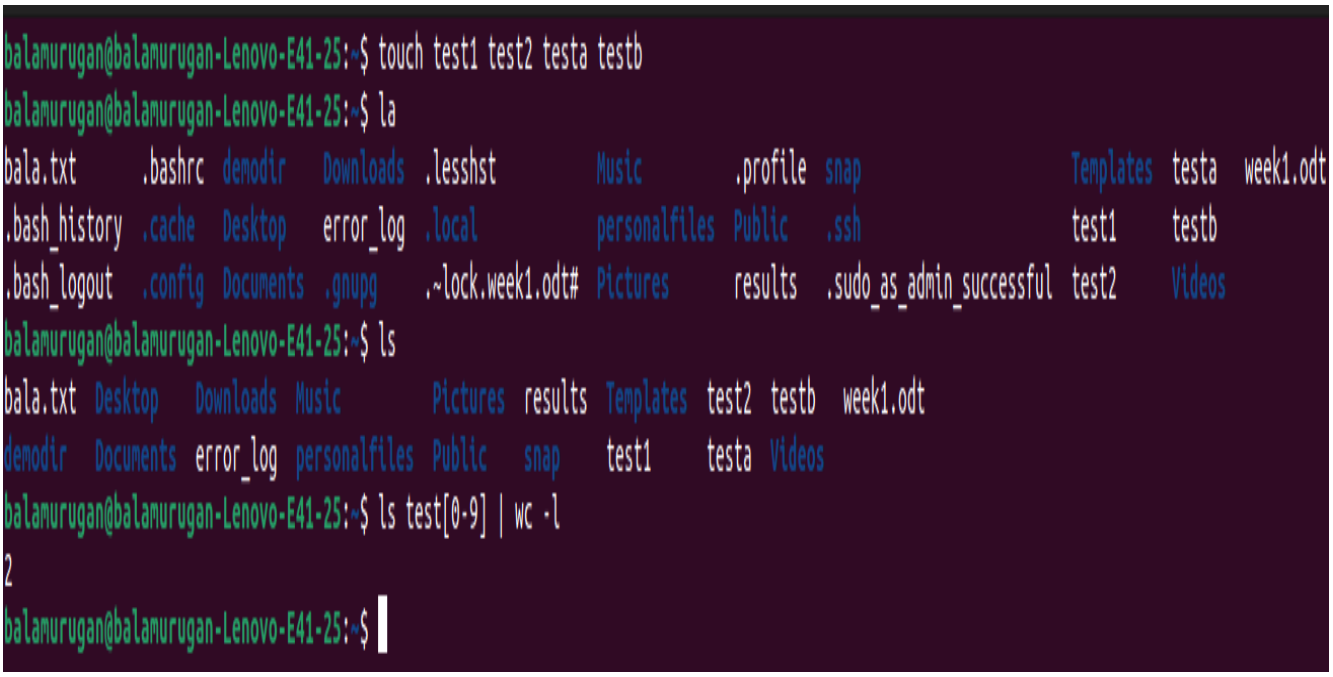
1) Create a file name error_log in your current directory. Suppose you do not have any file named aa11 in your current directory.

How can you redirect the error message to the file error_log when we apply the command "wc -l aa11" ?

How can you ensure that all the error log are appended to the error_log file?

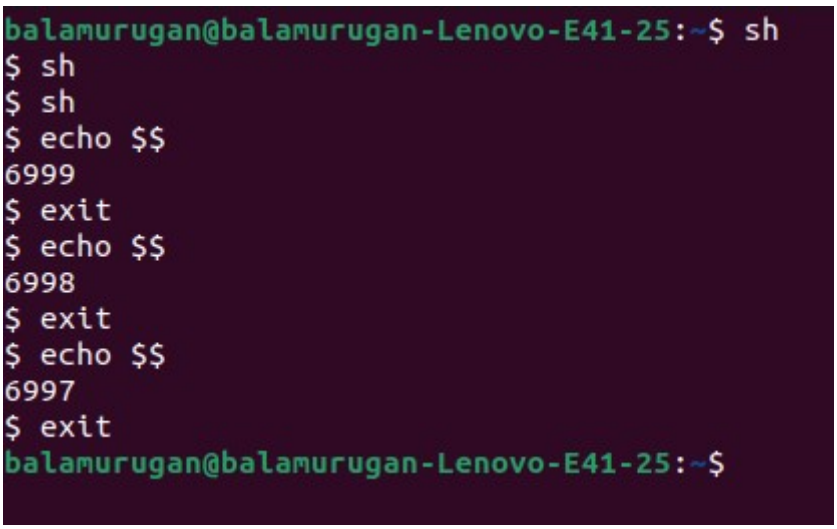
```
balamurugan@balamurugan-Lenovo-E41-25:~$ >error_log
balamurugan@balamurugan-Lenovo-E41-25:~$ ls
bala.txt  Documents  Music          Public     Templates
demodir   Downloads  personalfiles  results    Videos
Desktop   error_log  Pictures       snap       week1.odt
balamurugan@balamurugan-Lenovo-E41-25:~$ wc -l aa11 2>> error_log
balamurugan@balamurugan-Lenovo-E41-25:~$ cat error_log
wc: aa11: No such file or directory
```

2) Create files named test1, test2, testa, testb
How can you count the number of files starting with test and then having only one digit in their name using only a single line command ?



Linux process

1) Open a terminal. Now spawn three shell processes one after another i.e. first spawn one shell, then from the spawned shell, spawn one new shell and so on. Now, how can you see the PID of the current shell ? How can you see the PID of the shell which is the grandparent of the current shell?



2) How can you see all the processes (both system & user processes) in your computer?

The output can be quite large. How can you view the output as multipage output ?

How can you store the output in a file named process_info?

```
balamurugan@balamurugan-Lenovo-E41-25:~$ ps
    PID TTY          TIME CMD
   6974 pts/0    00:00:00 bash
   7414 pts/0    00:00:00 ps
balamurugan@balamurugan-Lenovo-E41-25:~$ ps -f
UID          PID    PPID  C STIME TTY          TIME CMD
balamur+    6974    6956  0 19:41 pts/0    00:00:00 bash
balamur+    7455    6974  0 19:58 pts/0    00:00:00 ps -f
balamurugan@balamurugan-Lenovo-E41-25:~$ ps -f | more
UID          PID    PPID  C STIME TTY          TIME CMD
balamur+    6974    6956  0 19:41 pts/0    00:00:00 bash
balamur+    7456    6974  0 19:58 pts/0    00:00:00 ps -f
balamur+    7457    6974  0 19:58 pts/0    00:00:00 more
balamurugan@balamurugan-Lenovo-E41-25:~$ ps -f > process_info
balamurugan@balamurugan-Lenovo-E41-25:~$ ls
bala.txt  Desktop    Downloads  Music          Pictures      Public   snap       test1  testa  Videos
demodir   Documents  error_log  personalfiles  process_info  results  Templates  test2  testb  week1.odt
balamurugan@balamurugan-Lenovo-E41-25:~$ cat process_info
UID          PID    PPID  C STIME TTY          TIME CMD
balamur+    6974    6956  0 19:41 pts/0    00:00:00 bash
balamur+    7470    6974  0 19:59 pts/0    00:00:00 ps -f
balamurugan@balamurugan-Lenovo-E41-25:~$
```