1. **pwd (print working directory)**

**Purpose:**

It displays the full path of current working directory.

**Syntax:**

**pwd**

**[Refer Screenshot1 which shows execution and output of this command]**

1. **vi (visual editor)**

**Purpose:**

vi editor is a full screen editor and has three modes of operation:

1. **Command mode**: In this mode, the characters we type, perform actions like as moving the cursor, cutting or copying text and searching for some particular text. So in this mode, any character that we pass from the keyboard does not show on the screen but performs a function. It is the default mode of vi, that is when we start vi, it comes in command mode.
2. **Insert mode:** In this mode (also known as input text mode), entered text is inserted into the file. Insertion of text can be performed only after changing from ‘command mode’ to ‘insert mode’ using the appropriate commands like i, I, o, O, R, s, S, a, A
3. **The ex mode:** In this mode, the user is permitted to give commands for saving the file or to exit the editor with or without saving the file, at the last line known as command line.

Below are the tables showing commands for insert mode and ex mode:

|  |  |  |
| --- | --- | --- |
| **Insert Mode** | | |
| **Command** | **Mode** | **Insertion Point** |
| a | append | just after the current character |
| A | Append | end of the current line |
| i | insert | just before the current character |
| I | Insert | beginning of the current line |
| o | open | new line below the current line |
| O | Open | new line above the current line |
| R | Replace | replaces text from cursor to right |
| s | replace | replaces single character under cursor with any number of characters |
| S | Replace | replaces entire line |
| **Ex Mode** | | |
| **Command** | **Action** | |
| :w | Saves file and remains in editing mode | |
| :w filename | Works like ‘Save As’ option | |
| :w! Filename | Works like ‘Save As’ option but overwrites the existing file | |
| :wq | Saves file and quits editing mode | |
| :q! | Quits vi editing mode abandoning the changes to a file | |

**Creating/ Editing a file with vi**

To create/ edit a file, use the command:

**Syntax:**

**vi filename**

If file named ‘filename’ doesn't exist then vi creates a new file, and If it already exists then text of the specified file appears on new screen which can further be edited.

**[Refer Screenshots 2.1 – 2.8 for creating a file with vi].**

**Description of Screenshots is as follows:**

|  |  |
| --- | --- |
| Screenshot | Description |
| 2.1 | Type command for creating a file in terminal window |
| 2.2 | vi editor opens in command mode and name of filename appears at bottom left corner of screen i.e.“students.txt” |
| 2.3 | After pressing ‘i’, we switch to insert mode, as shown at bottom left corner of screen |
| 2.4 | Type the content which must be placed inside the file |
| 2.5 | Press ‘esc’ to switch to command mode from insert mode |
| 2.6 | Press :wq to save the contents of file and to quit vi mode |
| 2.7 | Control now switches to terminal window |
| 2.8 | ls –lrt command confirms that file is created, and cat command displays the contents of file. |

1. **touch**

**Purpose:**

The touch command is used to create new empty [file](http://www.linfo.org/file.html)(s), if specified file does not already exist, and if it already exists then changes its most recent access and modification date and time to current data and time.

**Syntax:**

**touch [options] file\_name(s)**

**Below table shows various options:**

|  |  |
| --- | --- |
| **Option** | Description |
| **-a** | Changes only the access time. |
| **-c, --no-create** | Does not create any file. |
| **-d, --date=***STRING* | Parses the string STRING and use it instead of current time. |
| **-h, --no-dereference** | Affects each symbolic link instead of any referenced file (useful only on systems that can change the timestamps of a symbolic link). |
| **-m** | Changes only the modification time. |
| **-r, --reference=***FILE* | Uses this FILE's times instead of current time. |
| **-t STAMP** | Use [[CC]YY]MMDDhhmm[.ss] instead of current time. |
| **--time=***WORD* | Changes the specified time: WORD is access, atime, or use; or modify (equivalent to -a WORD), or mtime (equivalent to -m). |
| **--help** | Displays a help message, and exit. |
| **--version** | Displays version information, and exit. |

**[Refer Screenshots 3.1 – 3.6 for execution and output of the command]**

**Below is the description of the Screenshots:**

|  |  |
| --- | --- |
| **Screenshot** | **Description** |
| 3.1 | Touch command creates a file “**games”** as it does not already exist, and ls –lrt command shows that file is created, cat commands confirms that created file is empty, so vi command is used to edit the file. |
| 3.2 | vi editor screen opens in command mode, with file name at the bottom left corner of the screen. |
| 3.3 | After pressing ‘i’, mode switches from command to insert, now type the content. |
| 3.4 | After typing content, press ‘esc’ to return to command mode, and type ‘:wq’ to save and exit the vi editor. |
| 3.5 | We now return to terminal window. |
| 3.6 | Using cat command, we can see the content of file. |

1. **mkdir (Make Directory)**

**Purpose:**

It is used to create new directory/ subdirectory in the current directory.

**Syntax:**

**mkdir directoryname**

**[Refer Screenshot4 for execution and output of the command]**

**Description of Screenshot4 is as follows:**

|  |  |
| --- | --- |
| **Screenshot** | **Description** |
| 4 | mkdir creates a directory which can be seen with ls –lrt command. |

1. **rm (remove file/directory)**

**Purpose:**

Removes files/ directories.

**Syntax:**

**rm [option] filename/directoryname**

|  |  |
| --- | --- |
| **Option** | **Description** |
| -f | Force, removal of file, never prompts |
| -i | Prompts before every removal |
| -r, -R, --recursive | Removes directories and contents recursively |
| -v | Explains what is being done |
| --h | Displays help and exits |
| --v | Outputs version information and exits |

**[Refer Screenshots 5.1 - 5.4 for execution and output of command]**

**Description of Screenshots is as follows:**

|  |  |
| --- | --- |
| **Screenshot** | **Description** |
| 5.1 | ls –lrt command shows “games” file, which we want to delete |
| 5.2 | Using rm games command, file gets deleted, which can be confirmed using ls –lrt command |
| 5.3 | “Test” directory is created |
| 5.4 | If we try to remove directory using simple rm command, then it shows error, so use rm –r command to delete directory |

1. **ls**

**Purpose:**

It is used to display the list of files/ directories in the current directory.

**Syntax:**

**ls [option]**

**ls [option] directoryname**

|  |  |
| --- | --- |
| **Option** | **Description** |
| [ls -a](http://www.rapidtables.com/code/linux/ls/ls-a.htm) | list all files including hidden file starting with '.' |
| ls -i | list file's inode index number |
| [ls -l](http://www.rapidtables.com/code/linux/ls/ls-l.htm) | list with long format - show permissions |
| [ls -la](http://www.rapidtables.com/code/linux/ls/ls-l.htm) | list long format including hidden files |
| [ls -lh](http://www.rapidtables.com/code/linux/ls/ls-l.htm) | list long format with readable file size |
| [ls -ls](http://www.rapidtables.com/code/linux/ls/ls-l.htm) | list with long format with file size |
| [ls -r](http://www.rapidtables.com/code/linux/ls/ls-r.htm#reverse) | list in reverse order |
| [ls -R](http://www.rapidtables.com/code/linux/ls/ls-r.htm#recursive) | list recursively directory tree |
| [ls -s](http://www.rapidtables.com/code/linux/ls/ls-s.htm#size) | list file size |
| [ls -S](http://www.rapidtables.com/code/linux/ls/ls-s.htm#sort-size) | sort by file size |
| [ls -t](http://www.rapidtables.com/code/linux/ls/ls-t.htm) | sort by time & date |
| ls -X | sort by extension name |

**[Refer Screenshots 6.1 – 6.4 for execution and output of ls command]**

**Description of screenshots is as follows:**

|  |  |
| --- | --- |
| **Screenshot** | **Description** |
| 6.1 | ls –lrt lists files/ directories in current directory in long format with permissions, sorted by date and time and in reverse order (alphabetically). |
| 6.2 | ls –lrt lists files/ directories in root directory (i.e. /) in long format with permissions, sorted by date and time and in reverse order (alphabetically). |
| 6.3 | ls –s lists file/directory size along with file/directory name. |
| 6.4 | ls –X sorts by extension name in ascending order. |

1. **echo**

**Purpose:**

echo command writes its [arguments](http://www.linfo.org/argument.html) to [standard output](http://www.linfo.org/standard_output.html).

**Syntax:**

**echo [option(s)] [string(s)]**

**[Refer Screenshots 7.1 – 7.3 for execution and output of command].**

**Description of screenshots is as follows:**

|  |  |
| --- | --- |
| **Screenshot** | **Description** |
| 7.1 | If we type simple string, then error is returned, so we use echo command with or without quotes if string does not have any special character inside it. |
| 7.2 | echo $PATH command can be used to show a user's PATH environmental variable, which contains a colon-separated list of the directories that the system searches to find the executable program corresponding to a command issued by the user  echo $HOME shows the current user's [home directory](http://www.linfo.org/home_directory.html). |
| 7.3 | -e option is used to enable echo's interpretation of additional instances of the newline character as well as the interpretation of other special characters, such as horizontal tab, which is represented by \t. |

1. **cat (concatenate)**

**Syntax:**

Syntax of command is shown below:

|  |  |
| --- | --- |
| **Command** | **Purpose** |
| cat > filename | Creates a file if it does not exist and overwrites if it already exists and asks the user for content which is to be placed inside the file, and user can terminate the input by pressing ctrl + d command. |
| cat filename [filename,...] | Displays contents of single or multiple file(s). |
| cat –n filename | Displays the line number along with each line of a file. |
| cat file1 > file2 | Redirects standard output of a file into a new file or existing file with ‘**>**‘ (greater than) symbol. Careful, existing contents of **file2** will be overwritten by contents of **file1** file. |
| cat file1 >> file2 | Appends contents of existing file with ‘**>>**‘ (double greater than) symbol. Here, contents of **file1** file will be appended at the end of **file2** file. |
| cat < filename | Redirects standard input ‘**<**‘ (less than symbol), it uses file name **filename** as an input and output of the file will be shown in a terminal. |
| cat file1 file2 file3 > file4 | Creates a new file called **file4** and outputs from **file1, file2, file3** are redirected to **file4**. |
| Cat file1 file2 | sort > file3 | Creates a file **file3** and output of **cat** command is piped to **sort** command and sorted result is redirected to **file3**. |

**[Refer Screenshots 8.1 -8.3 for execution and output of command].**

|  |  |
| --- | --- |
| **Screenshot** | **Description** |
| 8.1 | Shows how a new file can be created and its contents can be shown, and if file already exists, then old contents are overridden by new contents. |
| 8.2 | Line numbers are shown along with each line of the file |
| 8.3 | Shows how contents of one file can be redirected to another file. |

1. **who**

**Purpose:**

This command prints information about all users who are currently logged in.

**Syntax:**

**who [option] [file] [am i]**

|  |  |
| --- | --- |
| **Option** | **Description** |
| **-a, --all** | Same as using the options -b -d --login -p -r -t -T -u. |
| **-b, --boot** | Display the time of the last system [boot](https://www.computerhope.com/jargon/b/boot.htm). |
| **-d, --dead** | Display dead [processes](https://www.computerhope.com/jargon/p/process.htm). |
| **-H, --heading** | Print a line of column headings. |
| **--ips** | Prints [IP](https://www.computerhope.com/jargon/i/ip.htm) addresses instead of [hostnames](https://www.computerhope.com/jargon/h/hostname.htm). With --lookup, canonicalizes based on stored IP, if available, rather than stored hostname. |
| **-l, --login** | Print system login processes. |
| **--lookup** | Attempts to canonicalize hostnames via [DNS](https://www.computerhope.com/jargon/d/dns.htm). |
| **-m** | Only prints information about the user and host associated with [standard input](https://www.computerhope.com/jargon/s/stdin.htm) (the [terminal](https://www.computerhope.com/jargon/t/terminal.htm) where the command was issued). This method adheres to the [POSIX](https://www.computerhope.com/jargon/p/posix.htm) standard. |
| **-p, --process** | Print active processes spawned by [init](https://www.computerhope.com/unix/telinit.htm). |
| **-q, --count** | Display all login names, and a count of all logged-on users. |
| **-r, --runlevel** | Print the current [runlevel](https://www.computerhope.com/jargon/r/runlevel.htm). |
| **-s, --short** | Print only name, line, and time fields, which is the default. |
| **-t, --time** | Print the last time the system [clock](https://www.computerhope.com/jargon/c/clock.htm) was changed, if the information is available. |
| **-T, -w, --mesg** | Add a [character](https://www.computerhope.com/jargon/c/charact.htm) which indicates the state of the terminal line: "+" if the terminal is writable, "-" if it is not, or "?" if a bad line is encountered. |
| **-u, --users** | Print the idle time for each user, and the process ID. |
| **--message** | Same as -T. |
| **--writable** | Same as -T. |
| **--help** | Displays a help message, and exit. |
| **--version** | Displays version information, and exit. |

**Key Notes:** If ***file*** is specified, **who** gathers its information from this file. Otherwise, it reads from a default file location.

If the arguments **"am i"** are specified, **who** assumes the **-m** option.

**[Refer Screenshots 9.1 – 9.2 for execution and output of the command]**

**Description of screenshots is as follows:**

|  |  |
| --- | --- |
| **Screenshot** | **Description** |
| 9.1 | **who** displays the username, line, and time of all currently logged-in sessions and **who am i** displays the same information, but only for the terminal session where the command was issued. |
| 9.2 | **who –a** displays all information, and **who –aH** displays all information along with columns headers for each column. |

1. **cd (change directory)**

**Purpose:**

It is used to change current [working directory](https://www.computerhope.com/jargon/c/currentd.htm). It is used to move around within the [hierarchy](https://www.computerhope.com/jargon/h/hierfile.htm) of the [file system](https://www.computerhope.com/jargon/f/filesyst.htm).

**Syntax:**

**cd [option] [directory]**

|  |  |
| --- | --- |
| **Option** | **Description** |
| -P | Does not follow symbolic links |
| -L | Follows symbolic links (default) |

**[Refer Screenshots 10.1 - 10.2 for execution and output of the command]**

**Description of screenshots is as follows:**

|  |  |
| --- | --- |
| **Screenshot** | **Description** |
| 10.1 | Command **cd Documents** changes from current directory i.e. **/home/acadgild** to **/home/acadgild/Documents** |
| 10.2 | **cd mydata** changes from **/home/acadgild/Documents** to /**home/acadgild/Documents/mydata** |
| 10.3 | **cd ..** it results in going one level up the directory  **cd ../..** it results in going two levels up the directory  **cd ~** it returns to the home directory |

1. **date**

**Purpose:**

date command is helpful to display date in several formats. It also allows to set system date and time.

**Syntax:**

**date [option] ... [+format]**

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

|  |  |
| --- | --- |
| **Option** | **Description** |
| **-d**, **--date=***STRING* | Display time described by [string](https://www.computerhope.com/jargon/s/string.htm) *STRING*, as opposed to the default, which is '**now**'. |
| **-f**, **--file=***DATEFILE* | Like **--date**, but processed once for each line of file *DATEFILE*. |
| **-I**[*TIMESPEC*], **--iso-8601**[**=***TIMESPEC*] | Output date/time in [ISO](https://www.computerhope.com/jargon/i/iso.htm) 8601 format. For values of *TIMESPEC*, use '**date**' for date only (the default), '**hours**', '**minutes**', '**seconds**', or '**ns**' for date and time to the indicated precision. |
| **-r**, **--reference=***FILE* | Display the last modification time of file *FILE*. |
| **-R**, **--rfc-2822** | Output date and time in [RFC](https://www.computerhope.com/jargon/r/rfc.htm) 2822 format. Example: **Mon, 07 Aug 2006 12:34:56 -0600** |
| **--rfc-3339=***TIMESPEC* | Output date and time in RFC 3339 format. *TIMESPEC* can be set to '**date**', '**seconds**', or '**ns**' for date and time to the indicated precision. Date and time components are separated by a single space, for example: **2006-08-07 12:34:56-06:00** |
| **-s**, **--set=***STRING* | Set time described by string *STRING*. |
| **-u**, **--utc**, **--universal** | Print or set Coordinated Universal Time. |
| **--help** | Display a help message and exit. |
| **--version** | Display version information and exit. |

**Date Format**

FORMAT is a sequence of characters which specifies how output will appear. It comprises some combination of the following sequences:

|  |  |
| --- | --- |
| **Format** | **Description** |
| **%%** | A literal percent sign ("**%**"). |
| **%a** | The abbreviated weekday name (e.g., **Sun**). |
| **%A** | The full weekday name (e.g., **Sunday**). |
| **%b** | The abbreviated month name (e.g., **Jan**). |
| **%B** | Locale's full month name (e.g., **January**). |
| **%c** | The date and time (e.g., **Thu Mar 3 23:05:25 2005**). |
| **%C** | The current century; like **%Y**, except omit last two digits (e.g., **20**). |
| **%d** | Day of month (e.g., **01**). |
| **%D** | Date; same as **%m/%d/%y**. |
| **%e** | Day of month, space padded; same as **%\_d**. |
| **%F** | Full date; same as **%Y-%m-%d**. |
| **%g** | Last two digits of year of ISO week number (see **%G**). |
| **%G** | Year of ISO week number (see **%V**); normally useful only with **%V**. |
| **%h** | Same as **%b**. |
| **%H** | Hour (**00**..**23**). |
| **%I** | Hour (**01**..**12**). |
| **%j** | Day of year (**001**..**366**). |
| **%k** | Hour, space padded ( **0**..**23**); same as **%\_H**. |
| **%l** | Hour, space padded ( **1**..**12**); same as **%\_I**. |
| **%m** | Month (**01**..**12**). |
| **%M** | Minute (**00**..**59**). |
| **%n** | A newline. |
| **%N** | Nanoseconds (**000000000**..**999999999**). |
| **%p** | Locale's equivalent of either **AM** or **PM**; blank if not known. |
| **%P** | Like **%p**, but lower case. |
| **%r** | Locale's 12-hour clock time (e.g., **11:11:04 PM**). |
| **%R** | 24-hour hour and minute; same as **%H:%M**. |
| **%s** | Seconds since 1970-01-01 00:00:00 UTC. |
| **%S** | Second (**00**..**60**). |
| **%t** | A tab. |
| **%T** | Time; same as **%H:%M:%S**. |
| **%u** | Day of week (**1**..**7**); 1 is **Monday**. |
| **%U** | Week number of year, with Sunday as first day of week (**00**..**53**). |
| **%V** | ISO week number, with Monday as first day of week (**01**..**53**). |
| **%w** | Day of week (**0**..**6**); 0 is **Sunday**. |
| **%W** | Week number of year, with Monday as first day of week (**00**..**53**). |
| **%x** | Locale's date representation (e.g., **12/31/99**). |
| **%X** | Locale's time representation (e.g., **23:13:48**). |
| **%y** | Last two digits of year (**00**..**99**). |
| **%Y** | Year. |
| **%z** | +hhmm numeric time zone (e.g., **-0400**). |
| **%:z** | +hh:mm numeric time zone (e.g., **-04:00**). |
| **%::z** | +hh:mm:ss numeric time zone (e.g., **-04:00:00**). |
| **%:::z** | Numeric time zone with "**:**" to necessary precision (e.g., **-04**, **+05:30**). |
| **%Z** | Alphabetic time zone abbreviation (e.g., EDT). |

By default, date pads numeric fields with zeroes. The following optional flags may follow '%':

|  |  |
| --- | --- |
| **Flags** | **Description** |
| **-** | ([Hyphen](https://www.computerhope.com/jargon/h/hyphen.htm)) do not pad the field. |
| **\_** | Pad with [spaces](https://www.computerhope.com/jargon/s/space.htm). |
| **0** | Pad with zeros. |
| **^** | Use upper case if possible. |
| **#** | Use opposite case if possible. |

**[Refer Screenshots for execution and output of command]**

|  |  |
| --- | --- |
| **Screenshot** | **Description** |
| 11.1 | **date** prints system date and time  **date "+DATE: %m/%d/%y%nTIME: %H:%M:%S"** prints the date in output format, (refer the screenshot) |
| 11.2 | Displays past date and future date |
| 11.3 | Displays weekday name and month name |
| 11.4 | Displays Date and Time in output format |

1. **cal (Calendar)**

**Purpose:**

The cal command is a command line utility for displaying a calendar in the terminal. It can be used to print a single month, many months or an entire year. It supports starting the week on a Monday or a Sunday, showing Julian dates and showing calendars for arbitrary dates passed as arguments.

**Syntax:**

**cal [options] [[[day] month] year]**

|  |  |
| --- | --- |
| **Option** | **Description** |
| **-1** | Display a single month, which is the default setting. |
| **-3** | Display three months: last month, this month, and next month. |
| **-s** | Display the calendar using Sunday as the first day of the week. |
| **-m** | Display Monday as the first day of the week. |
| **-j** | Display dates of the Julian calendar. |
| **-y** | Display a calendar for the entire current year. |

**[Refer Screenshots 12.1 – 12.3 for execution and output of this command]**

**Description of screenshots is as follows:**

|  |  |
| --- | --- |
| **Screenshot** | **Description** |
| 12.1 | **cal** and **cal -1** displays the calendar for the current month. |
| 12.2 | **cal -3** displaysthree months: last month, current month, next month**.** |
| 12.3 | **cal –y** displays the calendar for entire current year. |

1. **mv (move)**

**Purpose:**

This command is used to move or rename a file.

**Syntax:**

**mv [option] source destination**

|  |  |
| --- | --- |
| **Option** | **Description** |
| **--backup**[**=***CONTROL*] | Make a backup of each existing destination file, using the control method *CONTROL*. |
| **-b** | Like **--backup**, but does not accept an argument; the default backup method is used. |
| **-f**, **--force** | Do not prompt before overwriting existing files. |
| **-i**, **--interactive** | Prompt before overwriting each existing destination file, regardless of the file's permissions. If the answer to the prompt is negative, the file is skipped. |
| **-n**, **--no-clobber** | Do not overwrite any existing file. |

If you specify more than one of the above options **-i**, **-f**, or **-n**, only the final option specified takes effect.

|  |  |
| --- | --- |
| **--strip-trailing-slashes** | Remove any trailing slashes from each *SOURCE* argument. |
| **-S**, **--suffix=***SUFFIX* | Use the suffix *SUFFIX* for all backup files. The default *SUFFIX* is "**~**". |
| **-t**, **--target-directory=***DIRECTORY* | Move all *SOURCE* arguments into directory *DIRECTORY*. |
| **-T**, **--no-target-directory** | Treat *DEST* as a normal file, not as a directory. |
| **-u**, **--update** | Perform the move only if the *SOURCE* file is newer than the destination file, or the destination file does not already exist. |
| **-v**, **--verbose** | Operate verbosely. |
| **--help** | Displays a help message, and exit. |
| **--version** | Outputs version information, and exit. |

## Backup Methods

The default ("**simple**") suffix for backup files is '**~**', but it can be set differently with the **--suffix** option or the **SIMPLE\_BACKUP\_SUFFIX** environment variable.

The version control method may be selected via the **--backup** option or through the **VERSION\_CONTROL** environment variable. The possible control methods are:

|  |  |
| --- | --- |
| **none**, **off** | Never make backups, even if the **--backup** option is given. |
| **numbered**, **t** | Make numbered backups. |
| **existing**, **nil** | **Numbered** if numbered backups already exist, **simple** otherwise. |
| **simple**, **never** | Always make simple backups. |

**[Refer Screenshots 13.1 – 13.2 for execution and output of command]**

**Description of screenshots is as follows:**

|  |  |
| --- | --- |
| **Option** | **Description** |
| 13.1 | **students.txt** file is moved from **Documents** directory to **mydata** directory |
| 13.2 | Moves **students.txt** from **mydata** to its parent directory i.e. **Documents** directory |

1. **cp (copy)**

**Purpose:**

This command is used to copy files and directories.

**Syntax:**

**copy [option] source destination**

|  |  |
| --- | --- |
| **Option** | **Description** |
| cp -a | archive files |
| cp -f | force copy by removing the destination file if needed |
| cp -i | interactive - ask before overwrite |
| cp -l | link files instead of copy |
| cp -L | follow symbolic links |
| cp -n | no file overwrite |
| cp -R | recursive copy (including hidden files) |
| cp -u | update - copy when source is newer than destination |
| cp -v | verbose - print informative messages |

**[Refer Screenshots 14.1 – 14.2 for execution and output of command]**

**Description of Screenshots is as follows:**

|  |  |
| --- | --- |
| **Screenshot** | **Description** |
| 14.1 | Copies **students.txt** file from **Documents** directory to **Documents/mydata** directory |
| 14.2 | Copies entire **Documents** directory inside **Documents\_Copy** directory recursively |

1. **which**

**Purpose:**

**which** command is very small and simple command to locate executables in the system. It allows user to pass several command names as arguments to get their paths in the system. “which” command searches the path of executable in system paths set in $PATH [environment variable](http://www.thegeekstuff.com/2012/07/linux-export-command-examples/).

**Syntax:**

**which [option] argument(s)**

**-a : “**which” command with option “-a” displays all paths of executable matching to argument.

**[Refer Screenshot 15 for execution and output of which command]**