CS303 Operating System Lab-1

Submitted by : Parth Goyal
(2017csb1095)

Q1. Exploring Unix commands.

1) **pwd** : The pwd command prints the full system path of the current working directory.
    a) -L, --logical : use PWD from environment, even if it contains symlinks (To avoid symlinks pass the -P option)
    b) -P, --physical : avoid all symlinks
    SYNOPSIS : $ pwd [OPTION] ...
    Example :
    ```
    $ pwd
    /home/2017csb1095
    ```

    Windows Command : cd

2) **cd** : The cd command, also known as chdir(change directory), used to change the current working directory.
    a) $ cd [directory_name] : if directory is available in the current working directory, it makes this new directory as working directory.
    Examples :
    ```
    $ pwd
    /home/2017csb1095
    $ cd Desktop
    /home/2017csb1095/Desktop
    ```

    Windows Command : cd

3) **cp** : A command line utility for copying files and directories. It supports moving one or more files or folders with options for taking backups and preserving attributes.
    a) --backup : make a backup of each existing destination file
    b) --attributes-only : don't copy the file data, just the attributes.
    c) --parents : use full source file name under DIRECTORY
    d) --verbose : explain what is being done
    SYNOPSIS : cp [OPTION]... [-T] SOURCE DEST
    Examples :
    ```
    $ ls
    File1.txt        File2.txt
    $ cp --verbose File1.txt File2.txt
    'File1.txt' -> 'File2.txt'
    ```

    Windows Command : copy

4) **ls** : List directory contents, lists information about the FILES (the current directory by default). Sort entries alphabetically.
    a) -a, --all : do not ignore entries starting with .
    b) --author : with -l, print the author of each file
    c) -C : list entries by columns
    d) -d, --directory : list directories themselves, not their contents
    e) -l : use a long listing format
    f) -i : print the index number of each file

    SYNOPSIS : ls [OPTION]... [FILE]...

    Example :

```
$ ls
File1.txt       File2.txt
$ ls -l
total 8
56492468 -rw-r--r-- 1 2017csb1095 phdstudent 74 Aug 16 09:45 File1.txt
56492284 -rw-r--r-- 1 2017csb1095 phdstudent 74 Aug 16 09:47 File2.txt
```

    Windows Command : dir

5) **mkdir** : The mkdir command allows user to create directories or folders as they are referred to. It can create multiple directories at once and also set permissions when creating the directory.
    a) -m, --mode=MODE : set file mode
    b) -p, --parents : no error is existing, make parent directories as needed
    c) -v, --verbose : print a message for each created directory

    SYNOPSIS : mkdir [OPTION]... DIRECTORY...

    Example :

```
$ mkdir --verbose -p parent/daughter
mkdir: created directory 'parent'
mkdir: created directory 'parent/daughter'
```

    Windows Command : mkdir

6) **rm** : remove files or directories. It removes each specified file. By default, it does not remove directories.
    a) -f, --force : ignore nonexistent files and arguments
    b) -i : prompt before every removal

    SYNOPSIS : rm [OPTION]... [FILE]...

    Example :

```
$ ls
File1.txt       File2.txt       parent
$ rm --verbose -i File1.txt
removed 'File1.txt'
$ ls
File2.txt       parent
```

    Windows Command : del

7) **chmod** : change file mode bits

   a) -c, --changes : like verbose but report only when a change is made

   b) -f, --silent, --quiet : suppress most error messages

   c) --no-preserve-root : do not treat '/' specially

   SYNOPSIS : chmod [OPTION]... MODE[.. , MODE]... FILE…

   Example :

```
$ ls -l
-rw-r--r-- 1 2017csb1095 phdstudent   74 Aug 16 09:47 File2.txt
$ chmod u=rw File2.txt --verbose
mode of 'File2.txt' changed from 0444 (r--r--r--) to 0644 (rw-r—r--)
```

   Windows Command : attrib

8) **gzip** : It is a compressing tool used to truncate the file size. Use gunzip command to decompress a file.

   a) -a, --ascii : convert end-of-lines using local conventions.

   b) -d, --decompress

   c) -f, --force : force compression or decompression

   d) -k : keep the original file

   e) *SYNOPSIS : gzip [-acdfhklLnNrtvV19] [--rsyncable] [-S suffix] [name ...]*

   f) Example :

```
$ ls
File2.txt        parent
$ gzip -k File2.txt --verbose
File2.txt:     10.8% -- replaced with File2.txt.gz
$ ls
File2.txt  File2.txt.gz  parent
```

   Windows Command : (Not found)

9) **find** : search for files in a directory hierarchy

   a) -name : specify the name of the file/folder

   SYNOPSIS : find [options] [path] [expression]

   Example :

```
$ find ./ -name daughter
./Desktop/parent/daughter
./.local/share/Trash/files/parent/daughter
```

   Windows Command : find

10) **less** : less command doesn’t load entire file but loads it by part-by-part

a) –help

b) --version

SYNOPSIS : less [-[+]aABcCdeEfFgGiIJKLmMnNqQrRsSuUVwWX~]

```
[-b space] [-h lines] [-j line] [-k keyfile]
[-{oO} logfile] [-p pattern] [-P prompt] [-t tag]
[-T tagsfile] [-x tab,...] [-y lines] [-[z] lines]
[-# shift] [+[+]cmd] [--] [filename]...
```

<u>Windows Command : more</u>

11) **tail** : output the last part of files

a) -n, --lines : output the last NUM files, instead of last 10

b) -q, --quiet : never output headers giving file names

c) -c, --bytes : output the last NUM bytes

SYNOPSIS : tail [OPTION]... [FILE]...

Example :

```
$ tail File2.txt
  Hello, I’m Parth Goyal.
$ tail -c 18 File2.txt
  I’m Parth Goyal
```

<u>Windows Command : tail</u>

12) **top** : Display Linux Processes

a) -u : to specify user

SYNOPSIS : top -hv|-bcHiOSs -d secs -n max -u|U user -p pid -o fld -w [cols]

Example :

```
$ top
     1048 parthgo+  20   0  635740  71780  56912 S   2.0  0.9   0:40.91 Xorg
     1289 parthgo+  20   0 3749292 292356 105252 S   2.0  3.7   0:41.91 gnome-shell
     1822 parthgo+  20   0 1531080 204012 132508 S   1.0  2.6   0:42.69 soffice.bin
     1988 parthgo+  20   0  763236  38584  27412 S   1.0  0.5   0:07.79 gnome-term+
```

<u>Windows Command : mem</u>

13) **wc** : print newline, word, and byte counts for each file

a) -c : print the byte counts

b) -m : character count

c) -l : lines count

SYNOPSIS : wc [OPTION]... [FILE]...

Example :

```
$ wc File2.txt
     1  4 24 File2.txt
```

<u>Windows Command : fc</u>

14) **diff** : compare files line by line

a) –normal : output a normal diff

b) -q : report a brief detail

SYNOPSIS : diff [OPTION]... FILES

```
Example : $ diff -q File2.txt File1.txt
                    Files File2.txt and File1.txt differ
```

Windows Command : fc

15) **kill** : send a signal to a process

a) -l : list signal names

b) -L : list signal names in a nice table

SYNOPSIS : kill [options] <pid> [...]

```
Example : $ kill 1048
```

Windows Command : taskkill

16) **ifconfig** : interface configuration

SYNOPSIS : $ ifconfig

Windows Command : ipconfig

17) **shutdown** : Halt, power-off or reboot the machine

a) -r : reboot

b) -k : do not halt

c) -c : cancel pending shutdowns

SYNOPSIS : shutdown [OPTIONS...] [TIME] [WALL...]

```
Example : $ shutdown -r
```

Windows Command : shutdown

18) **which** : locate a command

a) -a : print all matching pathnames of each argument

SYNOPSIS : which [-a] filename ...

```
Example : $ which python
                    /home/2017csb1095/anaconda3/envs/envn/bin/python
```

Windows Command : where

19) **cat** : concatenate files and print on the standard output

a) -A : show all

b) -n : number all output lines

SYNOPSIS : cat [OPTION]... [FILE]...

Example :

```
$ cat File1.txt File2.txt
Hello, I'm Parth.
Hello, I'm Parth Goyal.
```

Windows Command : type

20) **mv** : move (rename) files

a) -b : make a backup

b) -n : do not overwrite an existing file

SYNOPSIS : mv [OPTION]... SOURCE... DIRECTORY

Example :

```
$ mv File1.txt File3.txt --verbose
renamed 'File1.txt' -> 'File3.txt'
```

Windows Command : (Not found)

21) **grep** : global regex print

a) -f : obtain patterns from FILE, one per line

b) -c : count of matching lines for each input file

SYNOPSIS : grep [OPTIONS] PATTERN [FILE...]

Example :

```
$ grep Parth File2.txt
Hello, I'm Parth Goyal.
```

Windows Command : find

22) **cut** : print selected parts of lines from each file to standard output

SYNOPSIS : cut [OPTION]... SOURCE... DIRECTORY

Example :

```
$ cut File2.txt -c 1
```

Windows Command : cut

23) **history** : GNU history library. It is able to keep track of lines read as input

a) -w : write history to a file

b) -c : clear the history

SYNOPSIS : history

Example : $ history

```
1993  cut File2.txt -c l
1994  cut File2.txt -c -l
1995  cut File2.txt -c
1996  cut --help
1997  cut File2.txt -c 1
1998  cut File2.txt -c 10
1999  cut File2.txt -c 10 --verbose
2000  cut File2.txt -c 10 -v
2001  kill -l
2002  man history
2003  history
```

Windows Command : F7

24) **echo** : Echo is used to output a string of characters to the stdout

a) -n : do not print the trailing new line

b) -e : enable interpretation of backslash escapes.

SYNOPSIS : echo [SHORT-OPTION]... [STRING]...

Example : $ echo OS LAB1

```
OS LAB1
```

Windows Command : echo

Q2. Explore vi Unix Editor.

Modes of Operation in vi editor : There are three modes of operation in vi.

- **Command Mode :** When vi starts up, it is in Command Mode. This mode is where vi interprets any characters we type as commands and thus does not display them in the window. This mode allows us to move through a file, and to delete, copy, or paste a piece of text. To enter into Command Mode from any other mode, it requires pressing the **[Esc]** key. If we press [Esc] when we are already in Command Mode, then vi will beep or flash the screen.
- **Insert Mode :** This mode enables you to insert text into the file. Everything that's typed in this mode is interpreted as input and finally, it is put in the file. The vi always starts in command mode. To enter text, you must be in insert mode. To come in insert mode you simply type i. To get out of insert mode, press the Esc key, which will put you back into command mode.

- **Last Line Mode (Escape Mode) :** Line Mode is invoked by typing a colon [:], while vi is in Command Mode. The cursor will jump to the last line of the screen and vi will wait for a command. This mode enables you to perform tasks such as saving files, executing commands.

Moving within a File (Navigation) :
To move around within a file without affecting text must be in command mode (press Esc twice). Here are some of the commands can be used to move around one character at a time.

Commands and their Description

- *k : Moves the cursor up one line.*
- *j : Moves the cursor down one line.*
- *h: Moves the cursor to the left one character position.*
- *l: Moves the cursor to the right one character position.*
- *0 or l: Positions cursor at beginning of line.*
- *$: Positions cursor at end of line.*
- *W: Positions cursor to the next word.*
- *B: Positions cursor to previous word.*
- *(: Positions cursor to beginning of current sentence.*
- *): Positions cursor to beginning of next sentence.*
- *H: Move to top of screen.*
- *nH: Moves to nth line from the top of the screen.*
- *M: Move to middle of screen.*
- *L: Move to bottom of screen.*
- *nL: Moves to nth line from the bottom of the screen.*
- *colon along with x: Colon followed by a number would position the cursor on line number represented by x.*

Editing and inserting in Files(Entering and Replacing Text):

To edit the file, we need to be in the insert mode. There are many ways to enter insert mode from the command mode.

- *I: Inserts text before current cursor location.*
- *I: Inserts text at beginning of current line.*
- *A: Inserts text after current cursor location.*

- *A: Inserts text at end of current line.*
- *O: Creates a new line for text entry below cursor location.*
- *O: Creates a new line for text entry above cursor location.*
- *R: Replace single character under the cursor with the next character typed.*
- *R: Replaces text from the cursor to right.*
- *S: Replaces single character under the cursor with any number of characters.*
- *S:Replaces entire line.*

Deleting Characters:

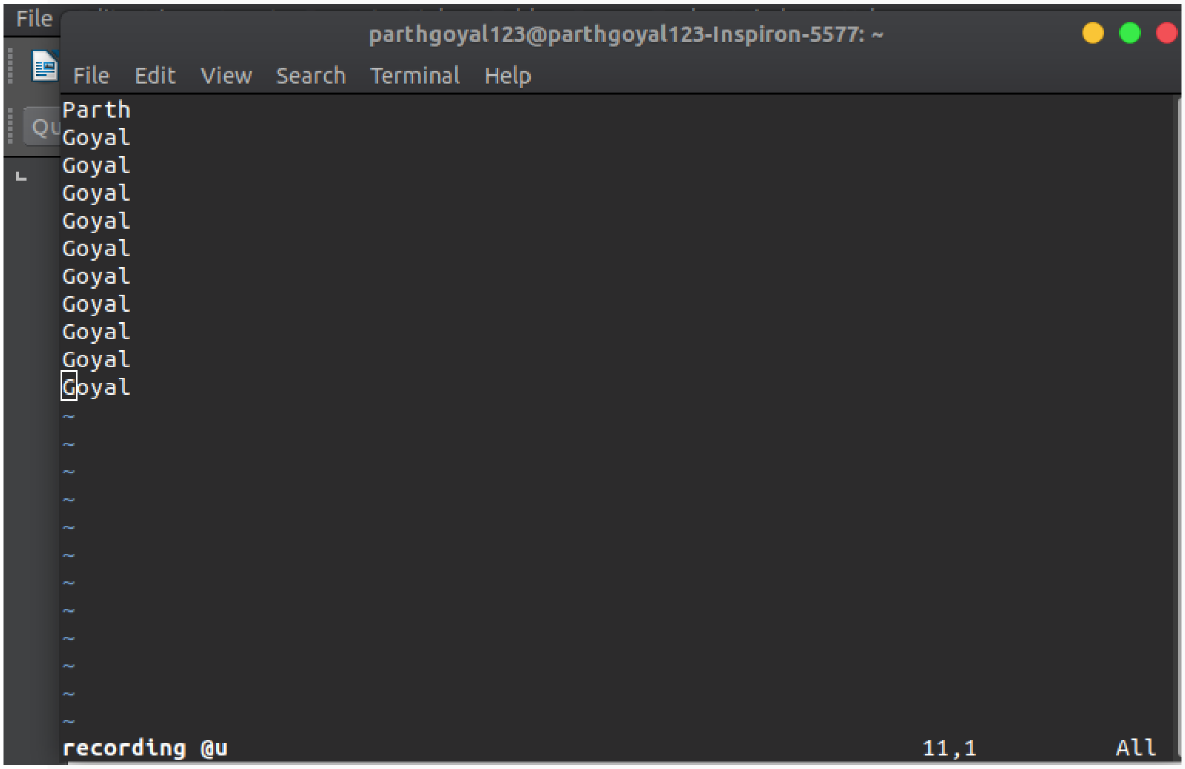
Here is the list of important commands which can be used to delete characters and lines in an opened file.

- *X: Deletes the character under the cursor location.*
- *X: Deletes the character before the cursor location.*
- *Dw: Deletes from the current cursor location to the next word.*
- *d^: Deletes from current cursor position to the beginning of the line.*
- *d$: Deletes from current cursor position to the end of the line.*
- *Dd: Deletes the line the cursor is on.*

Copy and Past Commands:

Copy lines or words from one place and paste them on another place by using the following commands.

- *Yy: Copies the current line.*
- *9yy: Yank current line and 9 lines below.*
- *p: Puts the copied text after the cursor.*
- *P: Puts the yanked text before the cursor.*


File
parthgoyal123@parthgoyal123-Inspiron-5577: ~
File Edit View Search Terminal Help

```
Parth
Goyal
Goyal
Goyal
Goyal
Goyal
Goyal
Goyal
Goyal
Goyal
Goyal
~
~
~
~
~
~
~
~
~
~
~
~
recording @u                                        11,1          All
```

Save and Exit Commands of the ex Mode :

Need to press [Esc] key followed by the colon (:) before typing the following commands:

- *q: Quit*
- *q!: Quit without saving changes i.e. discard changes.*
- *r fileName: Read data from file called fileName.*
- *wq: Write and quit (save and exit).*
- *w fileName: Write to file called fileName (save as).*
- *w! fileName: Overwrite to file called fileName (save as forcefully).*
- *!cmd: Runs shell commands and returns to Command mode.*

Q3. Exploring your System.

a) **Processor** : 8 Cores & Intel® Core™ i7-7700HQ CPU @ 2.80GHz × 8
b) **Capacity** : 1TB Hard disk. 128GB SSD. 8 GB RAM.
c) **Graphics** : Intel® HD Graphics 630 (Kaby Lake GT2)

d) Better Graphic Card or Increased RAM?

The **Graphic Card will make a better difference to the overall experience**. Graphics Card with 2GB is sufficient for all sorts of games. If you are going for more than two monitors then you might need to consider 4GB cards. Whether you are watching a movie, or playing a game or even moving a mouse pointer, everything is calculations in binary inside the CPU and GPU (**the GPU is responsible for rendering the display output**). So to store results of such huge amount of calculations, and to perform them, the CPU/GPU needs memory. That's what the RAM is- memory for storing and performing all the calculations. So wrapping up invest in a good Graphics Card and buy the fastest frequency RAM your motherboard can support. We hope, this has cleared a little bit of the query of ram vs graphics card.

Q4. Exploring the PROC command.

Command :

```
parthgoyal123@parthgoyal123-Inspiron-5577:~$ cat /proc/meminfo

MemTotal:        7932480 kB
MemFree:          274492 kB
MemAvailable:    2892388 kB
Buffers:          263424 kB
Cached:          3429488 kB
SwapCached:            0 kB
Active:          4667156 kB
Inactive:        2596216 kB
Active(anon):    3595660 kB
Inactive(anon):   888744 kB
Active(file):    1071496 kB
Inactive(file):  1707472 kB
Unevictable:          64 kB
Mlocked:              64 kB
SwapTotal:       2097148 kB
SwapFree:        2097148 kB
Dirty:              1296 kB
Writeback:             0 kB
AnonPages:       3570588 kB
Mapped:           835272 kB
Shmem:            926196 kB
Slab:             205384 kB
SReclaimable:     128612 kB
SUnreclaim:        76772 kB
KernelStack:       16224 kB
```

```
PageTables:        72928 kB
NFS_Unstable:          0 kB
Bounce:                0 kB
WritebackTmp:          0 kB
CommitLimit:     6063388 kB
Committed_AS:   12767512 kB
VmallocTotal:   34359738367 kB
VmallocUsed:           0 kB
VmallocChunk:          0 kB
HardwareCorrupted:     0 kB
AnonHugePages:         0 kB
ShmemHugePages:        0 kB
ShmemPmdMapped:        0 kB
CmaTotal:              0 kB
CmaFree:               0 kB
HugePages_Total:       0
HugePages_Free:        0
HugePages_Rsvd:        0
HugePages_Surp:        0
Hugepagesize:       2048 kB
DirectMap4k:      295776 kB
DirectMap2M:     7856128 kB
DirectMap1G:     1048576 kB
```

a)

```
MemTotal:        7932480 kB
MemFree:          274492 kB
```

About 3.46% is free

b)

```
parthgoyal123@parthgoyal123-Inspiron-5577:~$ ps
PID   TTY        TIME     CMD
9219  pts/1   00:00:00    bash
9399  pts/1   00:00:00    ps

parthgoyal123@parthgoyal123-Inspiron-5577:~$ grep ctxt /proc/9219/status
voluntary_ctxt_switches:    420
nonvoluntary_ctxt_switches:         3

parthgoyal123@parthgoyal123-Inspiron-5577:~$ grep ctxt /proc/9399/status
voluntary_ctxt_switches:    4
nonvoluntary_ctxt_switches:         0
```

Total voluntary switches : 426

```
Total involuntary switches : 3
```

c)

```
parthgoyal123@parthgoyal123-Inspiron-5577:~$ vmstat -f
9499 forks
```