Experiment 1:
Aim: Practice on basic Linux commands.

Basic Linux Commands

pwd

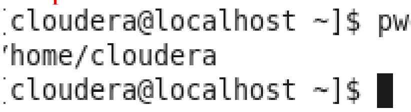
pwd (print working directory) – The pwd command is used to display the name of the current working directory in the Linux system using the terminal.

syntax

pwd [-LP]

Sr.No.	Option & Description
1	**-L (logical)**Display the value of $pwd if it names the current working directory
2	**-P (physical)**Display the physical directory, without any soft link
3	**--help**Displays a help message and then exits.

Output:



```
cloudera@localhost ~]$ pwd
home/cloudera
cloudera@localhost ~]$ █
```

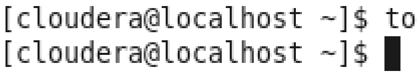
Create File using touch

touch command: It is used to create a file without any content. The file created using touch command is empty. This command can be used when the user doesn't have data to store at the time of file creation.

Syntax:

```
touch file_name
```

output



```
[cloudera@localhost ~]$ touch sample
[cloudera@localhost ~]$ █
```

To Add Content Existing File using vi

The vi editor tool is an interactive tool as it displays changes made in the file on the screen while you edit the file.

In vi editor you can insert, edit or remove a word as cursor moves throughout the file.

Commands are specified for each function like to delete it's x or dd.

The vi editor is case-sensitive. For example, **p** allows you to paste after the current line while **P** allows you to paste before the current line.

vi syntax:

vi **<fileName>**

Command mode

This is what you'll see when you'll press enter after the above command. If you'll start typing, nothing will appear as you are in command mode. By default vi opens in command mode.

Insert mode

To move to the insert mode press **i.** Although, there are other commands also to move to insert mode which we'll study in next page.

Look at the above snapshot, after pressing **i** we have entered into insert mode. Now we can write anything. To move to the next line press enter.

Once you have done with your typing, press **esc** key to return to the command mode.

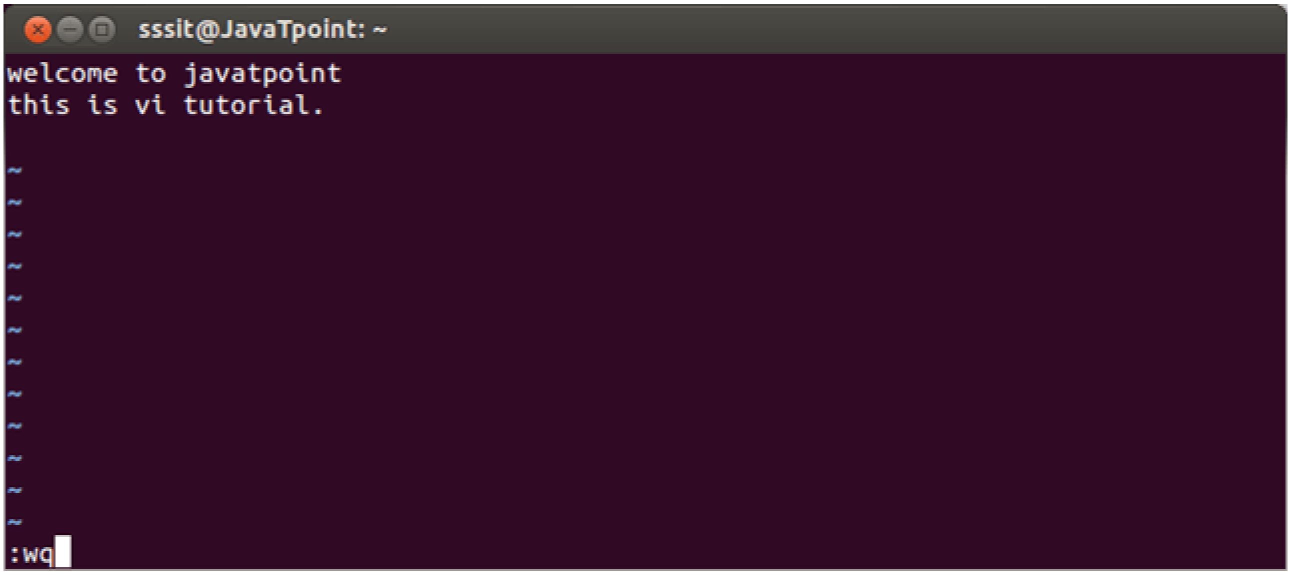
To save and quit

You can save and quit vi editor from command mode. Before writing save or quit command you have to press colon **(:).** Colon allows you to give instructions to vi.

exit vi table:

Commands	Action
:wq	Save and quit
:w	Save
:q	Quit
:w fname	Save as fname
ZZ	Save and quit
:q!	Quit discarding changes made
:w!	Save (and write to non-writable file)

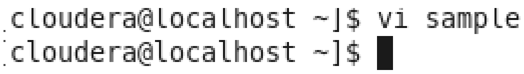
Type **:wq** to save and exit the file.



```
welcome to javatpoint
this is vi tutorial.

~
~
~
~
~
~
~
~
~
~
~
~
:wq
```

Look at the above snapshot, command :wq will save and quit the vi editor. When you'll type it in command mode, it will automatically come at bottom left corner.

Output



```
cloudera@localhost ~]$ vi sample
cloudera@localhost ~]$ █
```

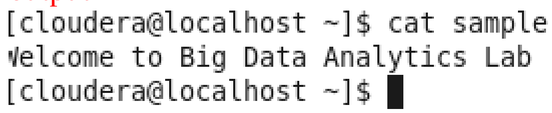
To see Content of file using cat command

Cat(concatenate) command is very frequently used in Linux. It reads data from the file and gives its content as output. It helps us to create, view, and concatenate files. So let us see some frequently used cat commands.

Syntax

```
cat file_name
```

output



```
[cloudera@localhost ~]$ cat sample
Welcome to Big Data Analytics Lab
[cloudera@localhost ~]$ 
```

To see List of file using ls command

In Linux, the command **"ls"** is one of the most commonly used. It's used to display a list of files and sub-directories in the current directory. If you're new to using the command line, the first command you should learn is probably **ls**.This command can be used by both regular users as well as system administrators.

syntax:

```
ls [ Options ] [File]
```

Options   Description

ls –a   list all files including hidden file starting with '.'.

ls –d   list directories - with ' */'.

ls –l   list with long format - show permissions.

ls –F   Append indicator (one of */=>@|) to entries.

ls –lh   This command will show you the file sizes in human readable format.

ls –r   list in reverse order.ls -ilist file's inode(index) number.

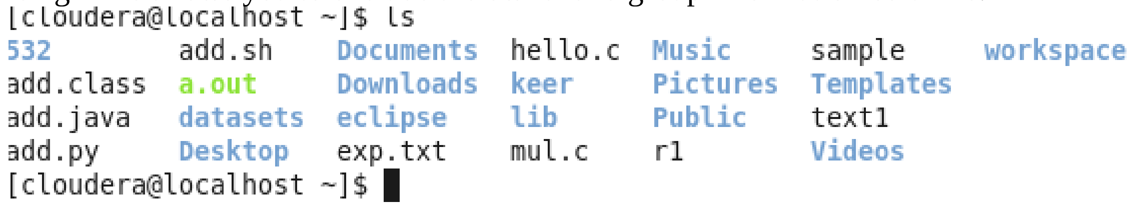
ls –ltr View Reverse Output Order by Date.

ls –t   sort by time & date

.ls –n   It is used to print group ID and owner ID instead of their names.

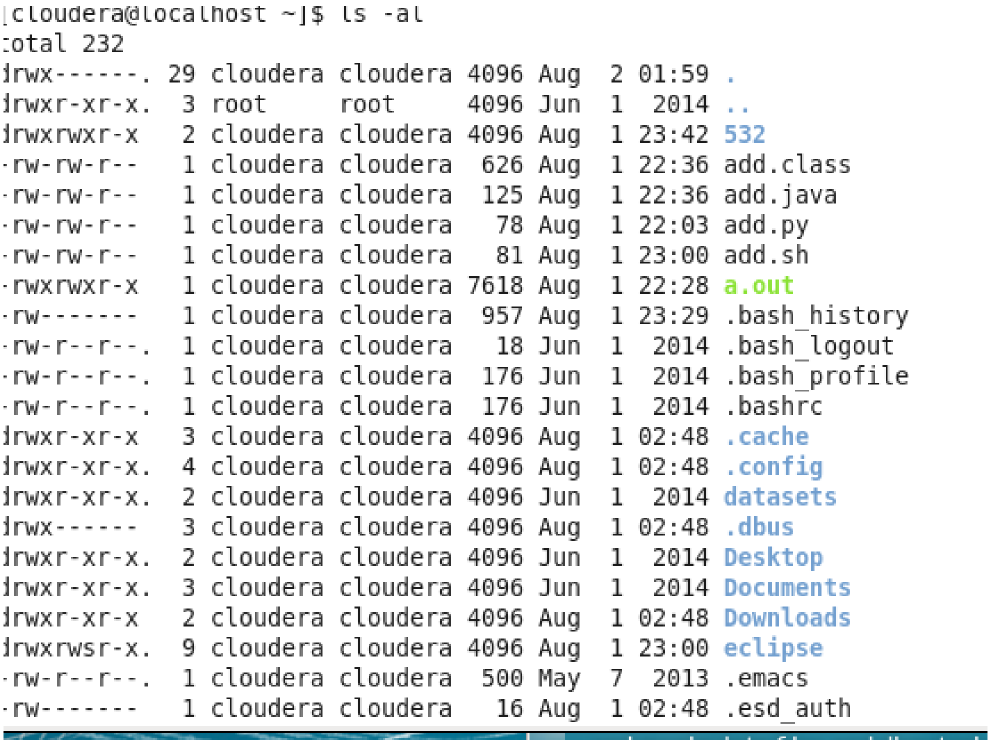
ls –m A list of entries separated by commas should fill the width.

ls –g This allows you to exclude the owner and group information columns.



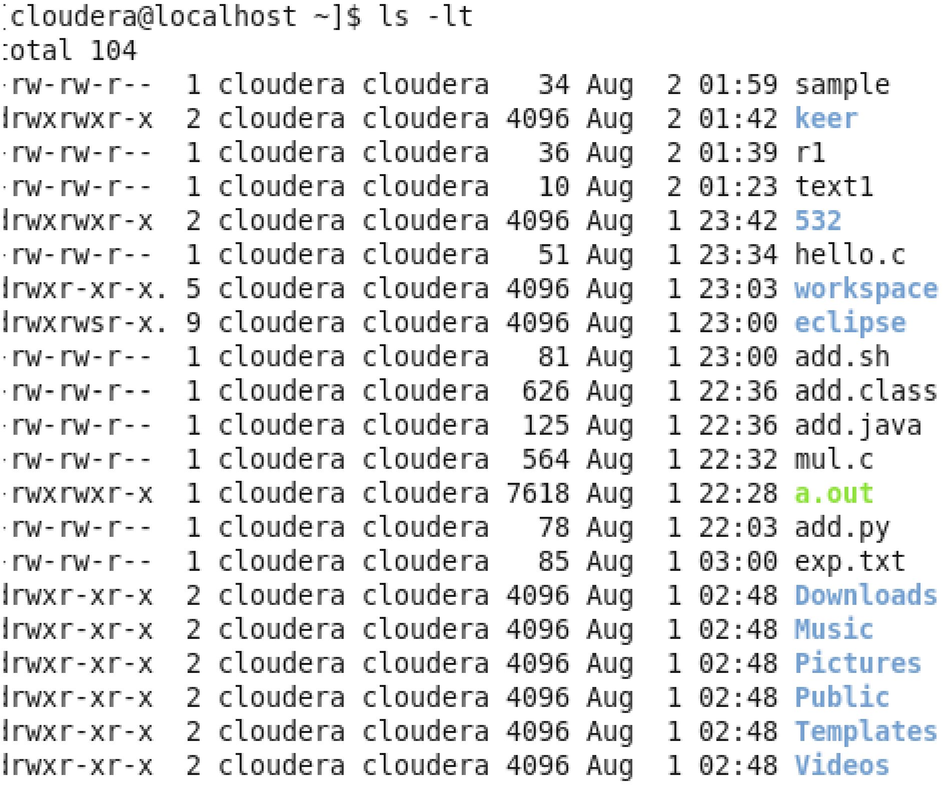
```
[cloudera@localhost ~]$ ls
532         add.sh    Documents  hello.c  Music     sample     workspace
add.class   a.out     Downloads  keer     Pictures  Templates
add.java    datasets  eclipse    lib      Public    text1
add.py      Desktop   exp.txt    mul.c    r1        Videos
[cloudera@localhost ~]$ 
```

To see hidden Files



```
[cloudera@localhost ~]$ ls -al
total 232
drwx------. 29 cloudera cloudera 4096 Aug  2 01:59 .
drwxr-xr-x.  3 root     root     4096 Jun  1  2014 ..
drwxrwxr-x   2 cloudera cloudera 4096 Aug  1 23:42 532
-rw-rw-r--   1 cloudera cloudera  626 Aug  1 22:36 add.class
-rw-rw-r--   1 cloudera cloudera  125 Aug  1 22:36 add.java
-rw-rw-r--   1 cloudera cloudera   78 Aug  1 22:03 add.py
-rw-rw-r--   1 cloudera cloudera   81 Aug  1 23:00 add.sh
-rwxrwxr-x   1 cloudera cloudera 7618 Aug  1 22:28 a.out
-rw-------   1 cloudera cloudera  957 Aug  1 23:29 .bash_history
-rw-r--r--.  1 cloudera cloudera   18 Jun  1  2014 .bash_logout
-rw-r--r--.  1 cloudera cloudera  176 Jun  1  2014 .bash_profile
-rw-r--r--.  1 cloudera cloudera  176 Jun  1  2014 .bashrc
drwxr-xr-x   3 cloudera cloudera 4096 Aug  1 02:48 .cache
drwxr-xr-x.  4 cloudera cloudera 4096 Aug  1 02:48 .config
drwxr-xr-x.  2 cloudera cloudera 4096 Jun  1  2014 datasets
drwx------   3 cloudera cloudera 4096 Aug  1 02:48 .dbus
drwxr-xr-x.  2 cloudera cloudera 4096 Jun  1  2014 Desktop
drwxr-xr-x.  3 cloudera cloudera 4096 Jun  1  2014 Documents
drwxr-xr-x   2 cloudera cloudera 4096 Aug  1 02:48 Downloads
drwxrwsr-x.  9 cloudera cloudera 4096 Aug  1 23:00 eclipse
-rw-r--r--.  1 cloudera cloudera  500 May  7  2013 .emacs
-rw-------   1 cloudera cloudera   16 Aug  1 02:48 .esd_auth
```

To see sorting of files



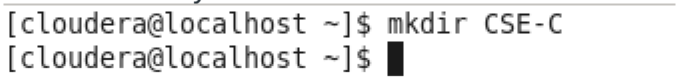
```
[cloudera@localhost ~]$ ls -lt
total 104
-rw-rw-r--  1 cloudera cloudera   34 Aug  2 01:59 sample
drwxrwxr-x  2 cloudera cloudera 4096 Aug  2 01:42 keer
-rw-rw-r--  1 cloudera cloudera   36 Aug  2 01:39 r1
-rw-rw-r--  1 cloudera cloudera   10 Aug  2 01:23 text1
drwxrwxr-x  2 cloudera cloudera 4096 Aug  1 23:42 532
-rw-rw-r--  1 cloudera cloudera   51 Aug  1 23:34 hello.c
drwxr-xr-x. 5 cloudera cloudera 4096 Aug  1 23:03 workspace
drwxrwsr-x. 9 cloudera cloudera 4096 Aug  1 23:00 eclipse
-rw-rw-r--  1 cloudera cloudera   81 Aug  1 23:00 add.sh
-rw-rw-r--  1 cloudera cloudera  626 Aug  1 22:36 add.class
-rw-rw-r--  1 cloudera cloudera  125 Aug  1 22:36 add.java
-rw-rw-r--  1 cloudera cloudera  564 Aug  1 22:32 mul.c
-rwxrwxr-x  1 cloudera cloudera 7618 Aug  1 22:28 a.out
-rw-rw-r--  1 cloudera cloudera   78 Aug  1 22:03 add.py
-rw-rw-r--  1 cloudera cloudera   85 Aug  1 03:00 exp.txt
drwxr-xr-x  2 cloudera cloudera 4096 Aug  1 02:48 Downloads
drwxr-xr-x  2 cloudera cloudera 4096 Aug  1 02:48 Music
drwxr-xr-x  2 cloudera cloudera 4096 Aug  1 02:48 Pictures
drwxr-xr-x  2 cloudera cloudera 4096 Aug  1 02:48 Public
drwxr-xr-x  2 cloudera cloudera 4096 Aug  1 02:48 Templates
drwxr-xr-x  2 cloudera cloudera 4096 Aug  1 02:48 Videos
```

Create Directory

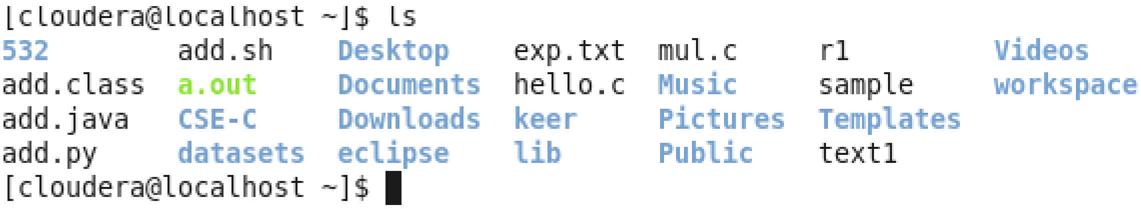
mkdir command in Linux allows the user to create directories (also referred to as folders in some operating systems). This command can create multiple directories at once as well as set the permissions for the directories. It is important to note that the user executing this command must have enough permission to create a directory in the parent directory, or he/she may receive a ‘permission denied’ error.

Syntax

mkdir directory-name



```
[cloudera@localhost ~]$ mkdir CSE-C
[cloudera@localhost ~]$ █
```



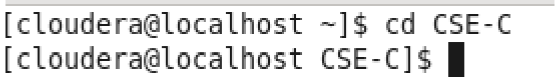
```
[cloudera@localhost ~]$ ls
532        add.sh     Desktop    exp.txt  mul.c     r1         Videos
add.class  a.out      Documents  hello.c  Music     sample     workspace
add.java   CSE-C      Downloads  keer     Pictures  Templates
add.py     datasets   eclipse    lib      Public    text1
[cloudera@localhost ~]$ █
```

Change Directory

cd command in Linux known as the change directory command. It is used to move efficiently from the current working directory to different directories in our System.

Syntax

cd directory-name



```
[cloudera@localhost ~]$ cd CSE-C
[cloudera@localhost CSE-C]$ █
```

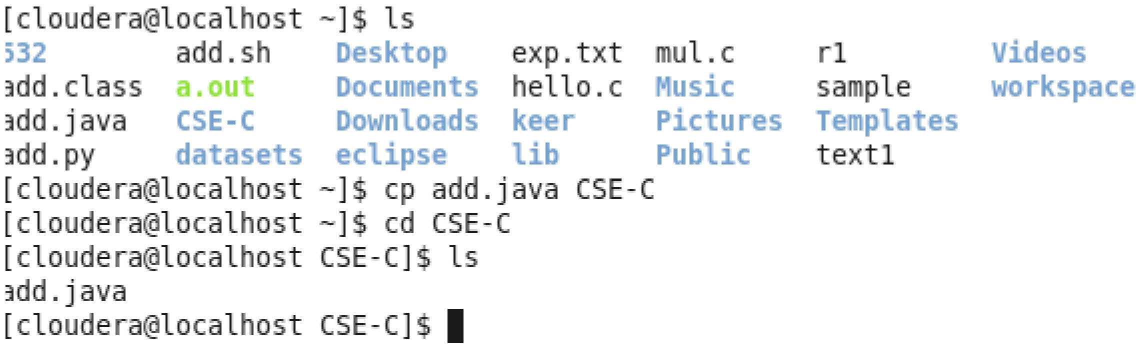
Copy file

cp stands for a **copy**. This command is used to copy files or groups of files or directories. It creates an exact image of a file on a disk with a different file name. *cp* command requires at least two filenames in its arguments.

Syntax:

```
cp [OPTION] Source Destination
cp [OPTION] Source Directory
cp [OPTION] Source-1 Source-2 Source-3 Source-n Directory
```

output:



```
[cloudera@localhost ~]$ ls
532        add.sh     Desktop    exp.txt  mul.c     r1         Videos
add.class  a.out      Documents  hello.c  Music     sample     workspace
add.java   CSE-C      Downloads  keer     Pictures  Templates
add.py     datasets   eclipse    lib      Public    text1
[cloudera@localhost ~]$ cp add.java CSE-C
[cloudera@localhost ~]$ cd CSE-C
[cloudera@localhost CSE-C]$ ls
add.java
[cloudera@localhost CSE-C]$ █
```

Move file

In UNIX-based operating systems like Linux and macOS, `**mv**` stands for “move”. But in this article, we will be talking about the “mv command in Linux”. As its name suggests this command is used to rename file directories and move files from one location to another within a file system.

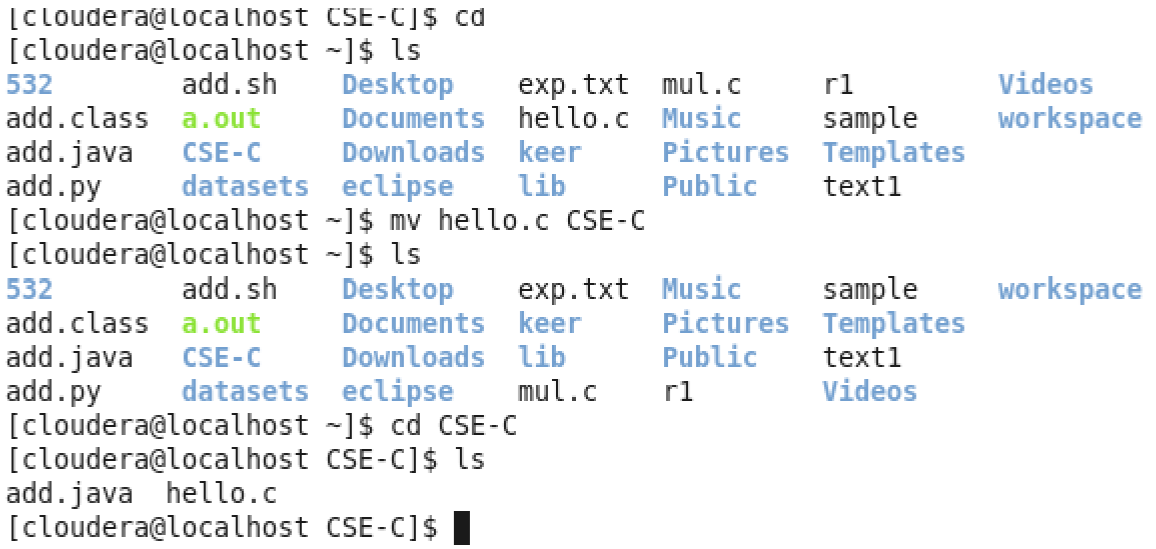
Two Distinct Functions of `mv` Command

- 1) Renaming a file or directory.
- 2) Moving a file or directory to another location

Syntax

mv [options(s)] [source_file_name(s)] [Destination_file_name]

output:



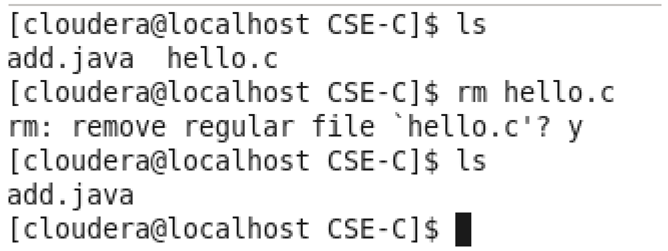
```
[cloudera@localhost CSE-C]$ cd
[cloudera@localhost ~]$ ls
532        add.sh    Desktop    exp.txt  mul.c     r1         Videos
add.class  a.out     Documents  hello.c  Music     sample     workspace
add.java   CSE-C     Downloads  keer     Pictures  Templates
add.py     datasets  eclipse    lib      Public    text1
[cloudera@localhost ~]$ mv hello.c CSE-C
[cloudera@localhost ~]$ ls
532        add.sh    Desktop    exp.txt  Music     sample     workspace
add.class  a.out     Documents  keer     Pictures  Templates
add.java   CSE-C     Downloads  lib      Public    text1
add.py     datasets  eclipse    mul.c    r1        Videos
[cloudera@localhost ~]$ cd CSE-C
[cloudera@localhost CSE-C]$ ls
add.java  hello.c
[cloudera@localhost CSE-C]$ █
```

Remove file

rm stands for **remove** here. rm command is used to remove objects such as files, directories, symbolic links and so on from the file system like UNIX.

Syntax

`rm [OPTION]... FILE...`



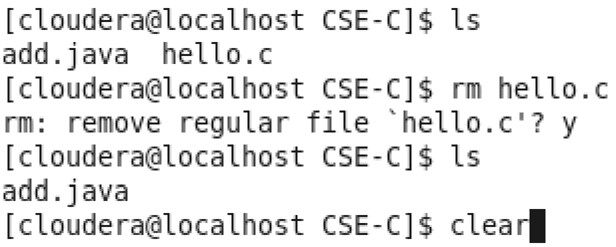
```
[cloudera@localhost CSE-C]$ ls
add.java  hello.c
[cloudera@localhost CSE-C]$ rm hello.c
rm: remove regular file `hello.c'? y
[cloudera@localhost CSE-C]$ ls
add.java
[cloudera@localhost CSE-C]$ █
```

Clear the Screen

clear is a standard Unix computer operating system command that is used to clear the terminal screen.

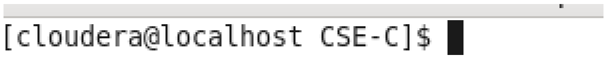
Syntax

Clear



```
[cloudera@localhost CSE-C]$ ls
add.java  hello.c
[cloudera@localhost CSE-C]$ rm hello.c
rm: remove regular file `hello.c'? y
[cloudera@localhost CSE-C]$ ls
add.java
[cloudera@localhost CSE-C]$ clear
```

After



```
[cloudera@localhost CSE-C]$
```

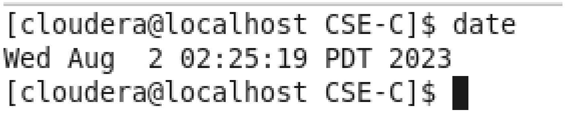
System Info-commands

Date

date command is used to display the system date and time. date command is also used to set date and time of the system.

Syntax:

```
date [OPTION]... [+FORMAT]

date [-u|--utc|--universal] [MMDDhhmm[[CC]YY][.ss]]
```

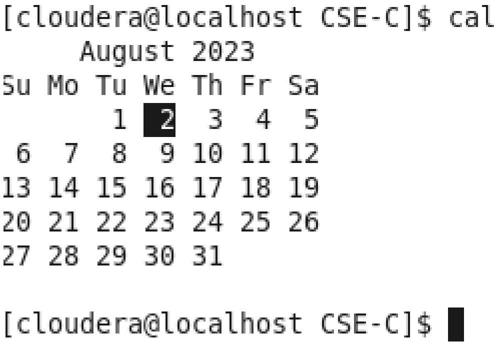


```
[cloudera@localhost CSE-C]$ date
Wed Aug  2 02:25:19 PDT 2023
[cloudera@localhost CSE-C]$
```

cal command is a calendar command in Linux which is used to see the calendar of a specific month or a whole year.

Syntax:

```
cal [ [ month ] year]
```



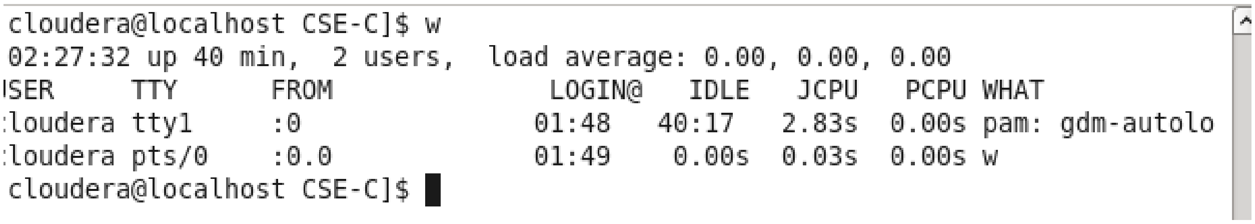
```
[cloudera@localhost CSE-C]$ cal
     August 2023
Su Mo Tu We Th Fr Sa
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31

[cloudera@localhost CSE-C]$
```

W command

The **'w'** command in Linux gives us important information about who is currently using the computer, how much the computer is being used, and what programs are running. It's a handy tool for people who take care of computer systems, as it helps them keep an eye on what users are doing,

Syntax of `w` command in Linux

```
w [options] user [...]
```



```
cloudera@localhost CSE-C]$ w
02:27:32 up 40 min,  2 users,  load average: 0.00, 0.00, 0.00
ISER     TTY      FROM              LOGIN@   IDLE   JCPU   PCPU WHAT
:loudera tty1     :0                01:48   40:17   2.83s  0.00s pam: gdm-autolo
:loudera pts/0    :0.0              01:49    0.00s  0.03s  0.00s w
cloudera@localhost CSE-C]$ █
```

Whoami

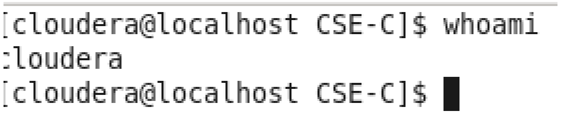
whoami command is used both in *Unix Operating System* and as well as in *Windows Operating System*.

- It is basically the concatenation of the strings **“who”,”am”,”i”** as **whoami**.
- It displays the username of the current user when this command is invoked.
- It is similar as running the id command with the options **-un**.

The earliest versions were created in 2.9 BSD as a convenience form for who am i, the Berkeley Unix who command’s way of printing just the logged in user’s identity. The GNU version was written by Richard Mlynarik and is part of the GNU Core Utilities (coreutils).

Syntax:

```
geekforgeeks@HP~: whoami
```



```
[cloudera@localhost CSE-C]$ whoami
:loudera
[cloudera@localhost CSE-C]$ █
```