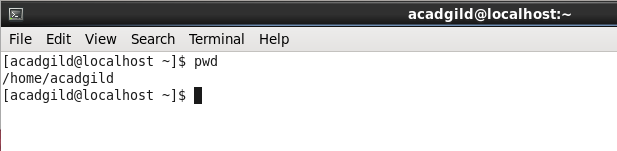
**Linux Commands**

1. *pwd* - prints the full pathname of the current working directory.

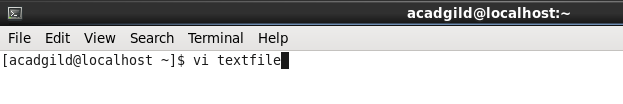
Eg:- pwd

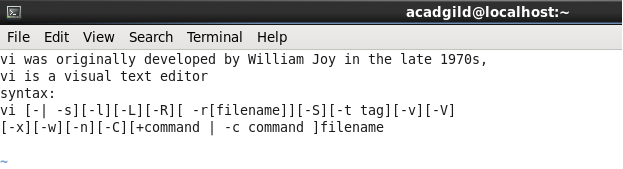


2. vi - vi is an interactive text editor which is display-oriented: the screen of your terminal acts as a window into the file you are editing. Changes you make to the file are reflected in what you see.

Using vi you can insert text anywhere in the file very easily. Most of the vi commands move the cursor around in the file. You can move the cursor forward and backward in units of characters, words, sentences, and paragraphs. Some of the operators, like d for delete and c for change, can be combined with the motion commands to make them operate on entire words, paragraphs, etc., in a natural way.

Eg:- vi firstfile



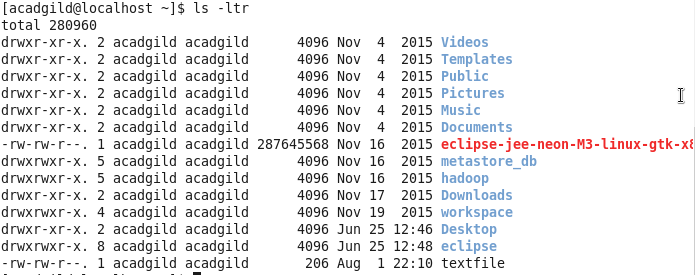
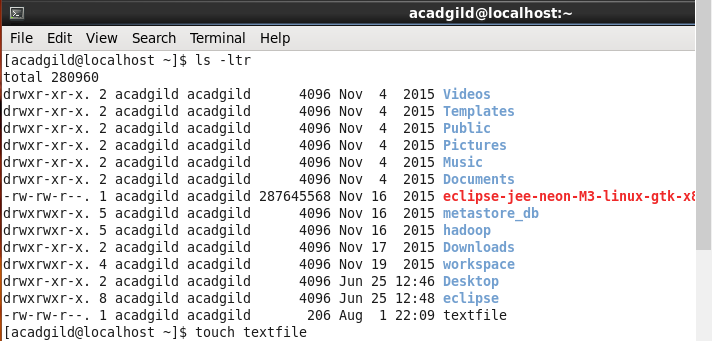


3. touch - The touch command updates the access and modification times of each FILE to the current system time.

If you specify a FILE that does not already exist, touch creates an empty file with that name (unless the -c or -h options are specified; see below).

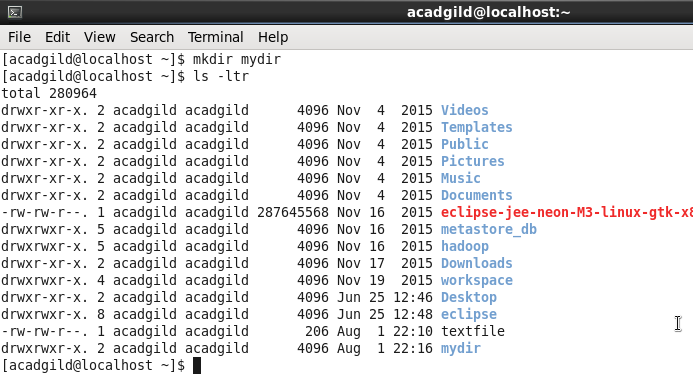
If the FILE argument is a dash ("-") is handled specially and causes touch to change the times of the file associated with standard output.

Eg:- touch file.txt



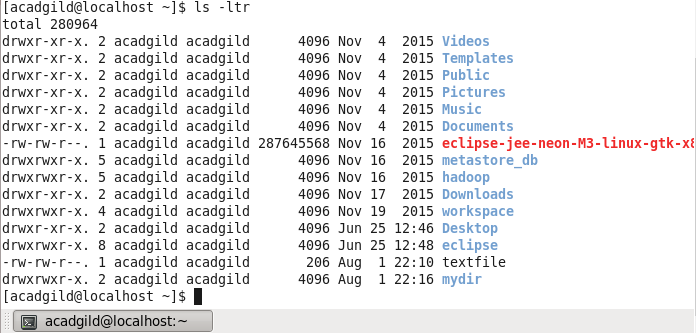
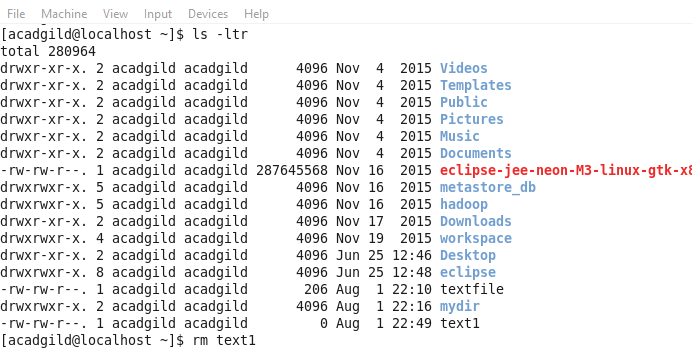
4. mkdir - mkdir is used to create directories on a file system. If the specified DIRECTORY does not already exist, mkdir creates it. More than one DIRECTORY may be specified when calling mkdir.

Eg:- mkdir mydir



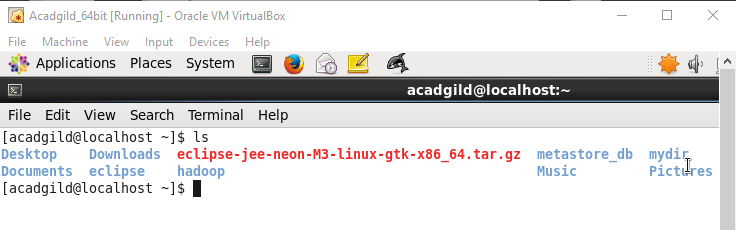
5. rm - The rm command removes (deletes) files or directories. rm removes each specified FILE. By default, it does not remove directories.

Eg:- rm myfile.txt



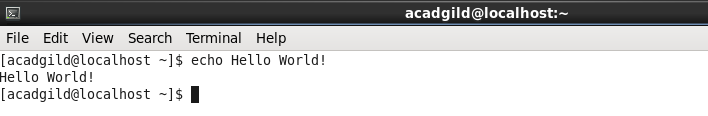
6. ls - Lists the contents of a directory. List information about the FILEs (the current directory by default). Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.

Eg:- ls



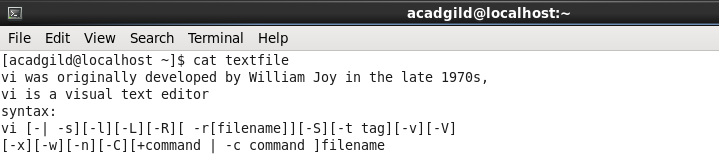
7. echo - echo displays a line of text. echo is a fundamental command found in most operating systems that offer a command line. It is frequently used in scripts, batch files, and as part of individual commands; anywhere you may need to insert text. Many command shells such as bash, ksh and csh implement echo as a built-in command.

Eg:- echo Hello, World!



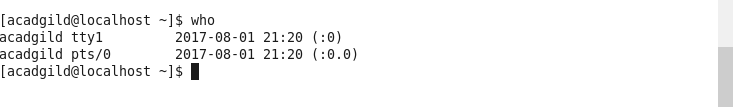
8. cat - cat stands for "catenate." It reads data from files, and outputs their contents. It is the simplest way to display the contents of a file at the command line.

Eg:- cat file.txt



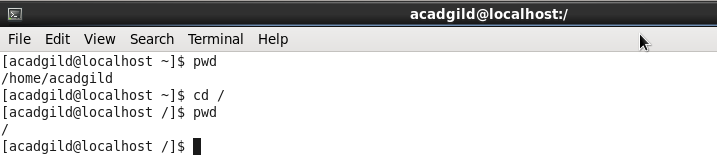
9. who - Displays who is logged on to the system. The who command prints information about all users who are currently logged in.

Eg:- who



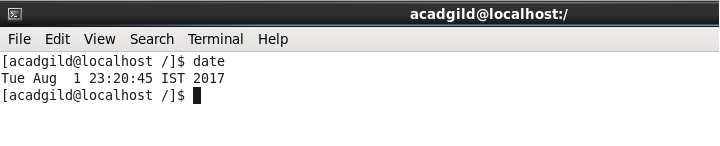
10. cd - CD (Change Directory) is a command used to switch directories. The CD command is an internal command.

Eg:- cd /



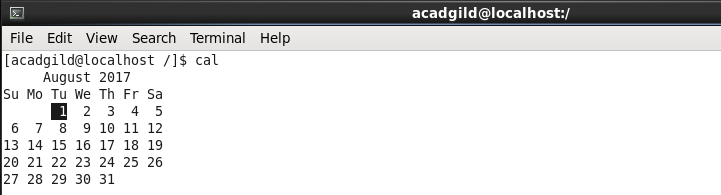
11. date - The date command is used to print out, or change the value of, the system's time and date information.

Eg:- date



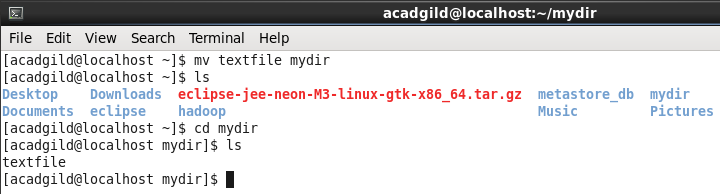
12. cal - Display a conveniently-formatted calendar from the command line. In general, if no options are given, cal displays the current month at the command line. It's a quick and convenient way to glance at the dates of the month, and can be useful as part of a login script.

Eg:- cal



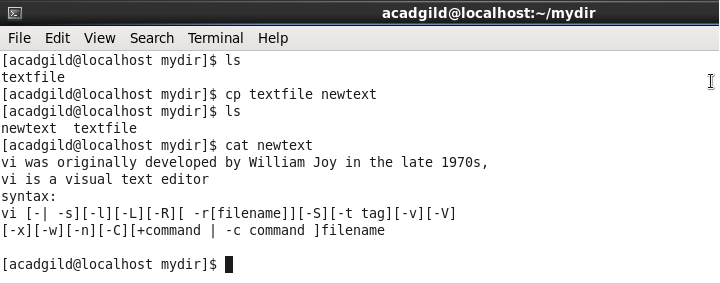
13. mv - The mv command is used to move or rename files. mv renames file SOURCE to DEST, or moves the SOURCE file (or files) to DIRECTORY.

Eg:- mv myfile.txt destination-directory



14. cp - The cp command is used to make copies of files and directories.

Eg:- cp origfile newfile



15.which - Locate the executable file associated with a given command. which returns the pathnames of the files (or links) which would be executed in the current environment, had the filename (or filenames) been given as a command (or commands) in a strictly POSIX-conformant shell. It does this by searching the paths in the PATH environment variable for executable files matching the names of the arguments. which does not follow symbolic links.

Eg:- which sh

