Chapter 3

Basic File management

1. Regular files : binary or contains text eg: shell scripts or commands in binary form
2. Block and character special device files: device files , that's how the peripheral devices communicate
   a. Character : "c" -
   b. Block : "b"
3. Symbolic links :

Compression and archiving

1. For faster transfer b/w the remote systems
2. Conserve space by archiving the big large files or directories into zips
   a. gzip
   b. Bzip2

The tar and star commands have the ability to preserve general file attributes such as ownership, owning group, and timestamp as well as extended attributes

gzip (compress) and gunzip (decompress)

➢ The gzip command is used to create a compressed file of each of the specified files and it adds the .gz extension to each fil e for identification.

➢ This tool can be used with the -r option to compress an entire directory tree, and with the -l option to display compression information about a gzipped file.

```
geek@ubuntu:~$ gzip fstab
geek@ubuntu:~$ ls
fstab.gz
geek@ubuntu:~$ []
```

tar (tap archive)

The tar (tape archive) command is used to create, append, update, list, and extract files or an entire directory tree to and from a single file, which is called a tarball or tarfile.

bzip2 and bunzip2 (Better because of the compression ration when compared to the gzip)

```
fstab
geek@ubuntu:~$ bzip2 fstab
geek@ubuntu:~$ ls
fstab.bz2
geek@ubuntu:~$ bunzip2 fstab.bz2
geek@ubuntu:~$ ls
fstab
geek@ubuntu:~$ █
```

File editor

VIM

TL;DR – How to Exit Vim

1. Press ESC once (*sometimes twice*)
2. Make sure you are using the English input method
3. The next step depends on the current status and your expectations:
- If you didn't make any changes, type :q and press Enter/return
- If you made some changes and would like to **keep** them, type :wq and press Enter/return

```
VIM - Vi IMproved

version 8.1.3741
by Bram Moolenaar et al.
Modified by team+vim@tracker.debian.org
Vim is open source and freely distributable

Help poor children in Uganda!
type  :help iccf<Enter>        for information

type  :q<Enter>                to exit
type  :help <Enter>  or  <F1>  for on-line help
type  :help version8<Enter>    for version info
```

expectations.
- If you didn't make any changes, type :q and press Enter/return
- If you made some changes and would like to **keep** them, type :wq and press Enter/return
- If you made some changes and would rather **discard** them, type :q! and press Enter/return

3 modes
➢ command mode

The command mode is the default mode of vim. The vim editor places you into this mode when you start it. While in the command mode, you can carry out tasks such as copy, cut, paste, move, remove, replace, change, and search on text, in addition to performing navigational operations

➢ Input mode

In the input mode, anything that is typed on the keyboard is entered into the file as text. Commands cannot be run in this mode. The input mode is also called the edit mode or the insert mode. You need to press the Esc key to return to the command mode.

➢ Last line mode

Video for VIM tutorial

Youtube : Vim Text Editor Basics - How to Use Vim on Linux / Mac / Unix (Beginners Guide on Ubuntu 20.04)
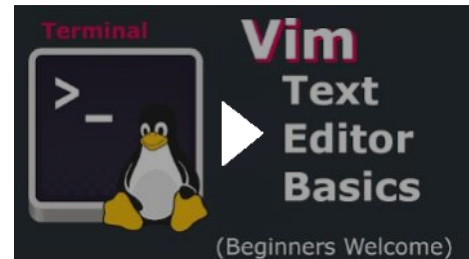

```
vim somefilename - this creates a new file called somefilename

j k h l / arrow keys - to navigate through the file using vim

i - lets you insert text wherever the cursor currently is
o - creates a new line under where your cursor and allows to insert text

:q! - quits out and disregards all changes to the file
:x - saves and exits out of a file

:/sometext - vim will try to find sometext inside the file
n - will go to the next occurance of sometext
N - will go to the previous occurance of sometext

:5 - this will go to line 5 in the file

dd - will delete the current line where cursor is located
5dd - will delete 5 lines after and including the line where the cursor is

u - this will undo the last change made

shift+g - takes you to the bottom of a file
```

Command	Action
:w	Writes changes into the file without quitting vim
:w file2	Writes changes into a new file called *file2* without quitting vim
:w!	Writes changes to the file even if the file owner does not have write per on the file
:wq	Writes changes to the file and quits vim
:wq!	Writes changes to the file and quits vim even if the file owner does not write permission on the file
:q	Quits vim if no modifications were made
:q!	Quits vim if modifications were made, but we do not wish to save them

Touch - to create a file

 cat - can be used to create short file eg : cat > Abhi.py (create a file named abhi)

vim - can be used to create file

The touch command has a few interesting options. The -d and -t options set a
specific date and time on a file; the -a and -m options enable you to change
only the access or the modification time on a file to the current system time;

Copying Files and Directories

Command - cp
The cp command copies one or more files within a directory or to another directory. To duplicate a file in the same directory, you must give a different name to
the target file. However, if the copy is being made to a different directory, you can use either the same filename or assign a different one.

Directories

The cp command with the -r (recursive) option copies an entire directory tree to another location

The cp command can also use -p, which can provide the ability to preserve the attributes (timestamp, permissions, ownership, etc.) of a file or directory
being copied

Moving and Renaming Files and Directories

Moving and Renaming Files , The mv command is used to move or rename files. The -i option can be specified for user confirmation if a file by that name
already exists.

Removing Files

You can remove a file using the rm command, which deletes one or more specified files

Removing Directories

The rmdir and rm commands erase directories. The rmdir command is used to delete empty directories, while rm requires the -d option to accomplish the
same. In addition, the -r or -R (recursive) flag with rm will remove a directory and all of its contents. Both commands support the -v switch for reporting what
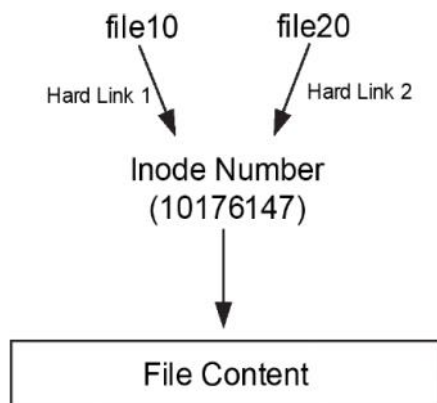they are doing.

File Linking

Soft Link

- ➢ Hard lInk
- ➢ Soft Link

Hard Link

2 files has same metadata and can't be retrieve / identify

Differences between Copying and Linking

Copying	Linking
Creates a duplicate of the source file. If either file is modified, the other file will remain intact.	Creates a shortcut that points to the source file. The source can be accessed or modified using either the source file or the link.
Each copied file stores its own data at a unique location.	All linked files point to the same data
Each copied file has a unique inode number with its unique metadata.	**Hard Link:** All hard-linked files share the same inode number, and hence the metadata. **Symlink:** Each symlinked file has a unique inode number, but the inode number store only the pathname to the source.
If a copy is moved, erased, or renamed, the source file will have no impact, and vice versa.	**Hard Link:** If the hard link is weeded out, the other file and the data will remain untouched. **Symlink:** If the source is deleted, the soft link will b broken and become meaningless. If the soft link is removed, the source will have no impact.
Copy is used when the data needs to be edited independent of the other.	Links are used when access to the same source is required from multip locations.
Permissions on the source and the copy are managed independent of each other.	Permissions are managed on the source file.

Important points to remember

1. 3 commands to identify the file type

   ➢ ls
   ➢ file
   ➢ stat

2. The last line mode (extended mode) allows users to copy or move lines

3. Kernel uses the major number to identify the device and minor number to identify the uniqueness of a device with in a particular device category

4. How to find a replace a word on vim

How to Find and Replace in Vim

:%s/"keyword"/"replace word"/g