Vim Commands & man Commands

-Snehal Deshmukh

Case 1:-Basic commands

- touch command: Create new files
- mkdir command: Make new folders
- cat command: Read a text file
- ls command: List the contents of a folder
- clear: Clear terminal screen
- exit: Close the terminal

Vim Commands
Case 2:-

- S- Deletes the line where the cursor is located & enters insert mode.
- s- Deletes the character highlighted by the cursor.
- d$- deletes what is beyond the cursor on the same line.
- $y- Copies the word from where the cursor is placed.

Vim Commands
Case 3:-

- p-Pastes to the position after the cursor
- P-Pastes to the position preceding the cursor
- Yw-Copies the word character where the cursor is placed

man Commands
Case 1:-

- man ls- It shows all commands (directory)
- q- Close terminal

GRUB / boot loader :-

- GRUB is an acronym for Grand Unified Boot Loader and is the primary and most crucial component for booting your device.
- most prevalent bootloader for Linux.
- it runs first when a machine is turned on, regular users rarely see GRUB in action. It functions automatically and requires no user input.

GRUB Commands:-

- boot – cd /boot
- List- ls
- Grub- cd grub

- cd : this command is used to change directory.
- cd /: this command is used to change directory to the root directory, The root directory is the first directory in your filesystem hierarchy.
- If we use **ls** in current directory, then we don’t need to use / to change the directory within the directory.

Absolute Path-:

➢ We have to mention the whole path starting from the root (/). Consider the below example:

✓ cat/home/snehal/sneha.txt

This is called an absolute path.

- nano /home/snehal/sneha.txt
- vi /home/snehal/sneha.txt
- head /home/snehal/sneha.txt
- tail /home/snehal/sneha.txt

Relative Path-:

- ➢ We can change our directory by using a relative path; a relative path is a location that is relative to the current directory.
- ✓ Ex. cat ~/sneha.txt

This is the relative path.

- ➢ tail /home/snehal/sneha.txt

Extra commands -:

- ➢ Change to the previous directory -: cd -
- ✓Change to Parent Directory -: cd ../ or cd ..

❖ Linux file structure :

- /bin: contains essential command binaries
- /sbin: contains essential system binaries
- /etc: contains system-wide configuration files
- /lib: contains system libraries
- /opt: contains optional software packages
- /var: contains files that are expected to change during the normal operation of the system, such as log files and data files for programs.
- /home: contains the home directories for individual users, where they can store their own files and settings.
- /tmp: contains temporary files that are automatically deleted when the system is rebooted.

- mnt – Mount Directory Temporary mount directory where sysadmins can mount filesystems.
- /media – Removable Media Devices
- Temporary mount directory for removable devices.
- For examples, /media/cdrom for CD-ROM; /media/floppy for floppy drives; /media/cdrecorder for CD writer
- 16. /srv – Service Data srv stands for service.
- Contains server specific services related data.
- For example, /srv/cvs contains CVS related data.

Ubuntu Terminal Shortcuts	Function
Ctrl + E	Move cursor to end of line
Ctrl + W	Delete the word before the cursor
Ctrl + C	Kill the current process
Ctrl + Z	Suspend the current process by sending the signal SIGSTOP
Ctrl + L	Clears the terminal output
TAB	Used to complete the command you are typing. If more than one command is possible, you can press it multiple times to scroll through the possible completions. If a very wide number of commands are possible, it can output a list of all possible completions.

/dev:-

- **locate** command: You can use the locate command to search for a file by name. For example, locate myfile will search the database for the file named "myfile" and will return the path of the file if it exists.

- **dev build** - Building and compiling code: For example, in a JavaScript project using webpack, the dev command can be used to run the webpack command to build the project.
- **dev test** - Running tests: For example, in a Node.js project using Jest, the dev command can be used to run the jest command to run all the tests.

/lib:-

- **lib create -n <library-name> -t <library-type>** - As a command for creating a library: The lib command could be used to create a library, such as creating a shared library or a static library.
- **lib dependencies -l <library-name>** - As a command for managing library dependencies: The lib command could be used to manage library dependencies, such as listing the dependencies, checking for missing dependencies, or updating the dependencies of a library.

/mnt:-

- **mnt –l** - As a command for managing mount points: The mnt command could be used to manage mount points, such as listing the currently mounted file systems, unmounting a file system, or changing the options of a mounted file system.
- **mnt -c /mnt/data** - As a command for creating a mount point: The mnt command could be used to create a mount point.
- **mount <sourceDevicePath> <destination path>** - for mounting file
- **unmount <sourceDevicePath> <destination path>** - for unmount

/etc:-

- **etc push -e production**- As a configuration management tool:
- **etc set -e ENV_VARIABLE=value** - As a command for managing environment variables:
- **etc useradd xyz** - As a system administration tool

/proc :-

- **cat /proc/stat** - To see the current CPU usage, you can use the command cat /proc/stat or top.
- **cat /proc/net/dev** - To see the current network usage, you can use the command cat /proc/net/dev

/run :-

- To find a file or directory, you can use the command find followed by the path to search and the name of the file or directory you want to find. For example, **find / -name file.txt** would search the entire file system for a file named "file.txt"

- To check the process status, you can use the command **ps** or **top**.
- run -t my_tests -As a command for running tests: The run command could be used to run tests for a specific application or library, such as unit tests or integration tests.

/var :-

- To create a new variable and assign a value, you can use the export command. For example, export MYVAR="Hello World" would create a new variable named "MYVAR" and assign the value "Hello World" to it.
- To modify the value of a variable, you can simply assign a new value to it. For example, **MYVAR**="Hello again" would change the value of the variable "**MYVAR**" to "Hello again".
- To remove a variable, you can use the unset command followed by the variable name. For example, **unset MYVAR** would remove the variable "**MYVAR**" from the shell environment.

/usr:-

- To copy a file from the /usr directory to another directory, you can use the command **cp /usr/filename /path/to/destination**. This will copy the file "filename" from the /usr directory to the directory specified by "/path/to/destination".(sudo cp /usr/test.txt /home)

- To move a file from the /usr directory to another directory, you can use the command **mv /usr/filename /path/to/destination**.