**Session1 Assignment2:**

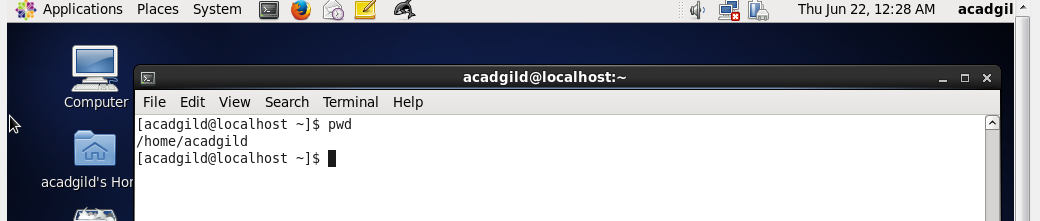
Explain the below linux commands with an example. Share the screenshot of each command with the output:

1. **Pwd:**

* pwd is "present working directory"
* pwd linux command prints the name of the current working directory
* pwd prints the full pathname of the current working directory

Synatx: pwd [OPTION]…

Example: pwd --- print the name of the current working directory



1. **Vi:**

It is a visual Text editor used to edit the file…

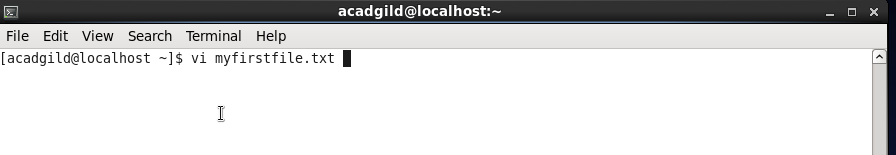
Synatx: vi [OPTION] [filename]….

Example:

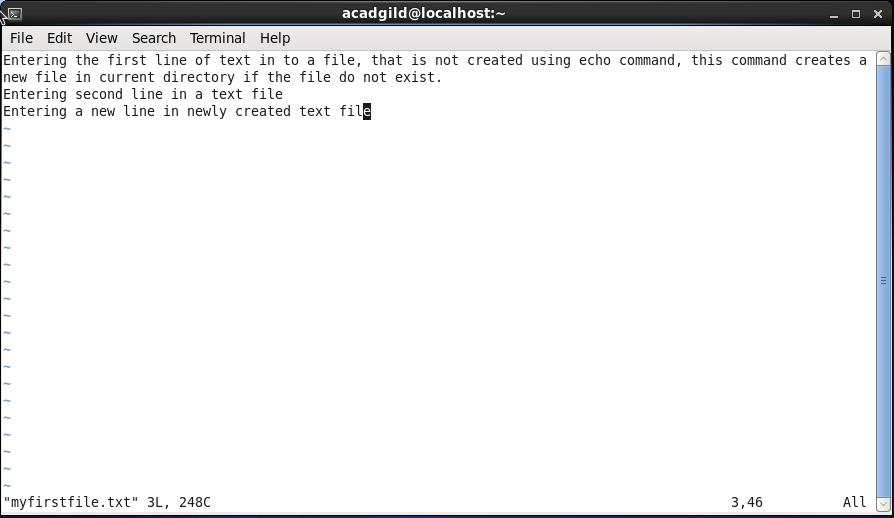
vi filename ---- edits file starting at line1

vi -r filename ---- recover filename that was being edited when system crashed

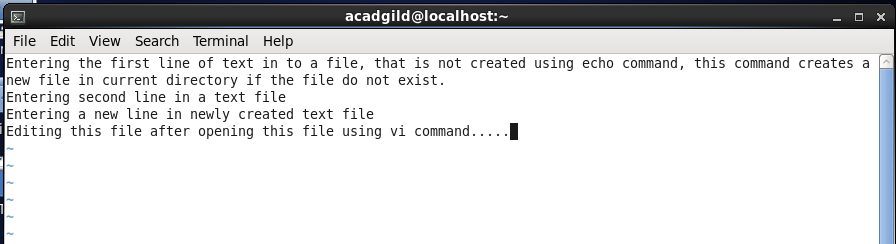
:q <Return> --- quits / exits vi



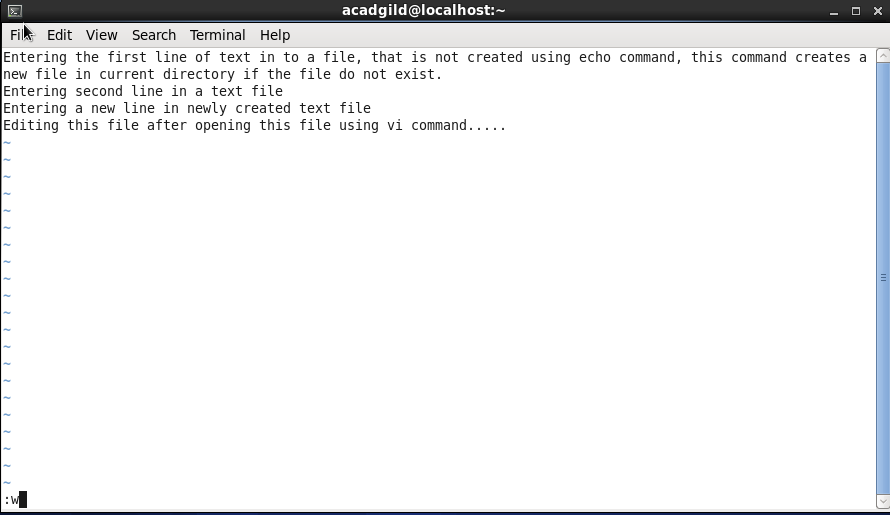
Opened file using **vi** linux command



Editing “myfirstfile.txt” file and saving the same…



Type **:w** to save the text / line typed in a file…



Again open the file using vi command to ensure you have saved recent typed text in a file…



1. **touch:**

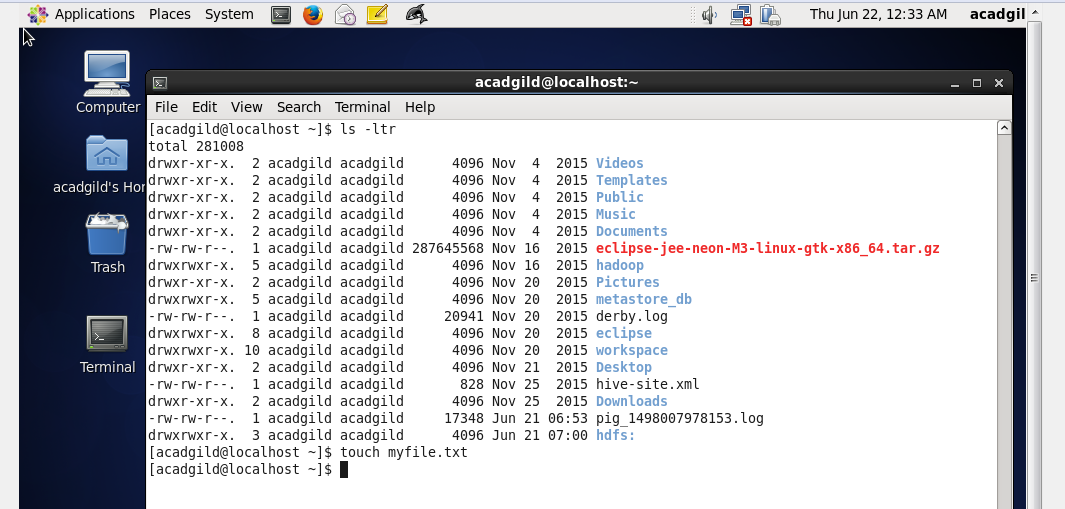
* touch linux command changes file timestamps
* touch linux command updates the access and modification times of each FILE to the current system time

Syntax: touch [OPTION]... FILE...

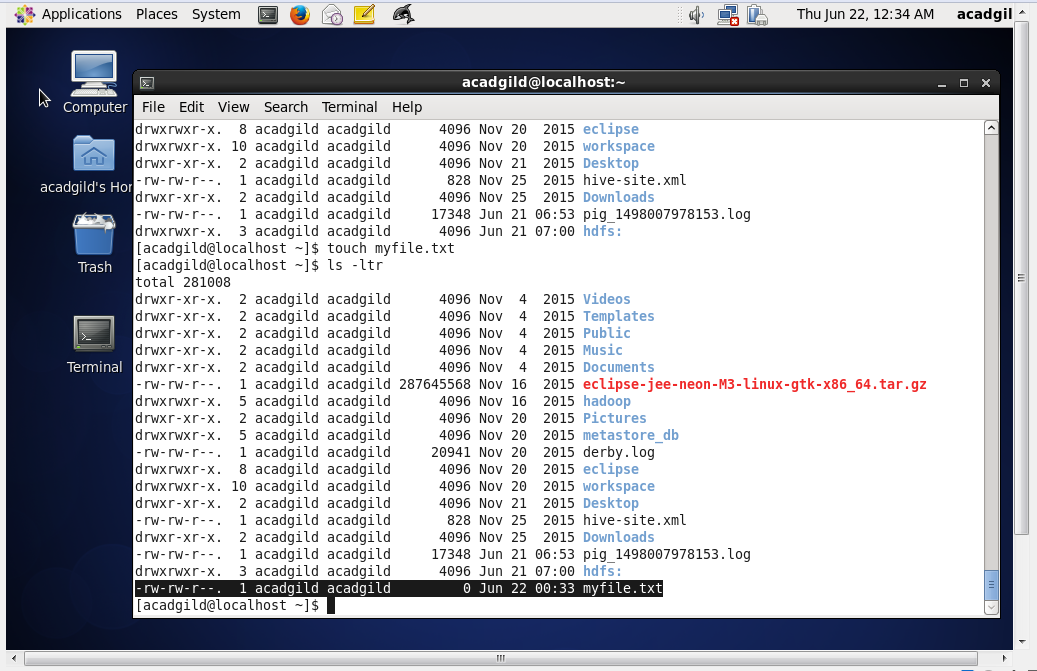
Example:

touch file.txt --- if file.txt exists, touch updates its access and modification times to the current time. If file.txt does not exist, it is created as new, empty file.

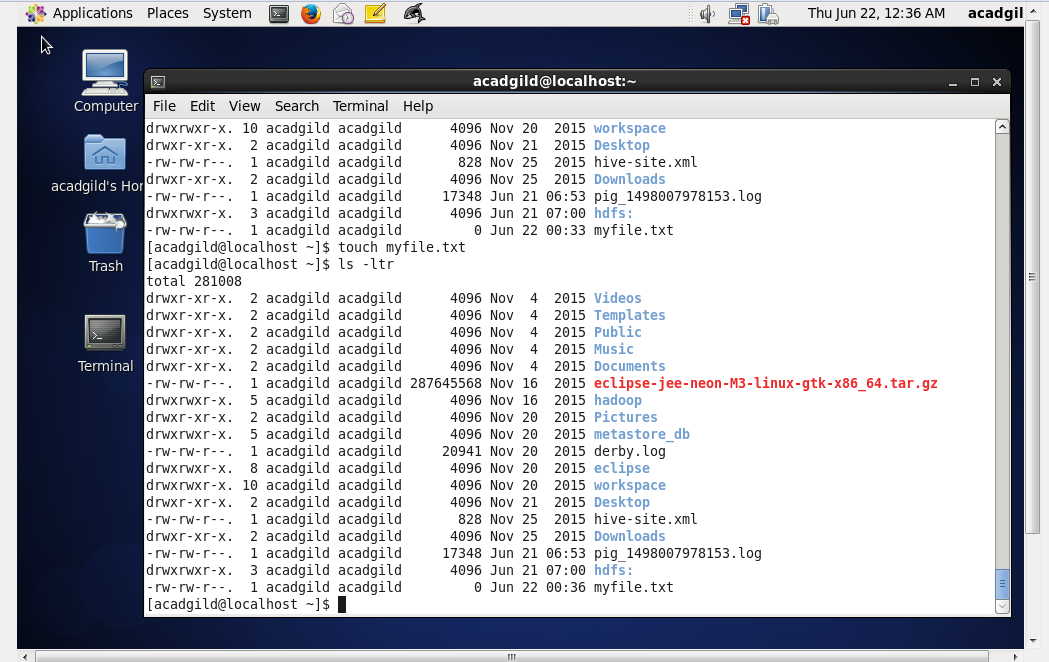
Creating “myfile.txt” file using touch command:



“myfile.txt” empty file created using touch command:



Modified creation time of file to current time using touch command:



1. **mkdir:**

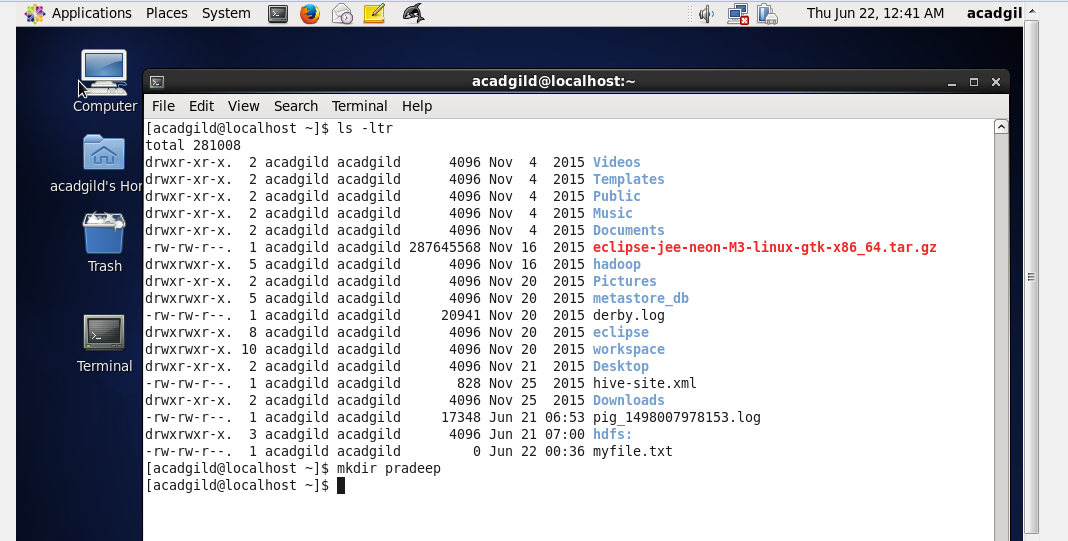
* mkdir is known as "make directory", mkdir is used to create directories on a file system.
* If the specified directory does not already exist, mkdir creates it. More than one directory may be specified when calling mkdir.

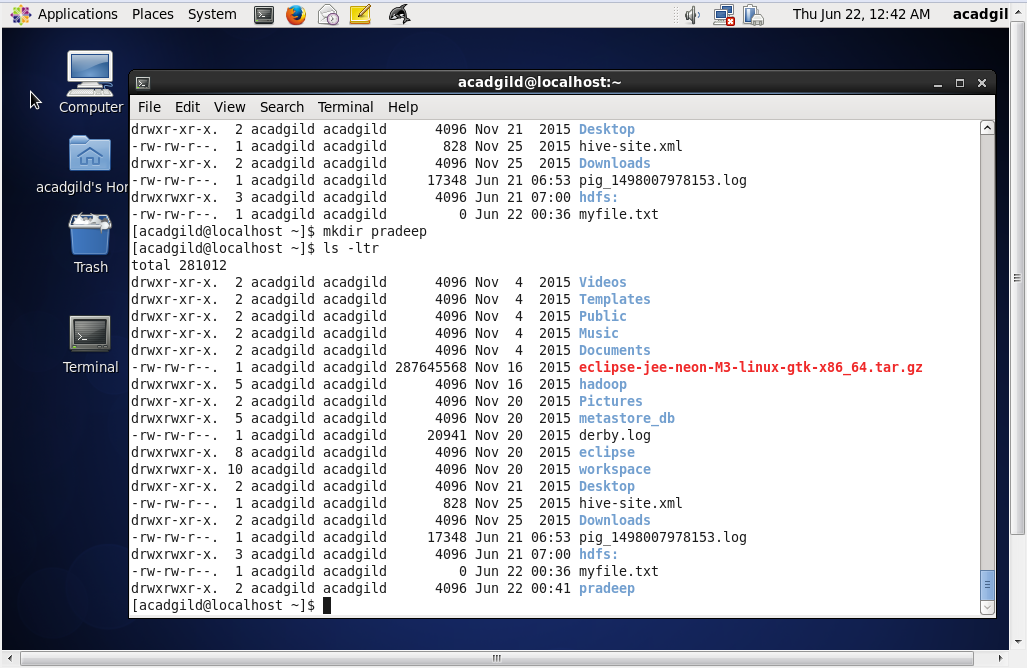
Syntax: mkdir[OPTION]...DIRECTORY...

Example:

mkdir mydirectory ----> this creates as directory named "mydirectory" on a file system.

Creating a new directory on file system:





1. **rm:**

* rm linux command removes or deletes files / directories.
* rm command removes each specified FILE.
* removal process unlinks a filename in a file system from data on the storage device.

Syntax: rm [OPTION]... FILE...

Examples:

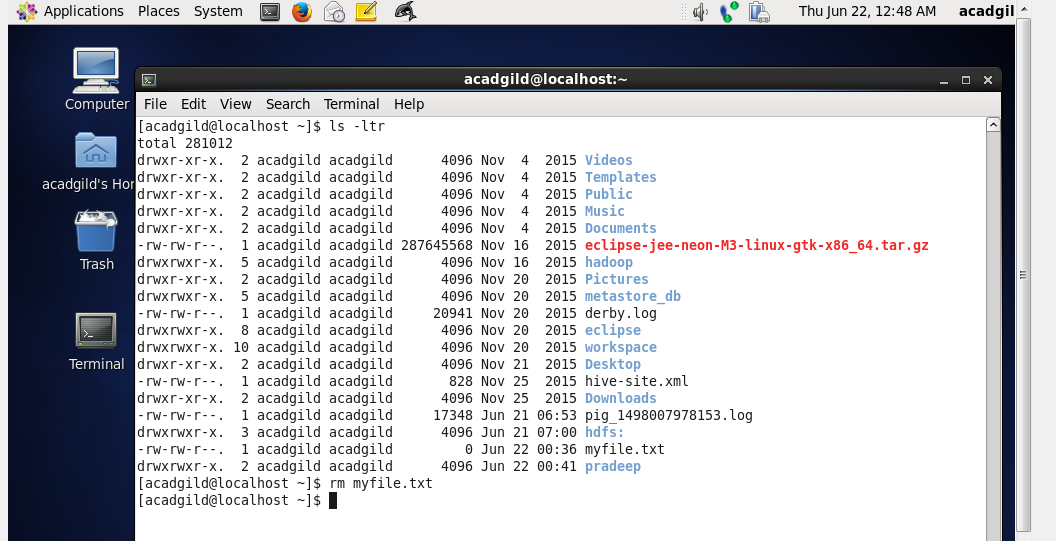
rm myfile.txt ---- removes the file "myfile.txt" , if the file is write-protected, then we will be prompted to confirm if the file has to really deleted.

rm -f myfile.txt ---- removes the file "myfile.txt" , even though the file is write protected.

rm\* ---- removes all files present in working directory

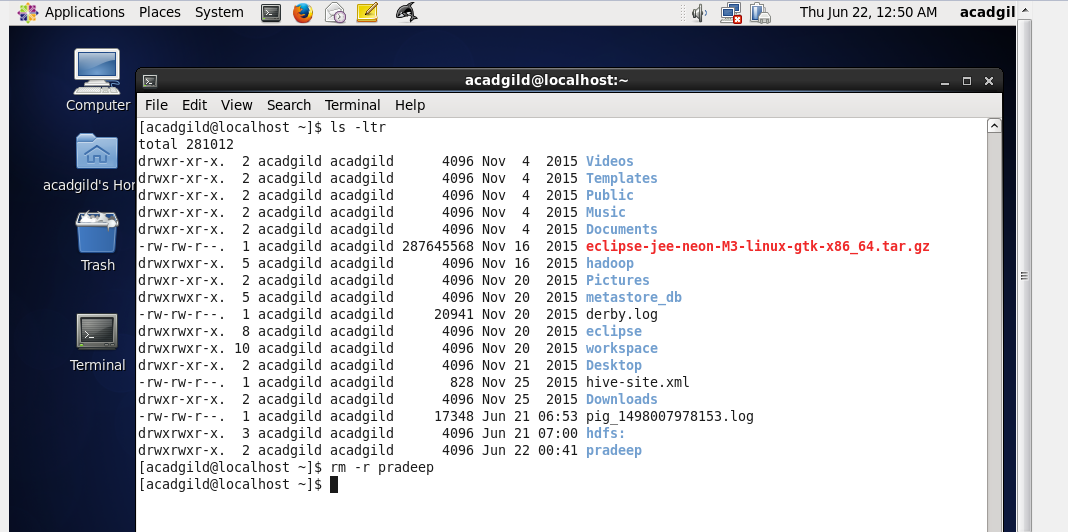
rm -r mydirectory ---- removes the directory named "mydirectory"

Removing file “myfile.txt” using rm command:

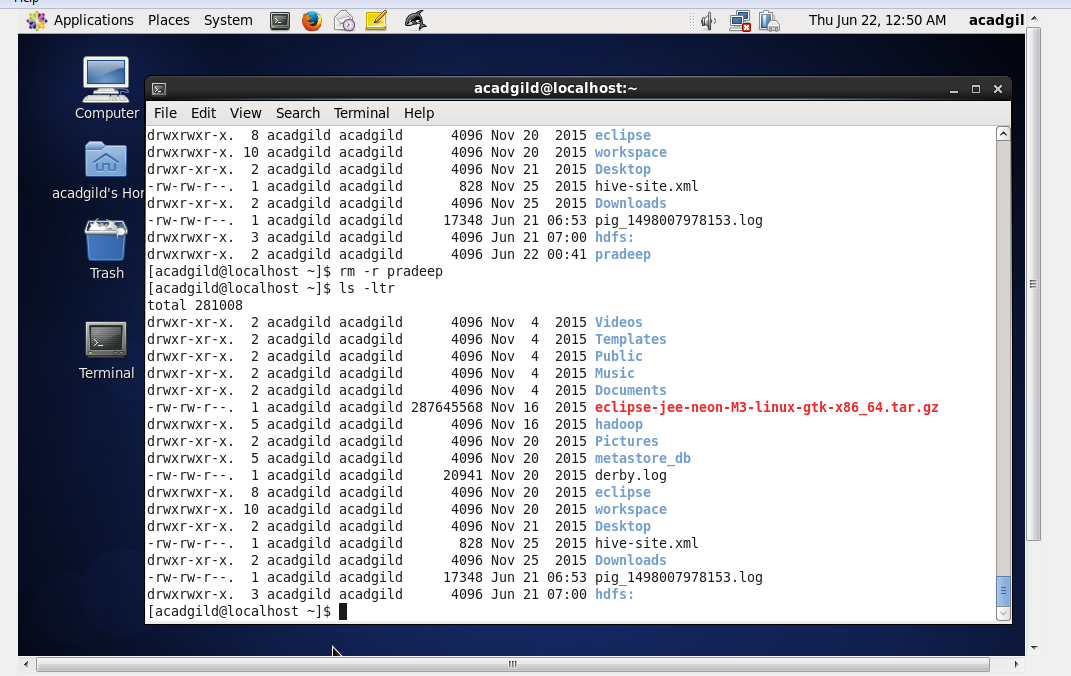




Removing directory “pradeep” using rm command:



directory “pradeep” does not exist:



1. **ls:**

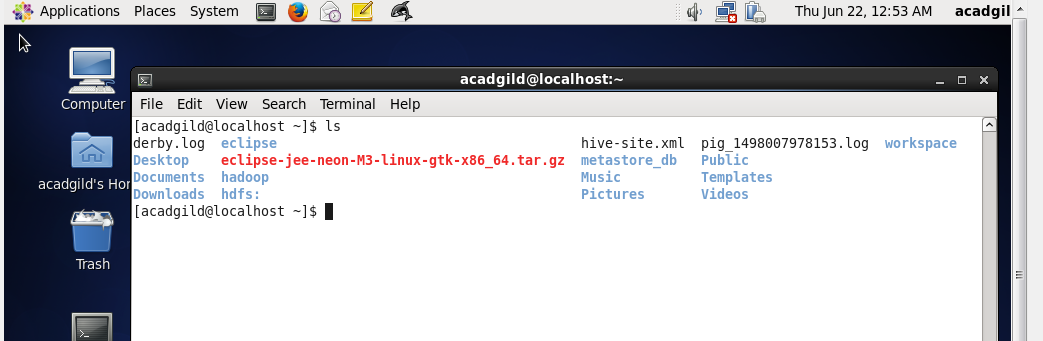
* ls linux command lists the contents of a directory.
* lists the information about the files that are in current directory, sort entries alphabetically if nothing specified.

Syntax: ls[OPTION]...[FILE]...

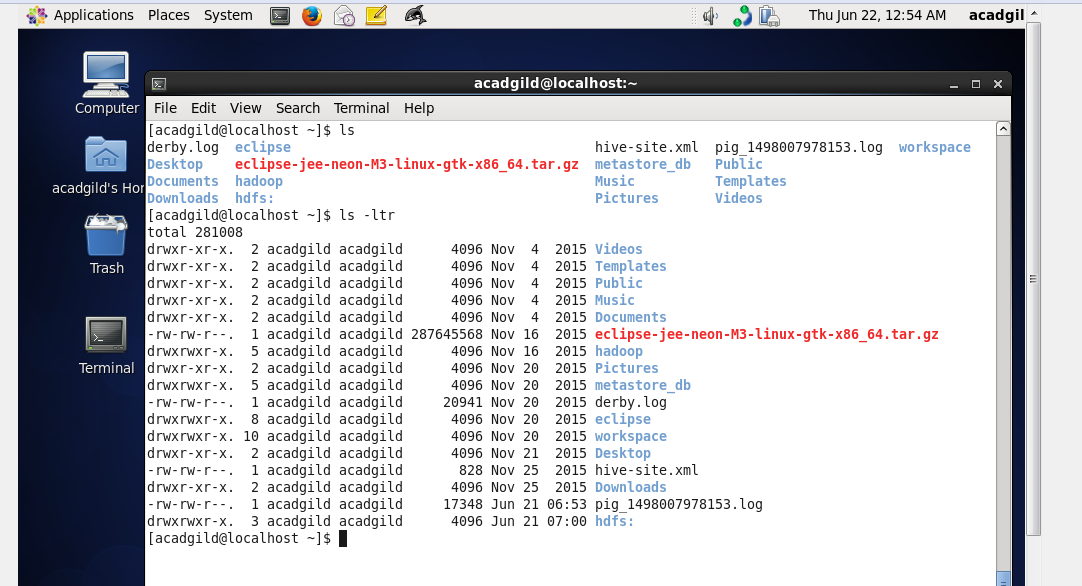
Examples:

ls -l ------ Lists the total files in the directory and subdirectories, names of the files in current directory, their permissions, file size and last modification date.

ls -ltr ----- list files sorted by the time they were last modified - most recently modified files will be last.



Ls –ltr command lists the contents of directory or lists the files sorted (most recently modified will be at last.



1. **echo:**

* echo displays a line of text
* echo is a fundamental command found in most of OS's, that offer a command line.

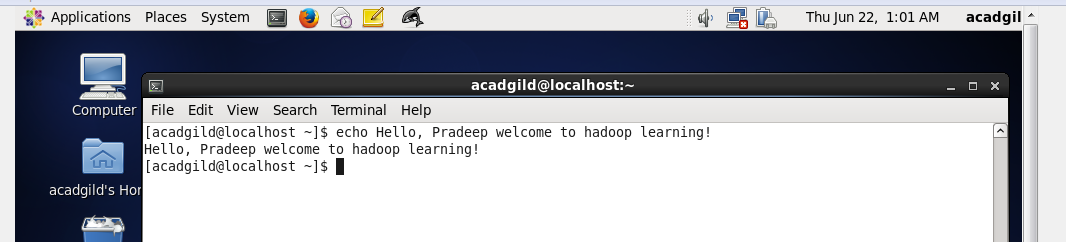
Syntax: echo[SHORT-OPTION]... [STRING]...

echo LONG-OPTION

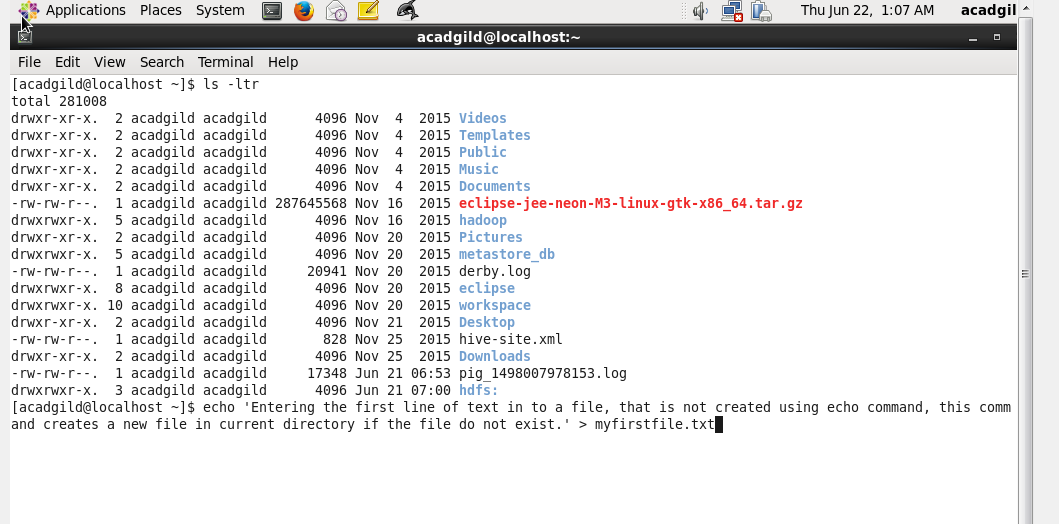
Examples:

echo Hello, World! ----> Output: Hello, World!

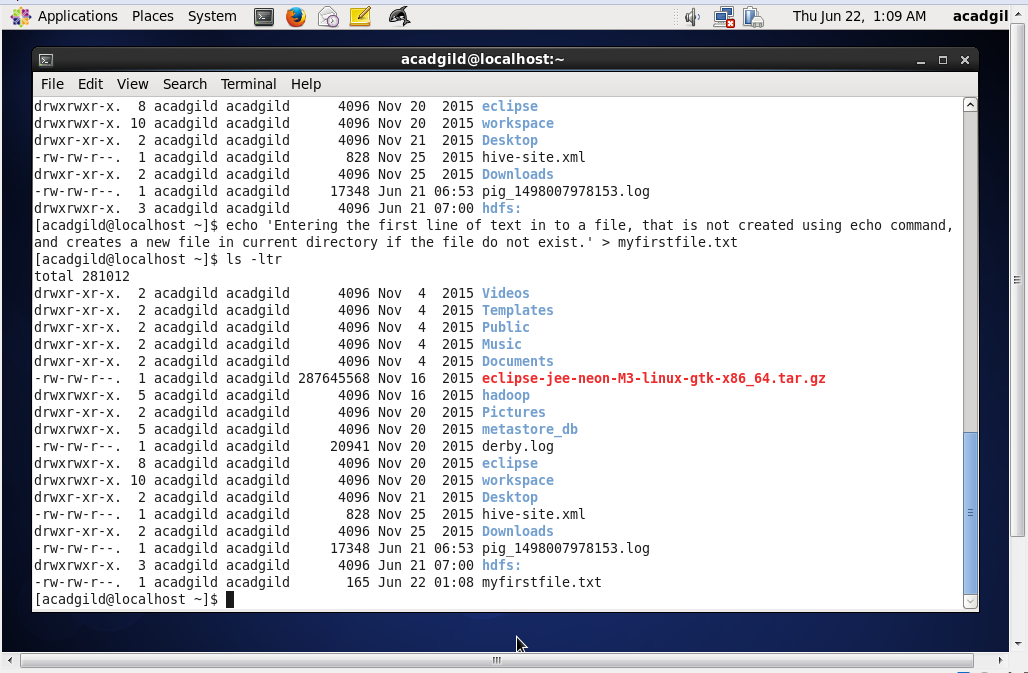
echo command is used to display a line of text.



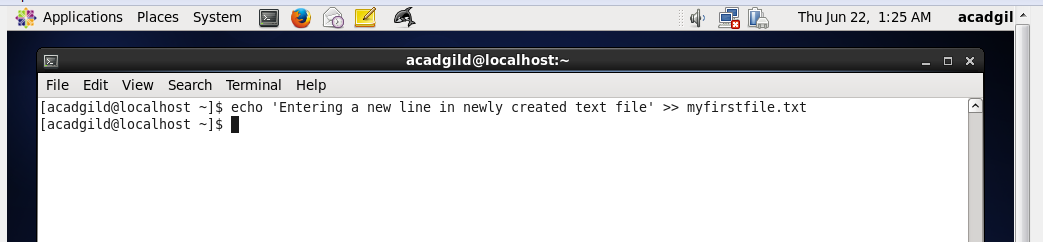
Creating a new file in current working directory and also entering text in to a file.



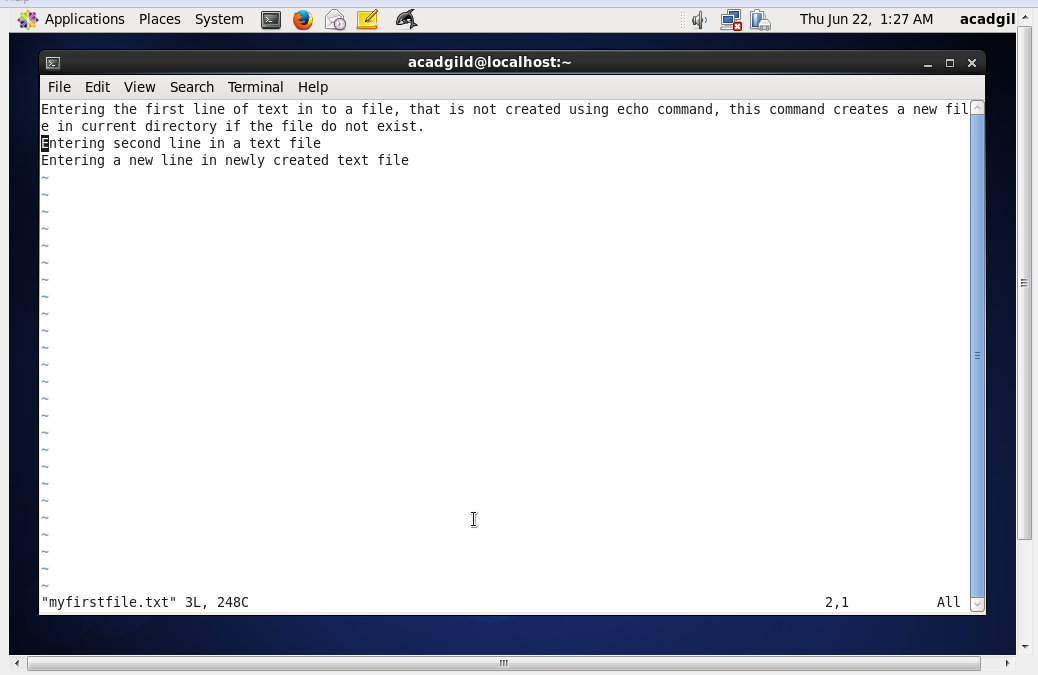
New file is created in current working directory.



Entering a new line in created file:



New line is added in to a file.



1. **cat:**

* cat stands for "catenate".
* cat command reads data from files, and outputs their contents.
* cat command is used to:

- display text files.

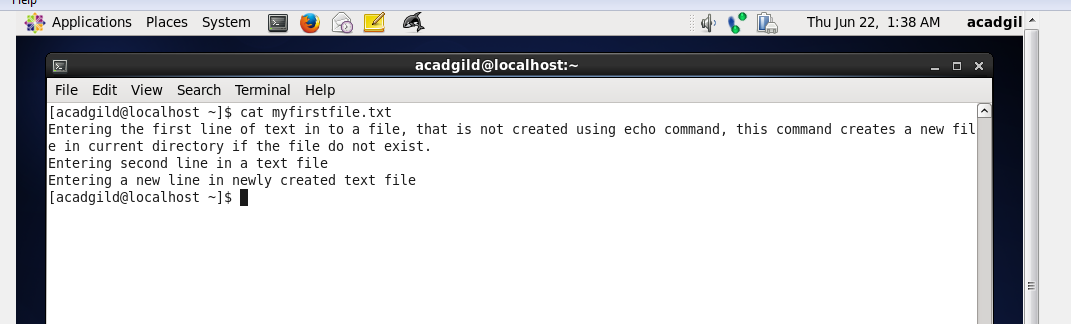
- copy text files in to a new document.

- Append the contents of a text file to the end of another text file, combining them.

Syntax: cat [OPTION]... [FILE]...

Examples:

1. cat mytext.txt ------> this line will read the contents of mytext.txt and send them to standard output [terminal screen]
2. cat mytext.txt mytext2.txt ----------> if more than one file name is specified, cat will display those files one after the other, catenating their contents to std o/p.
3. cat mytext.txt > newfile.txt ---------> copy a text file.
4. cat mytext.txt >> new-text-file.txt ----------> Append A text file's content to new text file.
5. cat -s file.txt -------> display the contents of file.txt, omitting any repeated blank lines.

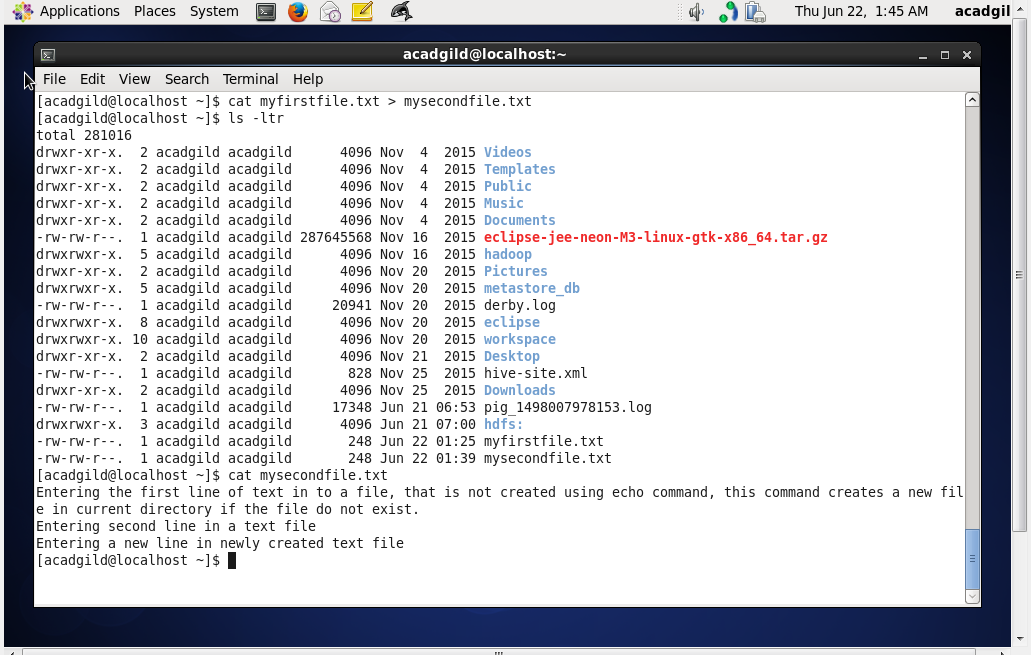


Using cat command, displaying contents of first file in second file (instead of displaying in standard output), here second file is created if it does not exist. If second file exist, existing contents in second file will be overridden.

Below screen shows that “mysecondfile.txt” is created



Contents of “mysecondfile.txt” is same as “myfirstfile.txt”:



1. **who:**

* who command displays who is logged on to the system.
* Who command prints information about all users who are currently logged in.

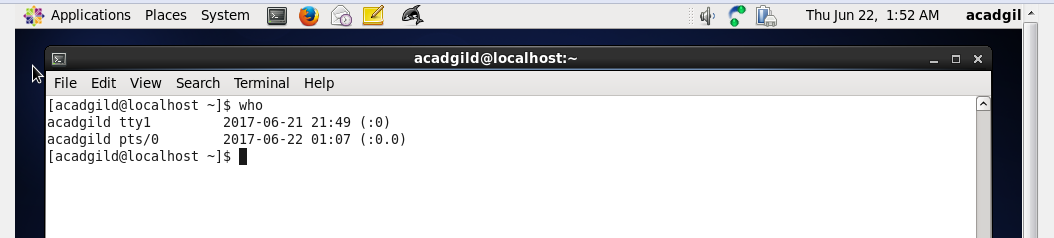
Syntax: who[OPTION]... [FILE] [am i]

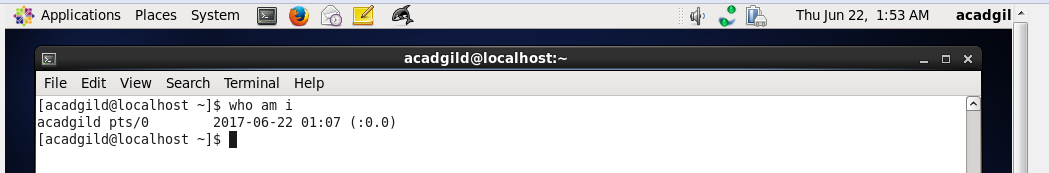
Examples:

who ---------> Displays the username, line, and time of all currently logged-in sessions.

who am i ---------> Displays same information as above, but only for the terminal session where the command was issued.

Who command displays the username, line and time of all currently logged in sessions:





1. **cd:**

* cd command is used to move around within the hierarchy of file system.
* cd stands for "change directory".
* cd changes the shell's current working directory.
* Representing root directory is by a single slash ("/"). To change root directory as our current working directory --> cd/
* Representing the working directory -> current directory is represented by a single dot (".")

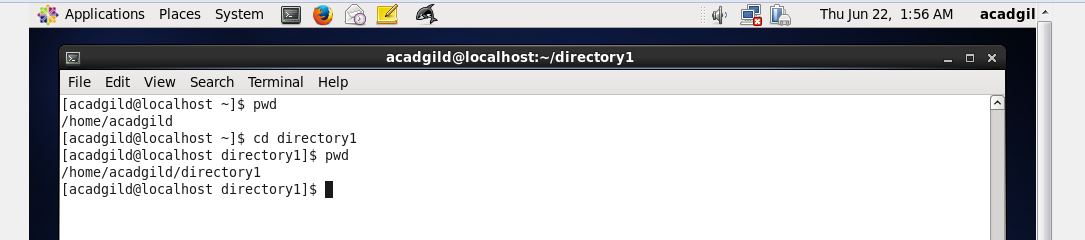
representing parent directory -> cd..

Syntax: cd [-L | -P] directory

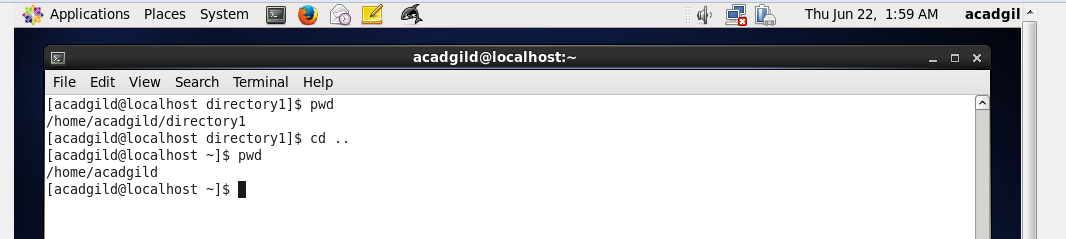
Example:

cd DirectoryName

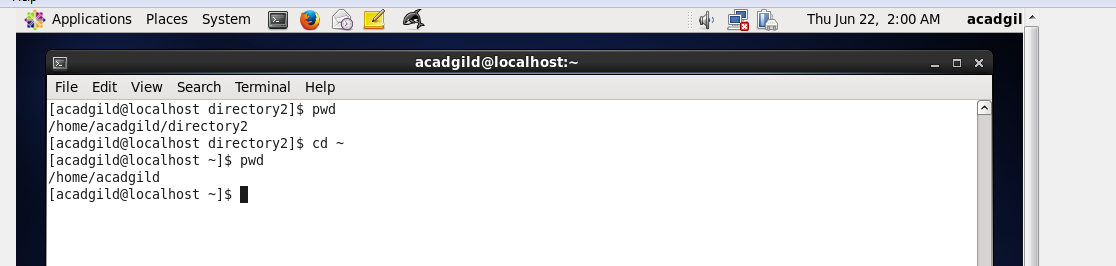
Using cd command changing directory:

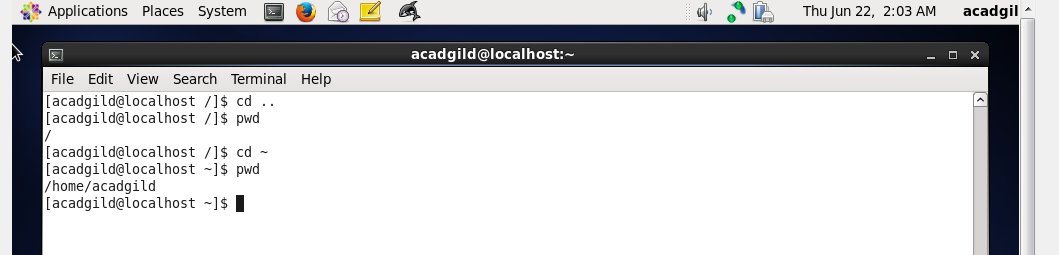


Representing parent directory:

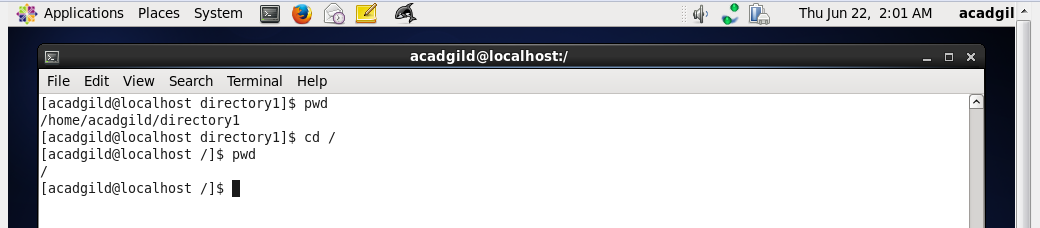


Representing home directory:

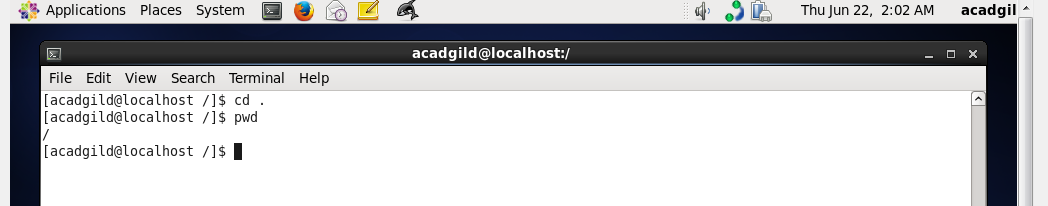




Representing root directory:



Representing current working directory:



1. **date:**

* date command is used to print out or change the value of system's time and date information.

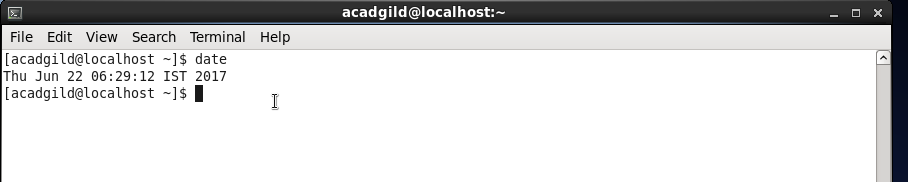
Syntax: date [OPTION] ... [+FORMAT]

Examples:

date ---> this command will output the system date and time with no options.

date -s "18/06/2017 20:10:30" ----> sets the system date and time to June 18,2017,20:10

Running **date** command with no options:



1. **cal:**

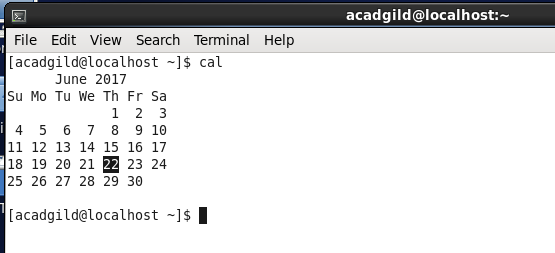
* cal displays a conveniently-formatted calendar from the command line.
* cal displays the current month at the command line, can be useful as part of a login script.

Syntax: cal [OPTIONS] [[[day] month] year]

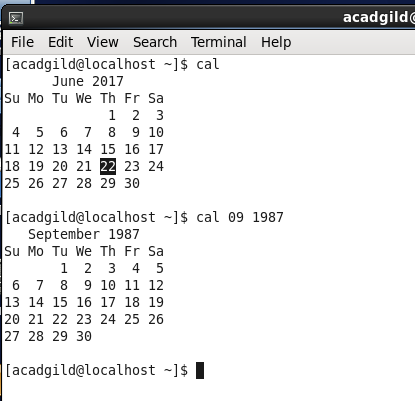
Examples:

cal --> displays the calendar of this month

cal 08 2017 --> displays the calendar for August of the year 2017



Using cal command displays the calendar for September of the year 1987



1. **mv:**

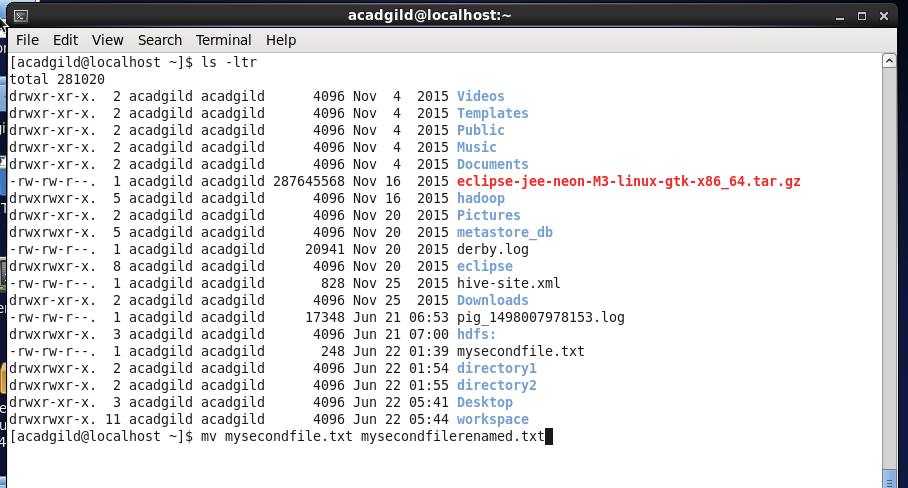
mv command is used to move or rename files.

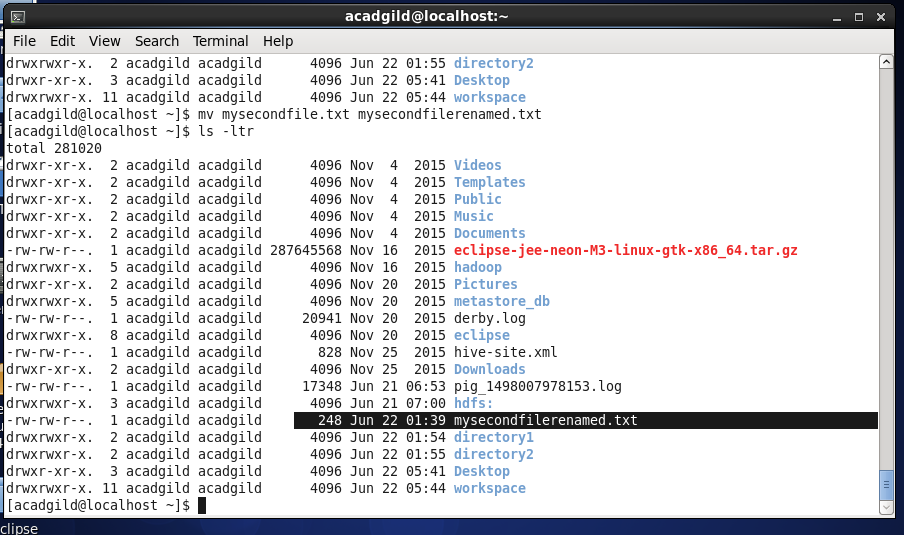
Syntax: mv [OPTION]... SOURCE ... DEST

Examples:

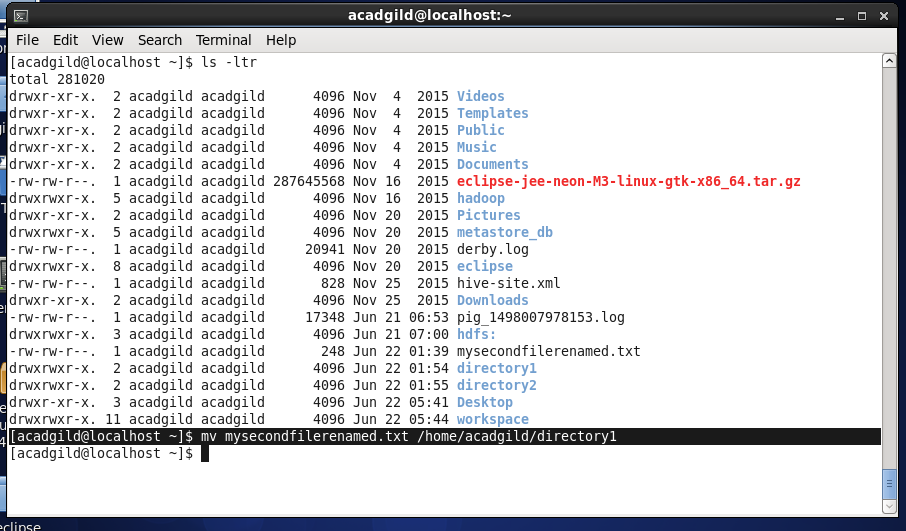
* mv file1.txt file2.txt ---> renaming files.
* mv myfile.txt destination-directory ---> moving "myfile.txt" to destination directory.

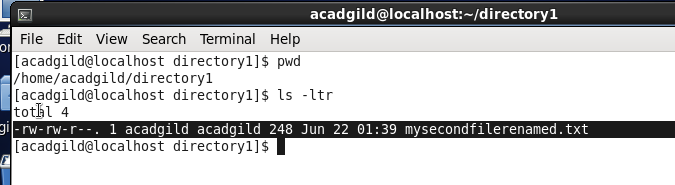
Renaming the file:





Moving the file in to another directory:





1. **cp:**

The cp command is used to make copies of files and directories:

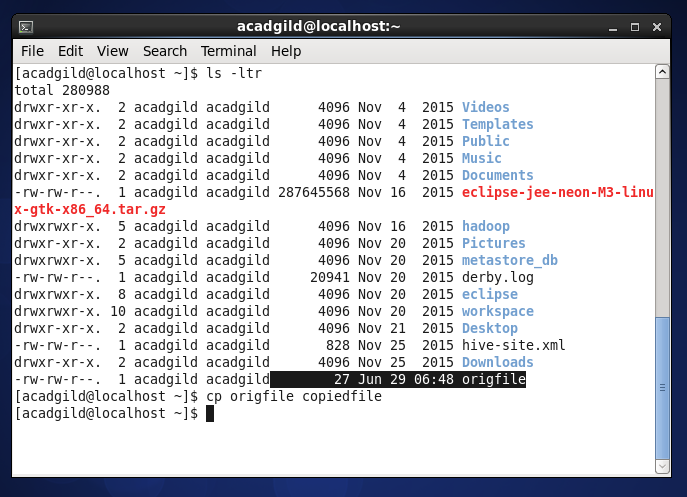
Syntax: cp [OPTION]...SOURCE...DIRECTORY

Examples:

* cp original\_file new\_file ----------> this copies a copy of the file in working directory
* cp -i oldfile newfile ----------------> if newfile already exists, following will be prompted: cp: overwrite 'newfile'?
* cp origfile /directory/subdirectory/newfile ----------> copies a file into another directory, and give it a new name..

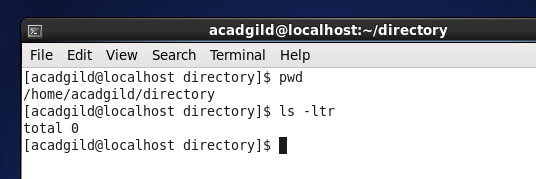
Creating a copy of file in working directory

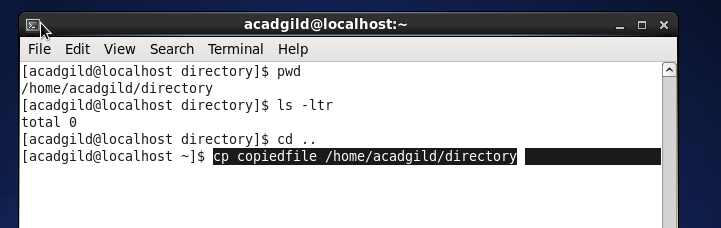


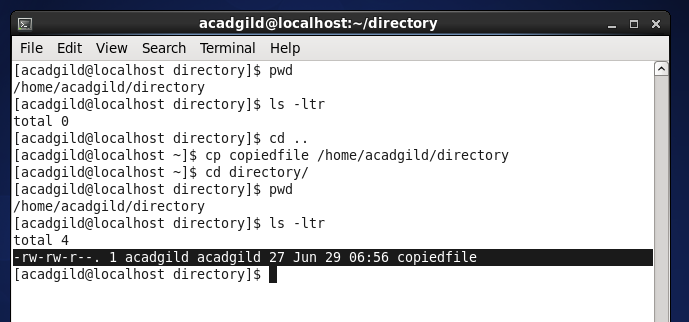




Copying a file in to another directory:







1. **which:**

* which command locate the executable file assosciated with a given command
* which command returns the pathnames of the files which would be executed in current environment

Syntax: which -a [filename]....

Example: which sh ------> this locates the pathname of the file which would be run if the "sh" command were executed -----> /bin/sh

