SIMPLE LINUX COMMANDS

1. Some basic commands (like cd, ls, file, touch, etc)
2. Alias command
3. Different fields after ls -l command execution
4. How to change file permission
5. Soft link and Hard link
6. Combining multiple commands
7. File descriptors
8. Pipe and tee operator

SIMPLE LINUX COMMANDS

Shortcut to open terminal= <ctrl>+<alt>+T

Commands-

- ☐ clear(ctrl+<L>)
- ☐ exit(ctrl+<D>)
- ☐ pwd- displays present working directory
- ☐ ls-listing
- ☐ ps- to which shell we are interacting with
- ☐ uname- OS, kernel, hardware architecture
- ☐ cd-change directory
- ☐ cd ..-go to parent directory
- ☐ cd ~-go to home
- ☐ cal- opens calander
- ☐ ncal-opens calendar in different orientation
- ☐ date-displays today's date
- ☐ man-opens user's manual for a command
- ☐ groups-gives the names of groups current user belongs to
- ☐ ls -l-longlisting of all files and folder in the current directory
- ☐ ls -li-longlisting of all files and folder in the current directory with inode number
- ☐ mkdir-to make directory
- ☐ rmdir-to remove directory
- ☐ touch- to make file or change file timestamp
- ☐ rm- to remove file
- ☐ gedit-to opens editor to edit file
- ☐ cp- copy file or directory
- ☐ mv-to rename file and to move file
- ☐ less-shows file content on one screen at a time
- ☐ cat-To display file content
- ☐ More- to read file page by page
- ☐ head -n 10 file_name- to display top 10 lines of file
- ☐ tail -n 10 file_name-to display last 10 lines of file
- ☐ wc -l file- count the number of lines in file
- ☐ stat-file size details
- ☐ which- where the command is located
- ☐ whatis- brief description of the commands
- ☐ type- what type of command is
- ☐ free-give the memory details
- ☐ file-type of file

Options

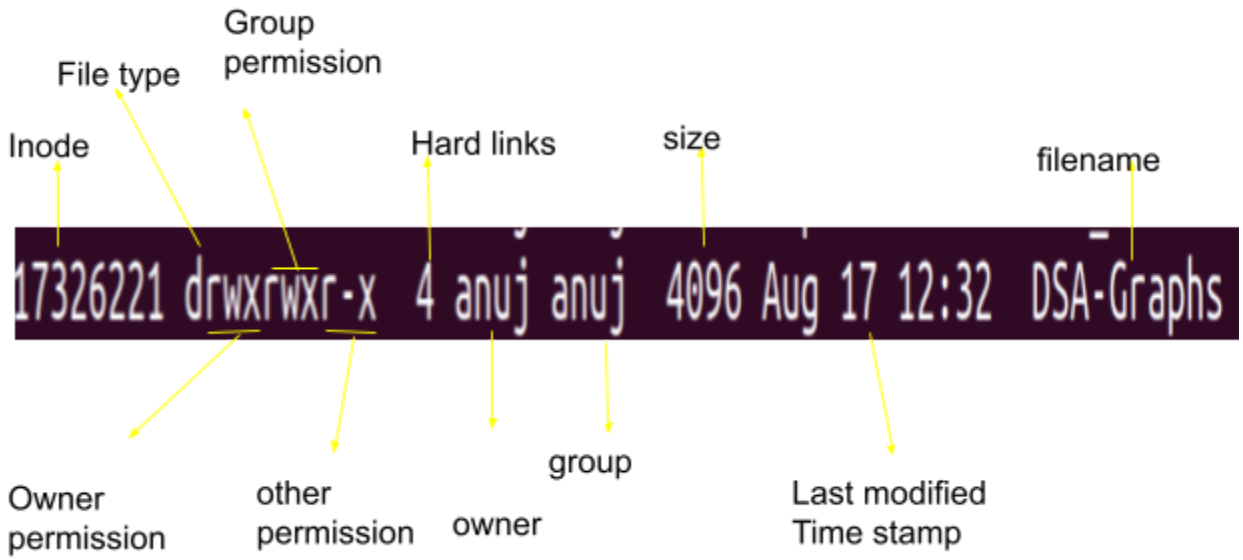
- ☐ -a(enables all hidden files to appear)
- ☐ -l(enable longlisting format)
- ☐ -h(enable text to appear in human readable format)

  * options can be given into any order

Alias/unalias command

alias ll= 'ls -l'
unalias ll

ls -li



First character after inode

-: Regular file
d: directory
l: symbolic link
c: character file
b: block file
s: socket file
p: named pipe

What is inode?

An inode is an index node. It serves as an unique identifier for a specific piece of metadata on a given file system

Changing permission of file
u-user
g-group
o-others

Chmod g-w data

rwx	octal
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

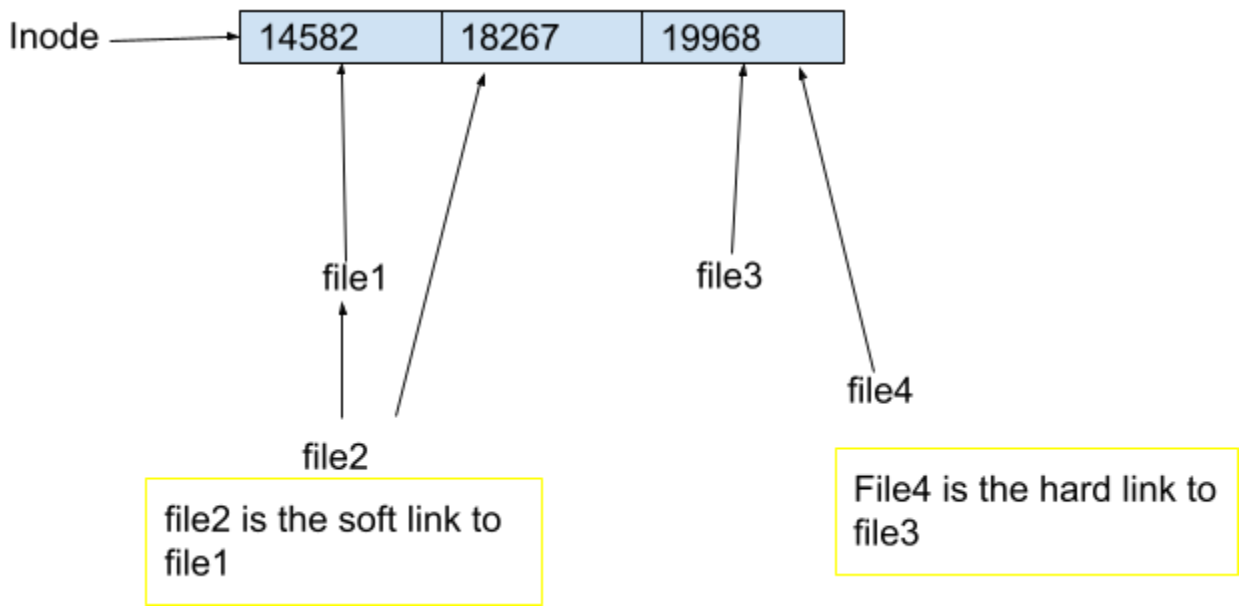
Links- soft and hard link

Soft link is a pointer to original file, just like icon in windows, with different inode number. If the original file is deleted in this case then the soft link will become useless!

ln -s file1 file2

Hardlink is a pointer to the same inode number, which means a different name of the same file. If the original link of the file is deleted , then the hard link will still contain the data that was in the original file.

ln file1 file2



Combining Commands:

Executing Complex Commands:
Command1;command2;command3;
Eg: ls -l ; date ; wc -l file 1.txt

Executing command only if the previous command is executed successfully
Command1 && command2
Eg wc -l file1.xt && date

Not executing command if the previous command is executed successfully
Command1 || command2
Eg: wc -l file1.txt|date

File Descriptors:

Every command in linux has three file descriptor, they are:

stdin- This is referred to as the *standard input* and the associated file descriptor is 0. Pointer to the stream which is coming from the keyboard as a user input

Stdout & stderr - This is referred to as the *standard output and standard error respectively* and the associated file descriptor is 1 and 2 respectively. Pointer to the stream where display of the output is made

Redirect operator(>):
In case if we want to output something to the file then in this case '>(redirect operator)' can be used.
This redirect operator '>' shifts the stdout pointer to the file1
Eg: ls -l > file1

In case, there is no file1 this command will create a new file with a name file1. If the file1 has already written content then the content of the file will be overwritten by the command output.

If we need to append the content to the given file, then '>>' operator will be used.

Let's see some more examples
Eg: cat > file
Whatever we write to the command line will be written to the file, after writing to the command line one can exit by pressing <ctrl>+d. If the file doesn't exist, then a new file will be created with the given name.

Error redirection

2>: redirect the stderr to the file1

Command > file1 2> file2
Output of he command will be output to the file1
If the error occurs while executing command then error is written to the file2

If we want to log both output and error to the same file, then
Command > file1 2> &1

In case we want to discard the error then sink (/dev/null) can be used:
Command > file1 2> /dev/null

|tee operator: will write the output of the command in the given file and display it onto the terminal

Pipe operator: this allows us to take the output of one command as the input of the another command.
Eg: ls -l > file1

Wc -l file
same

/ (forward slash) is used in Linux as the name of the root directory and to separate directory from sub directories.
*Root folder is the parent folder of itself