

UNIT 2: GETTING STARTED WITH UNIX, UNIX SHELL COMMAND, TEXT EDITING WITH VI

CS – 22: Operating Systems Concepts With Unix / Linux



1. Unix Architecture and Features and types Of Shell
2. Unix File System & Types of Files & Directory Permissions
3. Login Commands: passwd, logout, who, who am i, clear
4. File / Directory Related Command ls, cat, cd, pwd, mv, cp, rm, rmdir, mkdir, chmod, find, pg, more, less, head, tail, wc, touch
5. Finding Patterns in Files: grep, fgrep, egrep
6. Working with columns and fields: cut , paste, join
7. Tools for sorting: sort, uniq
8. Comparing files : cmp, comm, diff
9. Tools for mathematical calculations: bc, factor
10. Tools For Displaying Date and Time : cal, date
11. Communications: wall, mtod, write, finger
12. Process Related Commands: ps, background, nice, kill, at, batch, cron, crontab, wait, sleep
13. Mounting and Demounting a File System
14. vi editor

Assignment Questions

1. What is directory files?
2. Which command is used to remove any file in unix?
3. What is permission in unix?
4. The “who am I” command used for what?
5. What is shell? Explain any one shell type in detail.
6. Explain head and tail commands.
7. Explain unix file system.
8. Explain chmod, chown and chgrp commands.
9. Explain VI editor and it's modes.
10. Explain bc, kill, finger, mount and cal commands.

CS – 22 : Operating Systems Concepts With Unix / Linux

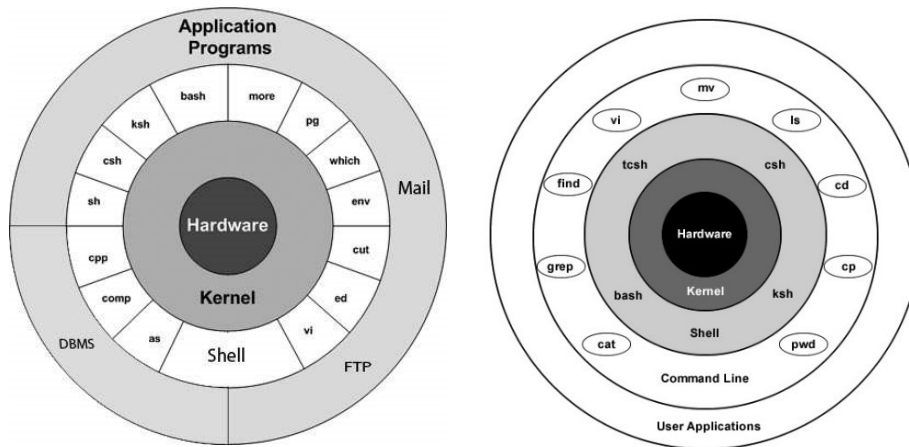
Unix Architecture and Features and types Of Shell

Architecture

The Unix operating system is a set of programs that act as a link between the computer and the user. The computer programs that allocate the system resources and coordinate all the details of the computer's internals is called the operating system or the kernel.

Users communicate with the kernel through a program known as the shell. The shell is a command line interpreter; it translates commands entered by the user and converts them into a language that is understood by the kernel.

- Unix was originally developed in 1969 by a group of AT&T employees Ken Thompson, Dennis Ritchie, Douglas McIlroy, and Joe Ossanna at Bell Labs.
- There are various Unix variants available in the market. Solaris Unix, AIX, HP Unix and BSD are a few examples. Linux is also a flavor of Unix which is freely available.
- Several people can use a Unix computer at the same time; hence Unix is called a multiuser system.
- A user can also run multiple programs at the same time; hence Unix is a multitasking environment.



Kernel – The kernel is the heart of the operating system. It interacts with the hardware and most of the tasks like memory management, task scheduling and file management.

Shell – The shell is the utility that processes your requests. When you type in a command at your terminal, the shell interprets the command and calls the program that you want. The shell uses standard syntax for all commands. C Shell, Bourne Shell and Korn Shell are the most famous shells which are available with most of the Unix variants.

Commands and Utilities – There are various commands and utilities which you can make use of in your day-to-day activities. cp, mv, cat and grep, etc. are few examples of commands and utilities. There are over 250 standard commands plus numerous others provided through 3rd party software. All the commands come along with various options.

Files and Directories – All the data of Unix is organized into files. All files are then organized into directories. These directories are further organized into a tree-like structure called the filesystem.

Features of UNIX

Multiuser System:

Unix provides multiple programs to run and compete for the attention of the CPU. This happens in 2 ways:

- Multiple users running multiple jobs
- Single user running multiple jobs

In UNIX, resources are actually shared between all the users, so-called a multi-user system. For doing so, computer give a time slice (breaking unit of time into several segments) to each user. So, at any instant of time, only one user is served but the switching is so fast that it gives an illusion that all the users are served simultaneously.

Multitask System:

A single user may run multiple tasks concurrently. Example: Editing a file, printing another on the printer & sending email to a person, and browsing the net too at the same time. The Kernel is designed to handle user's multiple needs.

The important thing here is that only one job can be seen running in the foreground, the rest all seems to run in the background. Users can switch between them, terminate/suspend any of the jobs.

The building-Block Approach:

The Unix developers thought about keeping small commands for every kind of work. So Unix has so many commands, each of which performs one simple job only. You can use 2 commands by using pipes ('|'). Example: \$ ls | wc Here, | (pipe) connects 2 commands to create a pipeline. This command counts the number of files in the directory. These types of connected commands that can filter/manipulate data in other ways are called filters.

The UNIX Toolkit:

Unix has a kernel but the kernel alone can't do much that could help the user. So, we need to use the host of applications that usually come along with the UNIX systems. The applications are quite diversified. General-purpose tools, text manipulation utilities (called filters), compilers and interpreters, networked programs, and system administration tools are all included. With every UNIX release, new tools are being added and the older ones are modified/ removed.

Pattern Matching:

Unix provides very sophisticated pattern matching features. The meta-char '*' is a special character used by the system to match a number of file names. There are several other meta-char in UNIX. The matching is not confined to only filename. Advanced tools use a regular expression that is framed with the characters from this set.

Programming Facility:

Unix provides shell which is also a programming language designed for programmers, not for casual end-users. It has all the control structures, loops, and variables required for programming purposes. These features are used to design the shell scripts (programs that can invoke the UNIX commands). Many functions of the system can be controlled and managed by these shell scripts.

Documentation:

It has a 'man' command that stands for the manual, which is the most important reference for any commands and their configuration files. Apart from the online documentation, there is a vast number of resources available on the Internet. If you're stuck with a problem, there are various UNIX newsgroups where you can post your concerns.

Login Commands

passwd: passwd command in Linux is used to change the user account passwords. The root user reserves the privilege to change the password for any user on the system, while a normal user can only change the account password for his or her own account.

Syntax:

passwd [options] [username]

Example:

Command: passwd

logout: logout command allows you to programmatically logout from your session. causes the session manager to take the requested action immediately.

who: The who command is used to get information about currently logged in user on to system. who command is used to find out the following information:

- Time of last system boot
- Current run level of the system
- List of logged in users and more.

who am i: 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.

clear: clear is a standard Unix computer operating system command that is used to clear the terminal screen. This command first looks for a terminal type in the environment and after that, it figures out the terminfo database for how to clear the screen. And this command will ignore any command-line parameters that may be present. Also, the clear command doesn't take any argument and it is almost similar to cls command on a number of other Operating Systems.

man: man command in Linux is used to display the user manual of any command that we can run on the terminal. It provides a detailed view of the command which includes name, synopsis, description, options, exit status, return values, errors, files, versions, examples, authors and see also.

Every manual is divided into the following sections:

- Executable programs or shell commands
- System calls (functions provided by the kernel)
- Library calls (functions within program libraries)
- Games
- Special files (usually found in /dev)
- File formats and conventions eg /etc/passwd
- Miscellaneous (including macro packages and conventions), e.g. groff(7)
- System administration commands (usually only for root)
- Kernel routines [Nonstandard]

File / Directory Related Command

ls: The ls command is used to list files or directories in Linux and other Unix-based operating systems. Just like you navigate in your File explorer or Finder with a GUI, the ls command allows you to list all files or directories in the current directory by default, and further interact with them via the command line.

ls option	Description
ls -a	In Linux, hidden files start with. (dot) symbol and they are not visible in the regular directory. The (ls -a) command will enlist the whole list of the current directory including the hidden files.
ls -d */	It is used to display only subdirectories.
ls -r	It is used to print the list in reverse order.
ls -R	It will display the content of the sub-directories also.
ls -lt	It will sort the list by displaying recently modified files at top.
ls --version	It checks the version of ls command.

cat: The 'cat' command is the most universal and powerful tool. It is considered to be one of the most frequently used commands. It can be used to display the content of a file, copy content from one file to another, concatenate the contents of multiple files, display the line number, display \$ at the end of the line, etc.

Syntax: cat <fileName>

Option	Function
cat > [fileName]	To create a file.
cat [oldfile] > [newfile]	To copy content from older to new file.
cat [file1 file2 and so on] > [new file name]	To concatenate contents of multiple files into one.
cat -n/cat -b [fileName]	To display line numbers.
cat -e [fileName]	To display \$ character at the end of each line.
cat [fileName] <<EOF	Used as page end marker.

cd: Linux cd command is used to change the current working directory (i.e., in which the current user is working). The "cd" stands for 'change directory.' It is one of the most frequently used commands in the Linux terminal. It is one of the most important and common commands in the Linux system and will be used repeatedly. With the help of this command, we can move all over our directories in our system. We can go to our previous directory or previous to the next directory, or anywhere.

Syntax: cd <directory-name>

pwd: 'pwd' stands for 'Print Working Directory'. As the name states, command 'pwd' prints the current working directory or simply the directory user is, at present. It prints the current directory name with the complete path starting from root (/).

mv: Linux mv command is used to move existing file or directory from one location to another. It is also used to rename a file or directory. If you want to rename a single directory or file then 'mv' option will be better to use.

Syntax: mv <file-name> <file-name>

Option	Function
mv -i	Asks for permission to over write.
mv *	Move multiple files to a specific directory.
mv -u	Only move those files that doesn't exist.

cp: 'cp' means copy. 'cp' command is used to copy a file or a directory.

Syntax: cp <file-name> <file-name/location>

Option	Function
cp -r	To copy a directory along with its sub directories.
cp file1 file2 directory name	To copy multiple file or directories in a directory.
cp -backup	To back up the existing file before over writing it.
cp -i	Asks for confirmation.
cp -p	Preserves attribute of a file.

rm: The 'rm' means remove. This command is used to remove a file. The command line doesn't have a recycle bin or trash unlike other GUI's to recover the files. Hence, be very much careful while using this command. Once you have deleted a file, it is removed permanently.

Syntax: rm <file-name/directory-name>

Option	Description
rm *extension	Used to delete files having same extension.
rm -r or R	To delete a directory recursively.
rm -i	Remove a file interactively.
rm -rf	Remove a directory forcefully.

mkdir: The mkdir stands for 'make directory'. With the help of mkdir command, you can create a new directory wherever you want in your system. Just type "mkdir <dir name>", in place of <dir name> type the name of new directory, you want to create and then press enter.

Syntax: mkdir <dirname>

rmdir: This command is used to delete a directory. But will not be able to delete a directory including a sub-directory. It means, a directory has to be empty to be deleted.

Syntax: rmdir <dirname>

chmod:

Linux chmod command is used to change the access permissions of files and directories. It stands for change mode. It cannot change the permission of symbolic links. Even, it ignores the symbolic links come across recursive directory traversal.

In the Linux file system, each file is associated with a particular owner and have permission access for different users. The user classes may be:

- Owner
- Group member
- Others (Everybody else)

The file permissions in Linux are the following three types:

- read (r)
- write (w)
- execute (x)

Syntax: chmod <options> <permissions> <file name>

Options	Description
F --quite --silent	It is used to suppress the error messages.
V	verbose It is used to display a diagnostic for every processed file.
R	recursive It is used to change files and directories recursively.

head: The head command, as the name implies, print the top N number of data of the given input. By default, it prints the first 10 lines of the specified files. If more than one file name is provided then data from each file is preceded by its file name.

Syntax: head <file name> <file name>

Option -n: print the first num of lines instead of the first 10

tail: The tail command, as the name implies, print the last N number of data of the given input. By default it prints the last 10 lines of the specified files. If more than one file name is provided then data from each file is precedes by its file name.

Option -n: print the first num of lines instead of the last 10

find: The find command in UNIX is a command line utility for walking a file hierarchy. It can be used to find files and directories and perform subsequent operations on them. It supports searching by file, folder, name, creation date, modification date, owner and permissions.

Syntax: find <location> <comparison-criteria> <search-term>

pg: pg displays a text file, pausing after each "page" (the height of the terminal screen). After each page, a prompt is displayed. The user may then either press the newline key to view the next page or one of the keys described below.

-number The number of lines per page. Usually, this is the number of CRT lines minus one.

-c Clear the screen before a page is displayed, if the terminfo entry for the terminal provides this capability.

-e Do not pause and display "(EOF)" at the end of a file.

more : more command is used to view the text files in the command prompt, displaying one screen at a time in case the file is large (For example log files). The more command also allows the user do scroll up and down through the page.

Syntax: more <file-name>

Following keys are used in 'more' command to scroll the page:

- Enter key: To scroll down page line by line.
- Space bar: To go to next page.
- b key: To go to the backward page.
- / key: Lets you search the string.

Options	Function
more -num	Limits the line displayed per page.
more -d	Displays user message at right corner.
more -s	Squeeze blank lines.
more +/string name	It helps to find the string.
more +num	Used to display the content from a specific line.

less: Less command is a Linux utility that can be used to read the contents of a text file one page(one screen) at a time. It has faster access because if file is large it doesn't access the complete file, but accesses it page by page. For example, if it's a large file and you are reading it using any text editor, then the complete file will be loaded to main memory. The less command doesn't load the entire file, but loads it part by part which makes it faster.

Syntax: less <file-name>

wc: Linux wc command helps in counting the lines, words, and characters in a file. It displays the number of lines, number of characters, and the number of words in a file.

Syntax: wc <file-name>

-c	It is used to print the byte counts.
-m	It is used to print the character counts.
-l	It is used to print the newline counts.

touch: touch command is a way to create empty files (there are some other methods also). You can update the modification and access time of each file with the help of touch command.

Syntax : touch <filename>

Finding Patterns in Files

grep: The grep filter searches a file for a particular pattern of characters, and displays all lines that contain that pattern. The pattern that is searched in the file is referred to as the regular expression (grep stands for global search for regular expression and print out).

Syntax: grep [options] pattern [files]

Options Description

-c : This prints only a count of the lines that match a pattern

-h : Display the matched lines, but do not display the filenames.

-i : Ignores, case for matching

-n : Display the matched lines and their line numbers.

-v : This prints out all the lines that do not matches the pattern

-w : Match whole word

-o : Print only the matched parts of a matching line, with each such part on a separate output line.

fgrep: The fgrep filter is used to search for the fixed-character strings in a file. There can be multiple files also to be searched. This command is useful when you need to search for strings which contain lots of regular expression metacharacters, such as “^”, “\$”, etc.

Syntax: fgrep [options] [-e pattern_list] [pattern] [file]

egrep: egrep is a pattern searching command which belongs to the family of grep functions. It works the same way as grep -E does. It treats the pattern as an extended regular expression and prints out the lines that match the pattern. If there are several files with the matching pattern, it also displays the file names for each line.

Syntax: egrep [options] 'PATTERN' files

Working with columns and fields

cut: The cut command in UNIX is a command for cutting out the sections from each line of files and writing the result to standard output. It can be used to cut parts of a line by byte position, character and field. Basically, the cut command slices a line and extracts the text. It is necessary to specify option with command otherwise it gives error. If more than one file name is provided then data from each file is not preceded by its file name.

Syntax: cut OPTION... [FILE]...

- b It is used to cut a specific section by bytes.
- c It is used to select the specified characters.
- d It is used to cut a specific section by a delimiter.
- n: It is used to ignore any option.

paste: Paste command is one of the useful commands in Unix or Linux operating system. It is used to join files horizontally (parallel merging) by outputting lines consisting of lines from each file specified, separated by tab as delimiter, to the standard output.

Syntax: paste [OPTION]... [FILES]

- d : Specify of a list of delimiters.

join: The join command in UNIX is a command line utility for joining lines of two files on a common field. Suppose you have two files and there is a need to combine these two files in a way that the output makes even more sense. For example, there could be a file containing names and the other containing ID's and the requirement is to combine both files in such a way that the names and corresponding ID's appear in the same line. join command is the tool for it. join command is used to join the two files based on a key field present in both the files. The input file can be separated by white space or any delimiter.

Tools for sorting: sort, uniq
Comparing files :

cmp

Comm: compare two sorted files line by line and write to standard output: the lines that are common, plus the lines that are unique. Before 'comm' can be used, the input files must be sorted.

- 1 suppress lines unique to file1.
- 2 suppress lines unique to file2.
- 3 suppress lines that appear in both files.

Syntax: comm [options]... File1 File2

diff: diff stands for difference. This command is used to display the differences in the files by comparing the files line by line. Unlike its fellow members, cmp and comm, it tells us which lines in one file have to be changed to make the two files identical.

The important thing to remember is that diff uses certain special symbols and instructions that are required to make two files identical. It tells you the instructions on how to change the first file to make it match the second file.

- a: add
- c: change
- d: delete

Syntax: diff [options] File1 File2

-i : By default this command is case sensitive. To make this command case in-sensitive use -i option with diff.

Tools for mathematical calculations

bc: bc command is used for command line calculator. It is similar to basic calculator by using which we can do basic mathematical calculations.

Arithmetic operations are the most basic in any kind of programming language. Linux or Unix operating system provides the bc command for doing arithmetic calculations.

Syntax: bc

Factor: The factor command in Linux is used to print the prime factors of the given numbers, either given from command line or read from standard input. The numbers given through standard input may be delimited by tabs, spaces or newlines.

Syntax: factor [NUMBER]

Tools For Displaying Date and Time

cal: If a user wants a quick view of the calendar in the Linux terminal, cal is the command for you. By default, the cal command shows the current month calendar as output.

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]

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. By default the date command displays the date in the time zone on which unix/linux operating system is configured. You must be the super-user (root) to change the date and time.

Syntax: date

Process Related Commands

ps: As we all know Linux is a multitasking and multi-user systems. So, it allows multiple processes to operate simultaneously without interfering with each other. Process is one of the important fundamental concept of the Linux OS. A process is an executing instance of a program and carry out different tasks within the operating system. Linux provides us a utility called ps for viewing information related with the processes on a system which stands as abbreviation for “Process Status”. ps command is used to list the currently running processes and their PIDs along with some other information depends on different options.

Syntax: ps

-ax List currently running process

-u <username> List process for specific user

-p <pid> List process with given PID

background:

To run a process in the background, Unix provides the ampersand (&) symbol. While executing a command, if this symbol is placed at the end of the command, then the command will be executed in the background. When you run a process in the background a number is displayed on the screen.

This number is nothing but the PID of the process that you have just executed in the background.

Example

```
sort abc.txt > new_abc.txt &  
17653
```

The task of sorting the file abc.txt and storing the output in new_abc.txt has now been assigned to the background, letting the user free to carry out any other task in the foreground.

Advantages and limitation of background processes

- On terminal of background process, no success or failure is reported on the screen. Then how do we keep track of it? That is where the PID displayed on the screen comes in handy. We can search for this PID in the output of ps to verify whether the process is still running or has been terminated.
- The output of a background process should always be redirected to a file. Otherwise, you would get a garbled screen showing the output of the background process along with whatever you are doing in the foreground.
- With too many processes running in the background the overall system performance is likely to degrade.
- If you log out while some of your processes are running in the background all these processes would be abandoned halfway through. This is natural because all your processes are children/grandchildren/great grandchildren of your sh (shell) process and when we log out the sh process dies along with all its children.

nice: nice command in Linux helps in execution of a program/process with modified scheduling priority. It launches a process with a user-defined scheduling priority. In this, if we give a process a higher priority, then Kernel will allocate more CPU time to that process. nice command in Linux helps in execution of a program/process with modified scheduling priority. It launches a process with a user-defined scheduling priority. In this, if we give a process a higher priority, then Kernel will allocate more CPU time to that process.

Syntax: nice [OPTION] [COMMAND]

-n Add integer N to the niceness (default is 10).

Kill : kill command in Linux (located in /bin/kill), is a built-in command which is used to terminate processes manually. kill command sends a signal to a process which terminates the process.

Syntax: kill <pid>

at: at command is a command-line utility that is used to schedule a command to be executed at a particular time in the future. Jobs created with at command are executed only once. The at command can be used to execute any program or mail at any time in the future. It executes commands at a particular time and accepts times of the form HH:MM to run a job at a specific time of day. This command is not available by default in the OS unless its installed.

cron: The cron is a software utility, offered by a Linux-like operating system that automates the scheduled task at a predetermined time. It is a daemon process, which runs as a background process and performs the specified operations at the predefined time when a certain event or condition is triggered without the intervention of a user. Dealing with a repeated task frequently is an intimidating task for the system administrator and thus he can schedule such processes to run automatically in the background at regular intervals of time by creating a list of those commands using cron. It enables the users to execute the scheduled task on a regular basis unobtrusively like

doing the backup every day at midnight, scheduling updates on a weekly basis, synchronizing the files at some regular interval.

Syntax: cron [-f] [-l] [-L loglevel]

-f : Used to stay in foreground mode, and don't daemonize.

-l : This will enable the Linux Standard Base compliant names for /etc/cron.d files.

-n : Used to add the Fully Qualified Domain Name in the subject when sending mails.

-L loglevel : This option will tell the cron what to log about the jobs with the following values:

1 : It will log the start of all cron jobs.

2 : It will log the end of all cron jobs.

4 : It will log all the failed jobs. Here the exit status will not equal to zero.

8 : It will log the process number of all the cron jobs.

wait: wait is an inbuilt command in the Linux shell. It waits for the process to change its state i.e. it waits for any running process to complete and returns the exit status.

Syntax: wait <pid>

sleep: sleep command is used to create a dummy job. A dummy job helps in delaying the execution. It takes time in seconds by default.

Syntax: sleep <seconds>

Mounting and demounting

Mount: The mount command attaches the filesystem of an external device to the filesystem of a system. It instructs the operating system that filesystem is ready to use and associate it with a particular point in the system's hierarchy. Mounting will make files, directories and devices available to the users.

Syntax: mount -t type <device> <directory>

It mounts the external storage devices like hard disks, pen drives, USBs etc.

umount: The umount command detaches the file system(s) mentioned from the file hierarchy. A file system is specified by giving the directory where it has been mounted. Note that a file system cannot be unmounted when it is 'busy' - for example, when there are open files on it, or when some process has its working directory there, or when a swap file on it is in use.

Syntax: umount <device>

-f force unmount

VI Editor

The default editor that comes with the UNIX operating system is called vi (visual editor). Using vi editor, we can edit an existing file or create a new file from scratch. we can also use this editor to just read a text file.

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.

Commands and their Description

vi <filename>: Creates a new file if it already not exists, otherwise opens existing file.

vi -R <filename>: Opens an existing file in read only mode.

view <filename>: Opens an existing file in read only mode.

Moving within a File(Navigation)/cursor command:

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 **|**: 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.

Control Commands (Scrolling): There are following useful commands which can used along with Control Key:

Commands and their Description:

CTRL+d: Move forward 1/2 screen.

CTRL+f: Move forward one full screen.

CTRL+u: Move backward 1/2 screen.

CTRL+b: Move backward one full screen.

CTRL+e: Moves screen up one line.

CTRL+y: Moves screen down one line.

CTRL+u: Moves screen up 1/2 page.

CTRL+d: Moves screen down 1/2 page.

CTRL+b: Moves screen up one page.

CTRL+f: Moves screen down one page.

CTRL+l: Redraws screen.

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 before the cursor location.

x: Deletes the character at 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 Paste 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.

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.