Linux Commands

alias command

alias command instructs the shell to replace one string with another string while executing the commands.

When we often have to use a single big command multiple times, in those cases, we create something called an alias for that command. Alias is like a shortcut command which will have the same functionality as if we are writing the whole command.

Creating an alias:

```
vegimohith@vegimohith-Inspiron-3542:~$ alias CD="cd Desktop"
vegimohith@vegimohith-Inspiron-3542:~$ CD
vegimohith@vegimohith-Inspiron-3542:~/Desktop$
```

Here CD will have same functionality as cd Desktop and move to Desktop directory

Options for alias command:

- alias -p: This option prints all the defined aliases in a reusable format.

```
vegimohith@vegimohith-Inspiron-3542:~/Desktop$ alias -p
alias CD='cd Desktop'
alias alert='notify-send --urgency=low -i "$([ $? = 0 ] && echo terminal || echo
 error)" "$(history|tail -n1|sed -e '\''s/^\s*[0-9]\+\s*//;s/[;&|]\s*alert$//'\'
')"'
alias egrep='egrep --color=auto'
alias fgrep='fgrep --color=auto'
alias grep='grep --color=auto'
alias l='ls -CF'
alias la='ls -A'
alias ll='ls -alF'
alias ls='ls --color=auto'
vegimohith@vegimohith-Inspiron-3542:~/Desktop$
```

- alias –help : It displays help Information.

```
vegimohith@vegimohith-Inspiron-3542:~/Desktop$ alias --help
alias: alias [-p] [name[=value] ... ]
    Define or display aliases.

    Without arguments, `alias' prints the list of aliases in the reusable
    form `alias NAME=VALUE' on standard output.

    Otherwise, an alias is defined for each NAME whose VALUE is given.
    A trailing space in VALUE causes the next word to be checked for
    alias substitution when the alias is expanded.

    Options:
      -p        print all defined aliases in a reusable format

    Exit Status:
    alias returns true unless a NAME is supplied for which no alias has been
    defined.
vegimohith@vegimohith-Inspiron-3542:~/Desktop$
```

Removing an alias:

```
vegimohith@vegimohith-Inspiron-3542:~/Desktop$ unalias CD
vegimohith@vegimohith-Inspiron-3542:~/Desktop$ CD
CD: command not found
vegimohith@vegimohith-Inspiron-3542:~/Desktop$
```

Here I am unaliasing CD

The problem with that alias is that it will only be available for your current terminal session.
If you open a new terminal session, the alias will no longer be available. If you wish to save your aliases across sessions you will need a permanent alias.

Creating permanent alias:
To keep aliases between sessions, you can save them in your user's shell configuration profile file. This can be:

- Bash – ~/.bashrc
- ZSH – ~/.zshrc
- Fish – ~/.config/fish/config.fish

---

file command:
used to determine the type of a file.

Displaying the file type:

```
vegimohith@vegimohith-Inspiron-3542:~$ file 1.cpp
1.cpp: C++ source, ASCII text
vegimohith@vegimohith-Inspiron-3542:~$ file 1.txt
1.txt: C++ source, ASCII text
vegimohith@vegimohith-Inspiron-3542:~$ file 3.py
3.py: ASCII text
vegimohith@vegimohith-Inspiron-3542:~$ file a.out
a.out: ELF 64-bit LSB pie executable, x86-64, version 1 (SYSV), dynamically link
ed, interpreter /lib64/ld-linux-x86-64.so.2, BuildID[sha1]=09f3d4f546a6d260b8a1d
9fef1bdf5f443ff7a94, for GNU/Linux 3.2.0, not stripped
vegimohith@vegimohith-Inspiron-3542:~$
```

Options for file command:

- **-b, –brief :** This is used to display just file type in brief mode.
- *** option :** Command displays the all files's file type.

- **directoryname/* option :** This is used to display all files, file types in a particular directory.
- **[range]* option:** To display the file type of files in a specific range.
- **-c option:** Cause a checking printout of the parsed form of the magic file. This option is usually used in conjunction with the *-m* flag to debug a new magic file before installing it.
- **-f option:** Read the names of the files to be examined from the namefile (one per line) before the argument list. Either namefile or atleast one filename argument must be present; to test the standard input, use '-' as a filename argument.
- **-F option :** File and file type are separated by *:*. But we can change separator using -F option
- **-i option:** To view mime types of files.
- **-N option:** Don't pad file names so that they align in the output.
- **-s option:** For special files
- **filenames:** Displays file types of multiple files
- **-z option:** Try to look inside compressed files
- **–help option:** Print a help message and exit.

---

md5sum command:

The md5sum command prints a 32-character (128-bit) checksum of the given file, using the MD5 algorithm.

MD5 Algorithm is used as checksum to verify data integrity
Suppose, anyone wants to install an operating system , so to verify if it's the correct CD, it's always a good idea to verify the .iso file using MD5 checksum, so that you don't end up installing the wrong software.

md5sum is **used to verify the integrity of files**, as virtually any change to a file will cause its MD5 hash to change. Most commonly, md5sum is used to verify that a file has not changed as a result of a faulty file transfer, a disk error or non-malicious meddling.

Display the hash value:

```
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ md5sum file1.txt
cb153111b4a4a9bd7e07027542508597  file1.txt
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ md5sum file2.txt
6c1b74adb1397d76576a404c45a096a2  file2.txt
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ md5sum file3.txt
7b3c88db4ed22d8782db409d3e2cc2f3  file3.txt
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$
```

Validate multiple files:

```
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ md5sum file1.txt file2.txt file3.txt > hashes
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ md5sum --check hashes
file1.txt: OK
file2.txt: OK
file3.txt: OK
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$
```

Here I have validated 3 files and found out that none of the files are modified.

Now, I will modify file3.txt and validate again

```
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ md5sum --check hashes
file1.txt: OK
file2.txt: OK
file3.txt: FAILED
md5sum: WARNING: 1 computed checksum did NOT match
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$
```

We can see that file3.txt has changed

Display only modified files:

```
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ md5sum --quiet --check hashes
file3.txt: FAILED
md5sum: WARNING: 1 computed checksum did NOT match
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$
```

Here only file3.txt was changed

---

head command:

The head command, as the name implies, prints 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.

Printing using head command:

```
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ head state.txt
Andhra Pradesh
Arunachal Pradesh
Assam
Bihar
Chhattisgarh
Goa
Gujarat
Haryana
Himachal Pradesh
Jammu and Kashmir
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$
```

Without any option the head command will print top 10 lines of the file

Options of head command:

- **-n num**: Prints the first 'num' lines instead of the first 10 lines. num is mandatory to be specified in command otherwise it displays an error.

```
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ head -n 5 state.txt
Andhra Pradesh
Arunachal Pradesh
Assam
Bihar
Chhattisgarh
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$
```

- **-c num:** Prints the first 'num' bytes from the file specified. Newline counts as a single character, so if head prints out a newline, it will count it as a byte. num is mandatory to be specified in command otherwise displays an error.

```
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ head -c 5 state.txt
Andhrvegimohith@vegimohith-Inspiron-3542:~/linuxcommands$
```

- **-q:** It is used if more than 1 file is given. Because of this command, data from each file is not preceded by its file name.

Without using -q option:

```
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ head state.txt capital.txt
==> state.txt <==
Andhra Pradesh
Arunachal Pradesh
Assam
Bihar
Chhattisgarh
Goa
Gujarat
Haryana
Himachal Pradesh
Jammu and Kashmir

==> capital.txt <==
Hyderabad
Itanagar
Dispur
Patna
Raipur
Panaji
Gandhinagar
Chandigarh
Shimla
```

With using -q option:

```
Srinagarvegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ head -q state.txt capital.txt
Andhra Pradesh
Arunachal Pradesh
Assam
Bihar
Chhattisgarh
Goa
Gujarat
Haryana
Himachal Pradesh
Jammu and Kashmir
Hyderabad
Itanagar
Dispur
Patna
Raipur
Panaji
Gandhinagar
Chandigarh
Shimla
Srinagarvegimohith@vegimohith-Inspiron-3542:~/linuxcommands$
```

- **-v:** By using this option, data from the specified file is always preceded by its file name.

```
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ head -v state.txt
==> state.txt <==
Andhra Pradesh
Arunachal Pradesh
Assam
Bihar
Chhattisgarh
Goa
Gujarat
Haryana
Himachal Pradesh
Jammu and Kashmir
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$
```

Print lines between M and N lines:

```
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ head -n 20 state.txt | tail -10
Jharkhand
Karnataka
Kerala
Madhya Pradesh
Maharashtra
Manipur
Meghalaya
Mizoram
Nagaland
Odisha
```

Print the lines between 10 and 20

---

bg command:

bg command in linux is used to place foreground jobs in the background.
This command is used to put the mentioned job in background

```
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ jobs
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ sleep 50
^Z
[1]+  Stopped                 sleep 50
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ jobs
[1]+  Stopped                 sleep 50
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ bg %1
[1]+ sleep 50 &
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ jobs
[1]+  Running                 sleep 50 &
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$
```

---

fc command:

The fc command is used to list, edit or re-execute the commands previously entered into an interactive shell.

---

mkdir:

mkdir command in Linux allows the user to create directories (also referred to as folders in some operating systems). This command can create multiple directories at once as well as set the permissions for the directories. It is important to note that the user executing this command must have enough permissions to create a directory in the parent directory, or he/she may receive a ‘permission denied’ error.

Creating new directory:

```
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ ls
capital.txt  file1.txt  file2.txt  file3.txt  hashes  state.txt
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ mkdir dir
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ ls
capital.txt  dir  file1.txt  file2.txt  file3.txt  hashes  state.txt
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$
```

Initially there is no “dir” directory, by using mkdir I have created “dir” directory

Creating multiple directories at a time:

```
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ ls
capital.txt  dir  file1.txt  file2.txt  file3.txt  hashes  state.txt
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ mkdir 1 2 3
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ ls
1  2  3  capital.txt  dir  file1.txt  file2.txt  file3.txt  hashes  state.txt
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$
```

Options of mkdir command:

- **–version:** It displays the version number, some information regarding the license and exits.
- **–help**: It displays the help related information and exits.
- **-v or –verbose**: It displays a message for every directory created.
- **-p**: A flag which enables the command to create parent directories as necessary. If the directories exist, no error is specified.

Creating sub directories:

```
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ mkdir -p first/second/third
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ ls
1  2  3  capital.txt  dir  file1.txt  file2.txt  file3.txt  first  hashes  state.txt
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ cd first
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands/first$ ls
second
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands/first$ cd second
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands/first/second$ ls
third
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands/first/second$
```

- **-m**: This option is used to set the file modes, i.e. permissions, etc. for the created directories.

---

tail command:

The tail command, as the name implies, prints 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 preceded by its file name.

Printing using tail command:

```
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ tail state.txt
Odisha
Punjab
Rajasthan
Sikkim
Tamil Nadu
Telangana
Tripura
Uttar Pradesh
Uttarakhand
West Bengalvegimohith@vegimohith-Inspiron-3542:~/linuxcommands$
```

Options of tail command:

- **-n num:** Prints the last 'num' lines instead of the last 10 lines. num is mandatory to be specified in command otherwise it displays an error. This command can also be written as without symbolizing 'n' character but '-' sign is mandatory.

```
West Bengalvegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ tail -n 3 state.txt
Uttar Pradesh
Uttarakhand
West Bengalvegimohith@vegimohith-Inspiron-3542:~/linuxcommands$
```

- **-c num:** Prints the last 'num' bytes from the file specified. Newline counts as a single character, so if tail prints out a newline, it will count it as a byte. In this option it is mandatory to write -c followed by positive or negative num depending upon the requirement. By +num, it displays all the data after skipping num bytes from the starting of the specified file and by -num, it displays the last num bytes from the file specified.
- **-q:** It is used if more than 1 file is given. Because of this command, data from each file is not preceded by its file name.
- **-f:** This option is mainly used by system administration to monitor the growth of the log files written by many Unix programs as they are running. This option shows the last ten lines of a file and will update when new lines are added. As new lines are written to the log, the console will update with the new lines. The prompt doesn't return even after work is over, so we have to use the interrupt key to abort this command. In general, the application writes error messages to log files. You can use the -f option to check for the error messages as and when they appear in the log file.
- **-v:** By using this option, data from the specified file is always preceded by its file name.

```
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ tail -v state.txt
==> state.txt <==
Odisha
Punjab
Rajasthan
Sikkim
Tamil Nadu
Telangana
Tripura
Uttar Pradesh
Uttarakhand
West Bengalvegimohith@vegimohith-Inspiron-3542:~/linuxcommands$
```

- **–version:** This option is used to display the version of tail which is currently running on your system.

```
West Bengalvegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ tail --version
tail (GNU coreutils) 8.32
Copyright (C) 2020 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <https://gnu.org/licenses/gpl.html>.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Written by Paul Rubin, David MacKenzie, Ian Lance Taylor,
and Jim Meyering.
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$
```

---

cat command:

Cat(concatenate) command is very frequently used in Linux. It reads data from the file and gives their content as output. It helps us to create, view, and concatenate files. So let us see some frequently used cat commands.

To show content of given file:

```
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ cat file1.txt
hi
hello
how are you
thanks.vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$
```

To show content of multiple files:

```
thanks.vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ cat file1.txt file2.txt
hi
hello
how are you
thanks.hi
hello to you
I am fine
Your welcome!vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$
```

View content of file with line number:

```
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ cat -n file1.txt
     1  hi
     2  hello
     3  how are you
     4  thanks.vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$
```

Create a new file:
Syntax: cat > newfile

Copy contents of one file to another file:

```
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ cat file1.txt > file2.txt
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$
```

here , file1.txt is the source file and file2.txt is the destination file.

Suppress repeated empty lines in output:
Syntax: cat -s file.txt

Append content of one file to end of another file:
Syntax: cat file1 >> file2

Display content in reverse order:

```
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ cat file1.txt > file2.txt
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ tac file1.txt
thanks.how are you
hello
hi
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ cat file1.txt
hi
hello
how are you
thanks.vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$
```

Highlight the end of the line:

```
thanks.vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ cat -E "file1.txt"
hi$
hello$
how are you$
thanks.vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$
```

To display the dashed files:
Syntax: cat -- "-dashfile"

To display lot of content which can’t fit in terminal:
Syntax: cat "filename" | more

To merge the contents of multiple files:
Syntax: cat "filename1" "filename2" "filename3" > "merged_filename"

To display the contents of all text files:
Syntax: cat *.txt

To write in an existing file:
Syntax: cat >> geeks.txt

---

find command:
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.

```
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ find 1.txt
1.txt
[1]+  Done                    sleep 50
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ find 2.txt
2.txt
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ find 5.txt
find: ‘5.txt’: No such file or directory
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$
```

Options of find command :

- -exec CMD: The file being searched which meets the above criteria and returns 0 for as its exit status for successful command execution.
- -ok CMD : It works the same as -exec except the user is prompted first.
- -inum N : Search for files with inode number ‘N’.
- -links N : Search for files with ‘N’ links.
- -name demo : Search for files that are specified by ‘demo’.
- -newer file : Search for files that were modified/created after ‘file’.
- -perm octal : Search for the file if permission is ‘octal’.
- -print : Display the path name of the files found by using the rest of the criteria.
- -empty : Search for empty files and directories.
- -size +N/-N : Search for files of ‘N’ blocks; ‘N’ followed by ‘c’can be used to measure size in characters; ‘+N’ means size > ‘N’ blocks and ‘-N’ means size < 'N' blocks.
- -user name : Search for files owned by user name or ID ‘name’.
- \(expr \) : True if ‘expr’ is true; used for grouping criteria combined with OR or AND.
- ! expr : True if ‘expr’ is false.

---

more command:

Similar to cat command, more command also displays content of a file, in case of larger files, 'cat' command output will scroll off your screen while 'more' command displays output one screenful at a time.
cat command will dump the entire content of a file on the screen whereas more commands will display content that would fit your screen and you can press enter to see the rest of the content line by line.

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 the next page.
- b key: To go to the backward page.
- / key: Lets you search the string.

Syntax: more <filename>

Options of more command:

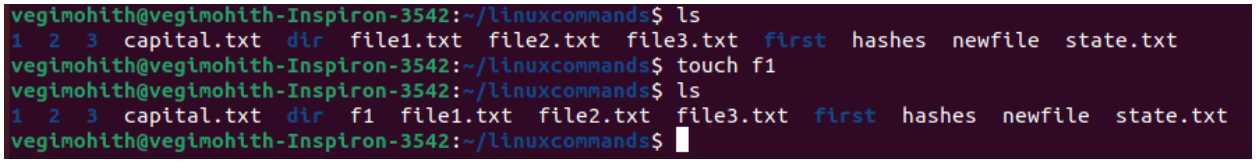
- more -num : limits no of lines to be displayed per page
- more -d: displays user message at right corner
- more -s: squeeze blank lines
- more +num: used to display content from specific line
- more +/string name: helps to find string

---

touch command:

The touch command is a standard command used in the UNIX/Linux operating system which is used to create, change and modify timestamps of a file.
It is used to create a file without any content. The file created using touch command is empty. This command can be used when the user doesn't have data to store at the time of file creation.

Create new file:



Create multiple files at a time:

```
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ ls
1  2  3  capital.txt  dir  f1  file1.txt  file2.txt  file3.txt  first  hashes  newfile  state.txt
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ touch f2 f3 f4
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ ls
1  2  3  capital.txt  dir  f1  f2  f3  f4  file1.txt  file2.txt  file3.txt  first  hashes  newfile  state.txt
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$
```

Options of touch command:

- **touch -a:** This command is used to change access time only. To change or update the last access or modification times of a file touch -a command is used.

```
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ touch -a f1
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ ll
total 52
drwxrwxr-x  7 vegimohith vegimohith 4096 Dec 22 22:39 ./
drwxr-x--- 31 vegimohith vegimohith 4096 Dec 22 17:06 ../
drwxrwxr-x  2 vegimohith vegimohith 4096 Dec 22 18:55 1/
drwxrwxr-x  2 vegimohith vegimohith 4096 Dec 22 18:55 2/
drwxrwxr-x  2 vegimohith vegimohith 4096 Dec 22 18:55 3/
-rw-rw-r--  1 vegimohith vegimohith   84 Dec 22 18:13 capital.txt
drwxrwxr-x  2 vegimohith vegimohith 4096 Dec 22 18:46 dir/
-rw-rw-r--  1 vegimohith vegimohith    0 Dec 22 22:38 f1
-rw-rw-r--  1 vegimohith vegimohith    0 Dec 22 22:39 f2
-rw-rw-r--  1 vegimohith vegimohith    0 Dec 22 22:39 f3
-rw-rw-r--  1 vegimohith vegimohith    0 Dec 22 22:39 f4
-rw-rw-r--  1 vegimohith vegimohith   28 Dec 22 17:59 file1.txt
-rw-rw-r--  1 vegimohith vegimohith   28 Dec 22 20:01 file2.txt
-rw-rw-r--  1 vegimohith vegimohith   41 Dec 22 17:38 file3.txt
drwxrwxr-x  3 vegimohith vegimohith 4096 Dec 22 19:02 first/
-rw-rw-r--  1 vegimohith vegimohith  132 Dec 22 17:09 hashes
-rw-rw-r--  1 vegimohith vegimohith    0 Dec 22 19:59 newfile
-rw-rw-r--  1 vegimohith vegimohith  299 Dec 22 17:59 state.txt
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$
```

```
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ stat f1
  File: f1
  Size: 0               Blocks: 0          IO Block: 4096   regular empty file
Device: 805h/2053d      Inode: 17694911    Links: 1
Access: (0664/-rw-rw-r--)  Uid: ( 1000/vegimohith)   Gid: ( 1000/vegimohith)
Access: 2021-12-22 22:47:08.045208641 +0530
Modify: 2021-12-22 22:38:07.508958359 +0530
Change: 2021-12-22 22:47:08.045208641 +0530
 Birth: 2021-12-22 22:38:07.508958359 +0530
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$
```

Here access time is changed

- **touch -c:** This command is used to check whether a file is created or not. If not created then don't create it. This command avoids creating files.
- **touch -c-d:** This is used to update access and modification time.
- **touch -m:** This is used to change the modification time only. It only updates last modification time.

```
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ touch -m f2
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ ll f2
-rw-rw-r-- 1 vegimohith vegimohith 0 Dec 22 22:54 f2
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$
```

here , modification time is changed

- **touch -r:** This command is used to use the timestamp of another file.

```
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ touch -r f2 f3
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ stat f2
  File: f2
  Size: 0               Blocks: 0          IO Block: 4096   regular empty file
Device: 805h/2053d      Inode: 17694914    Links: 1
Access: (0664/-rw-rw-r--)  Uid: ( 1000/vegimohith)   Gid: ( 1000/vegimohith)
Access: 2021-12-22 22:39:57.160017879 +0530
Modify: 2021-12-22 22:54:30.053574494 +0530
Change: 2021-12-22 22:54:30.053574494 +0530
 Birth: 2021-12-22 22:39:57.160017879 +0530
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ stat f3
  File: f3
  Size: 0               Blocks: 0          IO Block: 4096   regular empty file
Device: 805h/2053d      Inode: 17694915    Links: 1
Access: (0664/-rw-rw-r--)  Uid: ( 1000/vegimohith)   Gid: ( 1000/vegimohith)
Access: 2021-12-22 22:39:57.160017879 +0530
Modify: 2021-12-22 22:54:30.053574494 +0530
Change: 2021-12-22 22:56:20.625707919 +0530
 Birth: 2021-12-22 22:39:57.160017879 +0530
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$
```

Here, f3 is updated with the timestamp of f2.

- **touch -t** : This is used to create a file using a specified time.

```
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ touch -t 202103221212 fileName
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ stat fileName
  File: fileName
  Size: 0               Blocks: 0          IO Block: 4096   regular empty file
Device: 805h/2053d      Inode: 17694917    Links: 1
Access: (0664/-rw-rw-r--)  Uid: ( 1000/vegimohith)   Gid: ( 1000/vegimohith)
Access: 2021-03-22 12:12:00.000000000 +0530
Modify: 2021-03-22 12:12:00.000000000 +0530
Change: 2021-12-22 23:00:33.988069873 +0530
 Birth: 2021-12-22 23:00:33.988069873 +0530
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$
```

Here, the file is created with the given time.

---

chmod command:

The chmod command is used to change the access mode of a file.
The name is an abbreviation of change mode.

```
vegimohith@vegimohith-Inspiron-3542:~/Ignitarium/shell$ chmod +x 8.sh
vegimohith@vegimohith-Inspiron-3542:~/Ignitarium/shell$ ./8.sh
Enter the value of n:
5
Enter the numbers:
1
2
3
4
5
15
vegimohith@vegimohith-Inspiron-3542:~/Ignitarium/shell$
```

Here, x mode is used to grant permission to execute the file.

To grant permission to read the file we use r mode, to write the file we use w mode.

Operator	Description
+	Adds the specified modes to the specified classes
-	Removes the specified modes from the specified classes
=	The modes specified are to be made the exact modes for the specified classes

Mode	Description
r	Permission to read the file.
w	Permission to write (or delete) the file.
x	Permission to execute the file, or, in the case of a directory, search it.

---

sudo command:

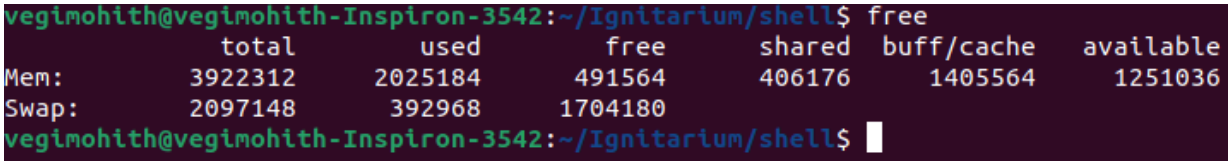
super user do

sudo command will elevate privileges , allow a user with proper permissions to execute a command as another user, such as the superuser.

---

free command:

displays the total amount of free space available along with the amount of memory used and swap memory in the system, and also the buffers used by the kernel.



Total - total memory space of the system

Used - amount of memory used and swapped

Free - free space available

---

echo command:

echo command in linux is used to display lines of text/string that are passed as an argument . This is a built-in command that is mostly used in shell scripts and batch files to output status text to the screen or a file.

Displaying text:

```
vegimohith@vegimohith-Inspiron-3542:~/Ignitarium/shell$ echo "displaying a text"
displaying a text
vegimohith@vegimohith-Inspiron-3542:~/Ignitarium/shell$
```

Options of echo command:

- \b : it removes all the spaces in between the text.
- \c : suppress trailing newline with backspace interpreter ‘-e‘ to continue without emitting new line.
- \n : this option creates a new line from where it is used.
- \t : this option is used to create horizontal tab spaces.
- \r : carriage return with backspace interpreter ‘-e‘ to have specified carriage return in output.
- \v : this option is used to create vertical tab spaces.
- \a : alert return with backspace interpreter ‘-e‘ to have sound alert.
- echo * : this command will print all files/folders, similar to ls command .

---

top command:

The **top** command is used to show the Linux processes. It provides a dynamic real-time view of the running system. Usually, this command shows the summary information of the system and the list of processes or threads which are currently managed by the Linux Kernel.

As soon as you run this command it will open an interactive command mode where the top half portion will contain the statistics of processes and resource usage. And Lower half contains a list of the currently running processes.

---

sort command:

Sort a file based on ASCII code

```
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ cat file.txt
abhishek
chitransh
satish
rajan
naveen
divyam
harshvegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ sort file.txt
abhishek
chitransh
divyam
harsh
naveen
rajan
satish
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$
```

---

ifconfig command:

used for displaying current network configuration information, setting up an ip address, netmask, or broadcast address to a network interface, creating an alias for the network interface, setting up hardware address, and enable or disable network interfaces.

```
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ ifconfig
enp7s0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 20:47:47:27:02:27  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 77971  bytes 8163738 (8.1 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 77971  bytes 8163738 (8.1 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlp6s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.29.43  netmask 255.255.255.0  broadcast 192.168.29.255
        inet6 2405:201:c01b:1193:47da:2f04:997d:a855  prefixlen 64  scopeid 0x0<global>
        inet6 fe80::d5a4:106c:bc7:bb25  prefixlen 64  scopeid 0x20<link>
        inet6 2405:201:c01b:1193:6db5:28cf:a25a:aaa8  prefixlen 64  scopeid 0x0<global>
        ether 60:6d:c7:a3:c6:4b  txqueuelen 1000  (Ethernet)
        RX packets 1190517  bytes 742493024 (742.4 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1001730  bytes 362194488 (362.1 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$
```

---

join command:

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.

```
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ cat 1.txt
1 AAYUSH
2 APAAR
3 HEMANT
4 KARTIKvegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ cat 2.txt
1 101
2 102
3 103
4 104vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ join 1.txt 2.txt
1 AAYUSH 101
2 APAAR 102
3 HEMANT 103
4 KARTIK 104
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$
```

---

kill command:

Used to terminate the process manually

kill command sends a signal to a process which terminates the process. If the user doesn't specify any signal which is to be sent along with the kill command then a default TERM signal is sent that terminates the process.

To kill the process with pid:

```
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ ps
    PID TTY          TIME CMD
  32087 pts/0    00:00:00 bash
  32102 pts/0    00:00:07 sublime_text
  32119 pts/0    00:00:00 plugin_host-3.3
  32122 pts/0    00:00:00 plugin_host-3.8
  36060 pts/0    00:00:00 top
  40318 pts/0    00:00:00 ps
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ kill 40318
bash: kill: (40318) - No such process
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$
```

---

ps command:

ps command is used to list the currently running processes and their PIDs along with some other information depending on different options.

```
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ ps
    PID TTY          TIME CMD
  32087 pts/0    00:00:00 bash
  32102 pts/0    00:00:07 sublime_text
  32119 pts/0    00:00:00 plugin_host-3.3
  32122 pts/0    00:00:00 plugin_host-3.8
  36060 pts/0    00:00:00 top
  40318 pts/0    00:00:00 ps
```

Options of ps command:

- View all the current processes. ps -A or ps -e
- View processes not associated with the terminal. ps -a
- View all processes except session leaders. ps -d
- View all processes associated with the terminal. ps -T
- View all the running processes. ps -r
- View all processes owned by you. ps -x

---

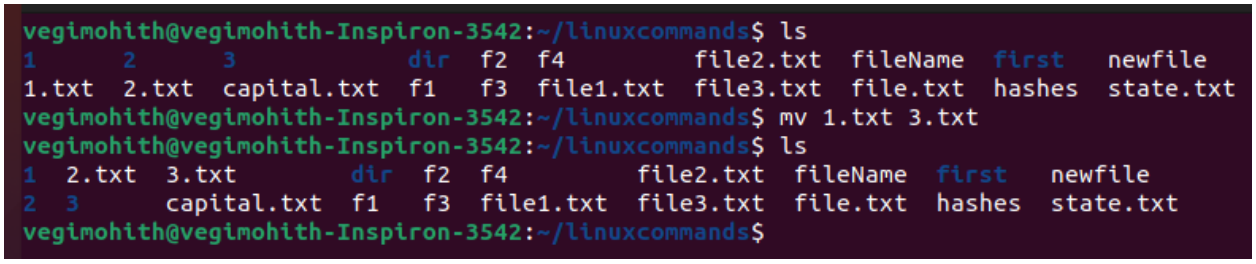
mv command:

mv stands for move. mv is used to move one or more files or directories from one place to another in a file system like UNIX. It has two distinct functions:

(i) It renames a file or folder.

(ii) It moves a group of files to a different directory.

No additional space is consumed on a disk during renaming. This command normally works silently and means no prompt for confirmation.

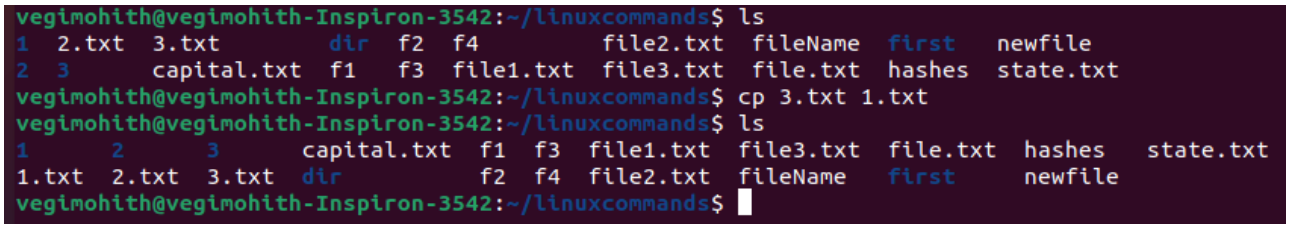


Here , 1.txt is renamed as 3.txt

---

cp command:

cp stands for copy. This command is used to copy files or groups of files or directory. It creates an exact image of a file on a disk with a different file name. cp command requires at least two filenames in its arguments.



Here, 3.txt is copied into 1.txt

---

history command:

Used to display the previously executed commands

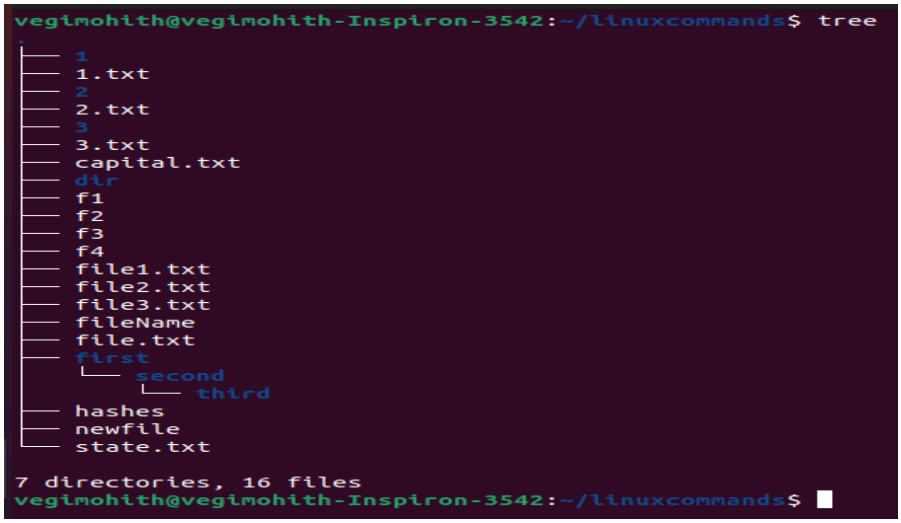
To limit the number of commands to be shown

history -n here n denotes no of commands to be shown

---

tree command:

tree is a recursive directory listing program that produces a depth-indented listing of files. With no arguments, the tree lists the files in the current directory. When directory arguments are given, the tree lists all the files or directories found in the given directories each in turn. Upon completion of listing all files and directories found, the tree returns the total number of files and directories listed.



Options of tree command:

–help : Outputs a verbose usage listing.

–version : Outputs the version of the tree.

-a : All files are printed. By default, the tree does not print hidden files (those beginning with a dot `.'). In no event does the tree print the file system constructs `.' (current directory) and `..' (previous directory).

-d : List directories only.

-f : Prints the full path prefix for each file.

-i : Tree will not print the indentation lines. Useful when used in conjunction with the -f option.

-l : Follows symbolic links to directories as if they were directories. Links that would result in a recursive loop are avoided.

-x : Stay on the current file system only, as with find -xdev.

-P pattern : List only those files that match the wild-card pattern.

Note : you must use the -a option to also consider those files beginning with a dot `.' for matching. Valid wildcard operators are `*' (any zero or more characters), `?' (any single character), `[…]' (any single character listed between brackets (optional – (dash) for character range may be used: ex: [A-Z]), and `[^…]' (any single character not listed in brackets) and `|' separates alternate patterns.

-I pattern : Do not list those files that match the wild-card pattern.

–prune : Makes tree prune empty directories from the output, useful when used in conjunction with -P or -I.

–filelimit # : Do not descend directories that contain more than # entries.

–timefmt format : Prints (implies -D) and formats the date according to the format string which uses the strftime syntax.
–noreport : Omits printing of the file and directory report at the end of the tree listing.
-p : Print the protections for each file (as per ls -l).
-s : Print the size of each file along with the name.
-u : Print the username, or UID # if no username is available, of the file.
-g : Print the group name, or GID # if no group name is available, of the file.
-D : Print the date of the last modification time for the file listed.
–inodes : Prints the inode number of the file or directory
–device : Prints the device number to which the file or directory belongs
-F : Append a `/' for directories, a `=' for socket files, a `*' for executable files and a `|' for FIFO's, as per ls -F
-q : Print non-printable characters in file names as question marks instead of the default carrot notation.
-N : Print non-printable characters as is instead of the default carrot notation.
-r : Sort the output in reverse alphabetical order.
-t : Sort the output by last modification time instead of alphabetically.
–dirsfirst : List directories before files.
-n : Turn colorization off always, over-ridden by the -C option.
-C : Turn colorization on always, using built-in color defaults if the LS_COLORS environment variable is not set. Useful to colorize output to a pipe.
-A : Turn on ANSI line graphics hack when printing the indentation lines.
-S : Turn on ASCII line graphics (useful when using linux console mode fonts). This option is now equivalent to `–charset=IBM437′ and will eventually be depreciated.
-L level : Max display depth of the directory tree.
-R : Recursively cross down the tree each level directories (see -L option), and at each of them execute the tree again adding `-o 00Tree.html' as a new option.
-H baseHREF : Turn on HTML output, including HTTP references. Useful for ftp sites. baseHREF gives the base ftp location when using HTML output. That is, the local directory may be `/local/ftp/pub', but it must be referenced as `ftp://host-name.organization.domain/pub' (baseHREF should be `ftp://hostname.organization.domain'). Hint: don't use ANSI lines with this option, and don't give more than one directory in the directory list. If you want to use colors via CSS stylesheet, use the -C option in addition to this option to force color output.
-T title : Sets the title and H1 header string in HTML output mode.
–charset charset : Set the character set to use when outputting HTML and for line drawing.
–nolinks : Turns off hyperlinks in HTML output.
-o file name : Send output to file name.

—----------

vmstat command:

vmstat command in Linux/Unix is a performance monitoring command of the system as it gives the information about processes, memory, paging, block IO, disk, and CPU scheduling.

```
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$ vmstat
procs -----------memory---------- ---swap-- -----io---- -system-- ------cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 0  0 454556 718700 121900 1493416    1    7    35   115  150  398 23  5 70  2  0
vegimohith@vegimohith-Inspiron-3542:~/linuxcommands$
```

---

export command:

It is used to mark variables and functions to be passed to child processes. Basically, a variable will be included in child process environments without affecting other environments.

---

lsusb command:

The lsusb command in Linux is used to display the information about USB buses and the devices connected to them. The properties displayed are speed, BUS, class, type details, etc.

```
ignitarium@IGN-BLR-LP-215:~/git/linuxshell$ lsusb
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 003: ID 04f2:b6d9 Chicony Electronics Co., Ltd Integrated Camera
Bus 001 Device 002: ID 04f3:0c4b Elan Microelectronics Corp. ELAN:Fingerprint
Bus 001 Device 004: ID 8087:0aaa Intel Corp.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
ignitarium@IGN-BLR-LP-215:~/git/linuxshell$
```

---

flush Command:

The flush command in the DM account flushes to disk all memory-resident buffers that are tagged as write-required.

All write-required buffers are periodically flushed to disk in the normal sequence of events, but this command is provided to ensure data integrity at a given moment. In case of an emergency reboot or problems with the system, flush is a safety measure.

---

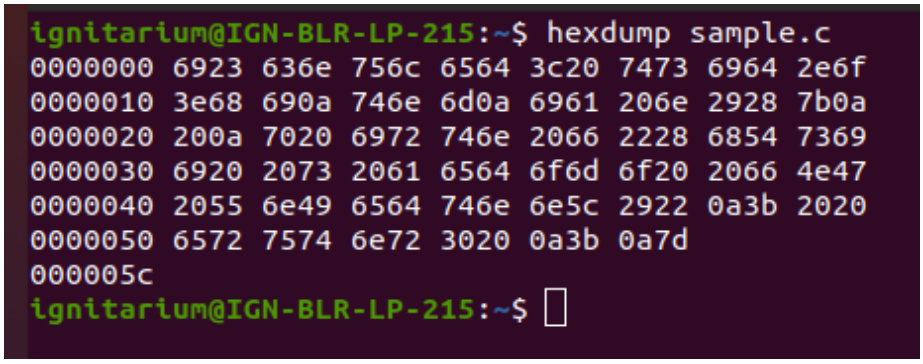
chown command:
The chown command is used to change the file Owner or group. Whenever you want to change ownership you can use the chown command.

Syntax: chown owner_name file_name

To change ownership to root: sudo chown root file_name

---

hexdump command:
hexdump - display file contents in hexadecimal, decimal, octal, or ascii



---

patch command:
patch is a command that takes the output from the diff and puts it into a file. Then, it can take the file output and overwrite another file with the changes.

---

set command:
set is a shell built-in that displays all shell variables, not only the environment ones, and also shell functions, which is what you are seeing at the end of the list.
Variables are displayed with a syntax that allow them to be set when the lines are executed or sourced.

---

printenv command:

Print all or part of environment

printenv prints the values of the specified environment VARIABLE(s). If no VARIABLE is specified, print name and value pairs for them all.

Usage :

printenv [VARIABLE]…

printenv OPTION

---

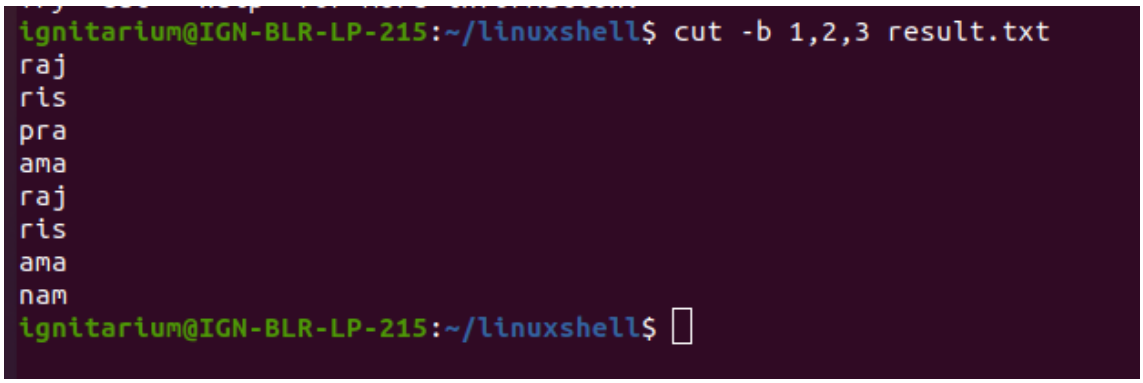
shift command:

- The shift command in UNIX is used to move the command line arguments to one position left. The first argument is lost when you use the shift command.
- Shifting command line arguments is useful when you perform a similar action to all arguments one-by-one, without changing the variable name. The shift command throws away the left-most variable (argument number 1) and reassigns values to the remaining variables.

---

cut command:

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.



---

test command:

The test command is used to check file types and compare values. Test is used in conditional execution. It is used for:

- File attributes comparisons
- Perform string comparisons.
- Basic arithmetic comparisons.

```
ignitarium@IGN-BLR-LP-215:~/linuxshell$ test 5 -gt 2 && echo "yes"
yes
ignitarium@IGN-BLR-LP-215:~/linuxshell$ test 5 -eq 5 && echo yes || echo no
yes
ignitarium@IGN-BLR-LP-215:~/linuxshell$
```

---

dd command:

dd is a command-line utility for Unix and Unix-like operating systems whose primary purpose is to convert and copy files.

---

rm command:

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.

```
ignitarium@IGN-BLR-LP-215:~$ ls
 1.c          Downloads                    fresher_training_plan_2020_21.xlsx   git_getting_started                    meld      Pictures   sample.c~  'Training Exercises'
 Desktop      editing_browsing             git                                  google-chrome-stable_current_amd64.deb Music     Public     snap        Videos
 Documents    fresher_training_plan_2020_21.ods  git_exercise_solutions         linuxshell                             new.txt   sample.c   Templates
ignitarium@IGN-BLR-LP-215:~$ rm new.txt
ignitarium@IGN-BLR-LP-215:~$ ls
 1.c          Downloads                    fresher_training_plan_2020_21.xlsx   git_getting_started                    meld      Public     snap                Videos
 Desktop      editing_browsing             git                                  google-chrome-stable_current_amd64.deb Music     sample.c   Templates
 Documents    fresher_training_plan_2020_21.ods  git_exercise_solutions         linuxshell                             Pictures  sample.c~  'Training Exercises'
ignitarium@IGN-BLR-LP-215:~$
```

Here, the new.txt file is removed.

---

pushd command:

pushd is a shell built-in command which allows us to easily manipulate the directory stack. This appends a directory to the top of the directory stack, or rotates the stack, making the new top of the stack the present working directory. The “d” in pushd stands for the directory as it pushes the directory path onto the stack. The directory stack increases in size after each pushd command. This stack is based on the Last In First Out (LIFO) principle.

```
ignitarium@IGN-BLR-LP-215:~$ pushd ~/Desktop
~/Desktop ~
ignitarium@IGN-BLR-LP-215:~/Desktop$ pushd ~/Templates
~/Templates ~/Desktop ~
ignitarium@IGN-BLR-LP-215:~/Templates$ pushd ~/Videos
~/Videos ~/Templates ~/Desktop ~
ignitarium@IGN-BLR-LP-215:~/Videos$ pushd ~/Downloads
~/Downloads ~/Videos ~/Templates ~/Desktop ~
ignitarium@IGN-BLR-LP-215:~/Downloads$
```

---

time command:

time command in Linux is used to execute a command and prints a summary of real-time, user CPU time and system CPU time spent by executing a command when it terminates. ‘Real‘ time is the time elapsed wall clock time taken by a command to get executed, while ‘user‘ and ‘sys‘ time are the number of CPU seconds that command uses in user and kernel mode respectively.

```
ignitarium@IGN-BLR-LP-215:~$ time sleep 3

real    0m3.093s
user    0m0.000s
sys     0m0.003s
ignitarium@IGN-BLR-LP-215:~$
```

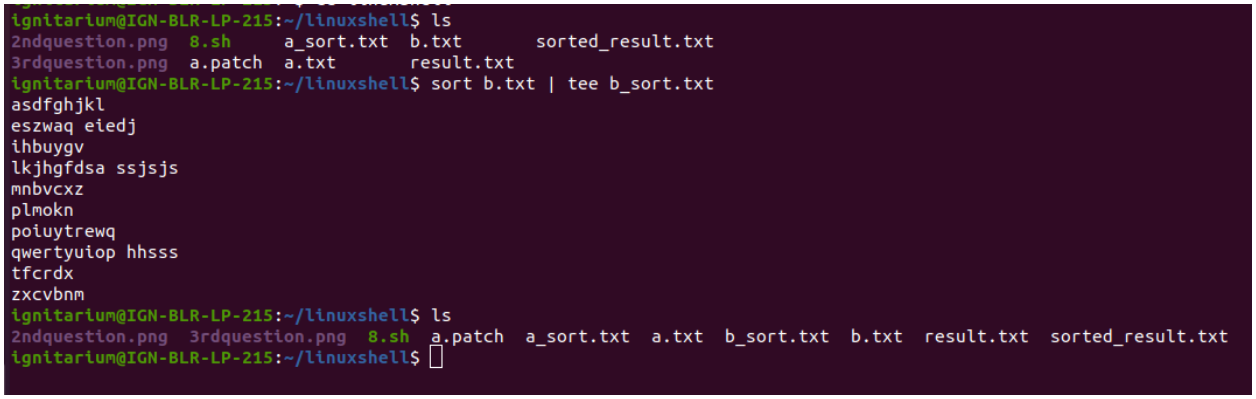
---

diff command:

diff stands for difference. This command is used to display the differences in the files by comparing the files line by line.

---

tee command:
reads the standard input and writes it to both the standard output and one or more files.



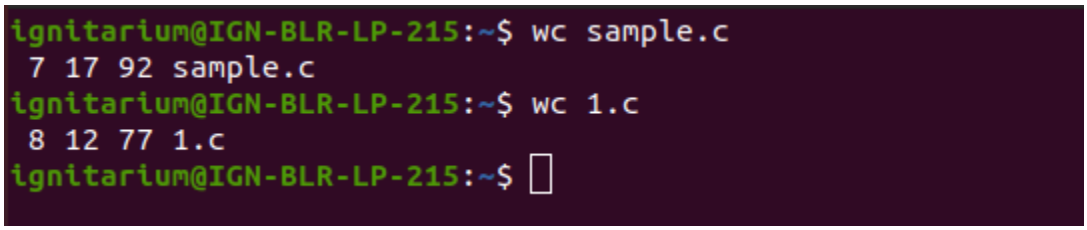
---

popd command:
popd command is used to remove directories from the directory stack. The “d” in popd stands for the directory as it removes the directory path onto the stack. After this command is executed, the present directory stack is displayed as a list of space-separated directories. The directory stack decreases in size after each popd command. This directory stack is based on the Last In First Out (LIFO) principle.

---

wc command:
wc stands for word count. As the name implies, it is mainly used for counting purposes.

- It is used to find out the number of lines, word count, byte and characters count in the files specified in the file arguments.
- By default it displays four-columnar output.
- First column shows number of lines present in a file specified, second column shows number of words present in the file, third column shows number of characters present in file and fourth column itself is the file name which is given as argument.



Options of wc command:

- -l: This option prints the number of lines present in a file. With this option wc command displays two-columnar output, 1st column shows number of lines present in a file and 2nd itself represents the file name.
- -w: This option prints the number of words present in a file. With this option wc command displays two-columnar output, 1st column shows number of words present in a file and 2nd is the file name.
- -c: This option displays the count of bytes present in a file. With this option it displays two-column output, 1st column shows number of bytes present in a file and 2nd is the file name.
- -m: Using -m option 'wc' command displays count of characters from a file.
- -L: The 'wc' command allows an argument -L, it can be used to print out the length of the longest (number of characters) line in a file.
- –version: This option is used to display the version of wc which is currently running on your system.

---

dos2unix command :

converts a DOS text file to UNIX format.

---

unix2dos command :

converts a Unix text file to DOS format

---

less command:

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 a file is large it doesn't access the complete file, but accesses it page by page.

---

which command:

which command in Linux is a command which is used to locate the executable file associated with the given command by searching it in the path environment variable. It has 3 return status as follows:

- 0 : If all specified commands are found and executable.
- 1 : If one or more specified commands is nonexistent or not executable.
- 2 : If an invalid option is specified.

```
ignitarium@IGN-BLR-LP-215:~$ which c cpp java python
/usr/bin/cpp
/usr/bin/java
ignitarium@IGN-BLR-LP-215:~$
```

---

pwd command:

pwd stands for Print Working Directory. It prints the path of the working directory, starting from the root.

pwd -L: Prints the symbolic path.

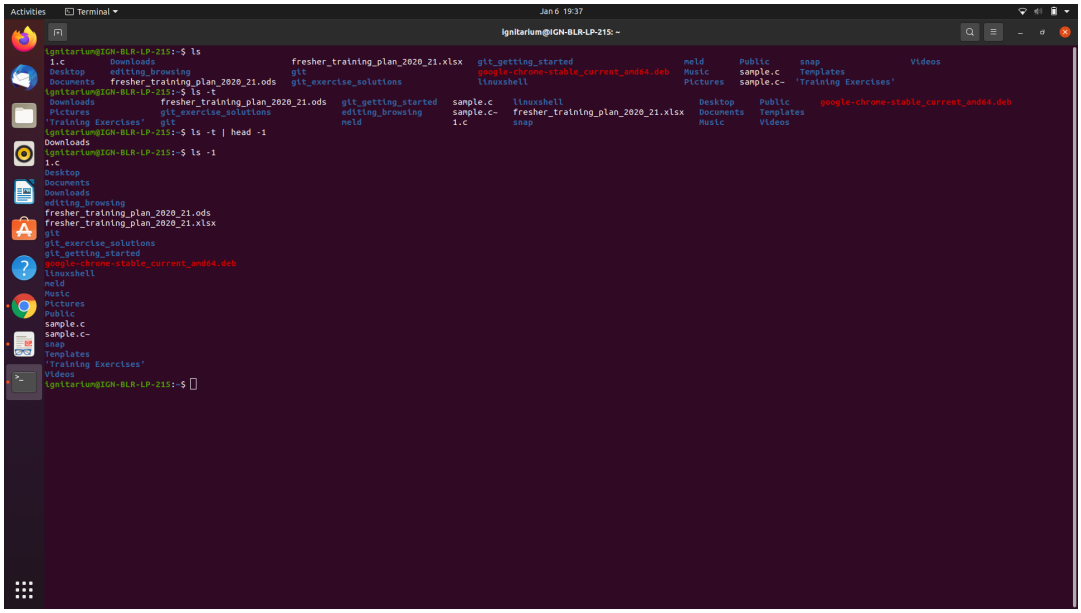
pwd -P: Prints the actual path.

```
ignitarium@IGN-BLR-LP-215:~$ pwd -L
/home/ignitarium
ignitarium@IGN-BLR-LP-215:~$ pwd -P
/home/ignitarium
ignitarium@IGN-BLR-LP-215:~$
```

---

ls command:

ls is a Linux shell command that lists directory contents of files and directories.



dmesg command:
The dmesg command, also called “driver message” or “display message” is used to examine the kernel ring buffer and print the message buffer of the kernel. The output of this command contains the messages produced by the device drivers.

---

locate command:
locate command in Linux is used to find the files by name. There are two most widely used file searching utilities accessible to users are called find and locate. The locate utility works better and faster than find command counterpart because instead of searching the file system when a file search is initiated, it would look through a database.

```
ignitarium@IGN-BLR-LP-215:~$ locate "*.c" -n 10
/home/ignitarium/1.c
/home/ignitarium/sample.c
/home/ignitarium/editing_browsing/a.c
/home/ignitarium/editing_browsing/b.c
/home/ignitarium/git_getting_started/sample.c
/snap/core18/1988/usr/lib/python3/dist-packages/probert/_nl80211module.c
/snap/core18/1988/usr/lib/python3/dist-packages/probert/_rtnetlinkmodule.c
/snap/core18/2253/usr/lib/python3/dist-packages/probert/_nl80211module.c
/snap/core18/2253/usr/lib/python3/dist-packages/probert/_rtnetlinkmodule.c
/snap/core20/1270/usr/lib/python3/dist-packages/probert/_nl80211module.c
ignitarium@IGN-BLR-LP-215:~$
```

Options of locate command:

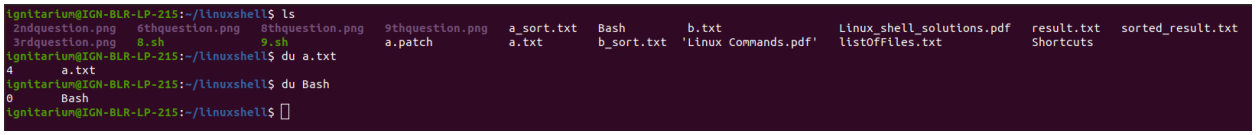
- -b, –basename : Match only the base name against the specified patterns, which is the opposite of –wholename.
- -c, –count : Instead of writing file names on standard output, write the number of matching entries only.
- -d, –database DBPATH : Replace the default database with DBPATH. DBPATH is a : (colon) separated list of database file names. If more than one –database option is specified, the resulting path is a concatenation of the separate paths. An empty database file name is replaced by the default database. A database file name – refers to the standard input. Note that a database can be read from the standard input only once.
- -e, –existing : Print only entries that refer to files existing at the time the location is run.
- -L, –follow : When checking whether files exist (if the –existing option is specified), follow trailing symbolic links. This causes broken symbolic links to be omitted from the output. This option is the default behavior. The opposite can be specified using –nofollow.
- -h, –help : Write a summary of the available options to standard output and exit successfully.
- -i, –ignore-case : Ignore case distinctions when matching patterns.

- -l, –limit, -n LIMIT : Exit successfully after finding LIMIT entries. If the –count option is specified, the resulting count is also limited to LIMIT.
- -m, –mmap : Ignored, but included for compatibility with BSD and GNU locate.
- -P, –nofollow, -H : When checking whether files exist (if the –existing option is specified), do not follow trailing symbolic links. This causes broken symbolic links to be reported like other files.
  This option is the opposite of –follow.
- -0, –null : Separate the entries on output using the ASCII NUL character instead of writing each entry on a separate line. This option is designed for interoperability with the –null option of GNU xargs.
- -S, –statistics : Write statistics about each read database to standard output instead of searching for files and exit successfully.
- -q, –quiet : Write no messages about errors encountered while reading and processing databases.
- -r, –regexp REGEXP : Search for a basic regexp REGEXP. No PATTERNs are allowed if this option is used, but this option can be specified multiple times.
- –regex : Interpret all PATTERNs as extended regexps.
- -s, –stdio : Ignored, for compatibility with BSD and GNU locate.
- -V, –version : Write information about the version and license of locate on standard output and exit successfully.
- -w, –wholename : Match only the whole path name against the specified patterns. This option is the default behavior. The opposite can be specified using –basename.

---

du command:

du command, short for disk usage, is used to estimate file space usage.
The du command can be used to track the files and directories which are consuming excessive amounts of space on the hard disk drive.



---

xxd command:

xxd produces a hexadecimal or binary dump of a file in several different formats. It can also do the reverse, converting from its hex dump format back into the original data.

```
ignitarium@IGN-BLR-LP-215:~/linuxshell$ xxd 8.sh
00000000: 6563 686f 2022 456e 7465 7220 7468 6520  echo "Enter the
00000010: 7661 6c75 6520 6f66 206e 3a20 220a 7265  value of n: ".re
00000020: 6164 206e 0a0a 7375 6d3d 300a 6563 686f  ad n..sum=0.echo
00000030: 2022 456e 7465 7220 7468 6520 6e75 6d62   "Enter the numb
00000040: 6572 733a 2022 0a66 6f72 2828 693d 313b  ers: ".for((i=1;
00000050: 693c 3d6e 3b69 2b2b 2929 0a64 6f0a 0972  i<=n;i++)).do..r
00000060: 6561 6420 6e75 6d0a 0973 756d 3d24 2828  ead num..sum=$((
00000070: 7375 6d20 2b20 6e75 6d29 290a 646f 6e65  sum + num)).done
00000080: 0a65 6368 6f20 2253 756d 206f 6620 6769  .echo "Sum of gi
00000090: 7665 6e20 6e75 6d62 6572 7320 6973 3a20  ven numbers is:
000000a0: 220a 6563 686f 2024 7375 6d0a            ".echo $sum.
ignitarium@IGN-BLR-LP-215:~/linuxshell$
ignitarium@IGN-BLR-LP-215:~/linuxshell$
```

---

lspci command:

lspci command is a utility on linux systems used to find out information about the PCI busses and devices connected to the PCI subsystem.

```
ignitarium@IGN-BLR-LP-215:~/linuxshell$ lspci
00:00.0 Host bridge: Intel Corporation Device 9b61 (rev 0c)
00:02.0 VGA compatible controller: Intel Corporation UHD Graphics (rev 02)
00:04.0 Signal processing controller: Intel Corporation Xeon E3-1200 v5/E3-1500 v5/6th Gen Core Processor Thermal Subsystem (rev 0c)
00:08.0 System peripheral: Intel Corporation Xeon E3-1200 v5/v6 / E3-1500 v5 / 6th/7th/8th Gen Core Processor Gaussian Mixture Model
00:12.0 Signal processing controller: Intel Corporation Comet Lake Thermal Subsytem
00:14.0 USB controller: Intel Corporation Device 02ed
00:14.2 RAM memory: Intel Corporation Device 02ef
00:14.3 Network controller: Intel Corporation Wireless-AC 9462
00:15.0 Serial bus controller [0c80]: Intel Corporation Serial IO I2C Host Controller
00:16.0 Communication controller: Intel Corporation Comet Lake Management Engine Interface
00:17.0 SATA controller: Intel Corporation Comet Lake SATA AHCI Controller
00:1d.0 PCI bridge: Intel Corporation Device 02b0 (rev f0)
00:1d.2 PCI bridge: Intel Corporation Device 02b2 (rev f0)
00:1e.0 Communication controller: Intel Corporation Device 02a8
00:1f.0 ISA bridge: Intel Corporation Device 0284
00:1f.3 Audio device: Intel Corporation Device 02c8
00:1f.4 SMBus: Intel Corporation Device 02a3
00:1f.5 Serial bus controller [0c80]: Intel Corporation Comet Lake SPI (flash) Controller
01:00.0 Ethernet controller: Realtek Semiconductor Co., Ltd. RTL8111/8168/8411 PCI Express Gigabit Ethernet Controller (rev 15)
02:00.0 Unassigned class [ff00]: Realtek Semiconductor Co., Ltd. RTS522A PCI Express Card Reader (rev 01)
ignitarium@IGN-BLR-LP-215:~/linuxshell$
```

THANK YOU!!!