Shell Programming - Class Notes

Linux

→ It is an Open Source operating system. It is available free to use and user can modify it according their need.
→ The founder or linux is Linun Torvards. It available since 1991.
→ An Open Source Community is working behind the updation and upgradation of the linux code.
Feature

1. No Cost / Low Cost
2. Multi-Tasking
3. Security
4. Multi-User
5. Stable and Scalable
6. Networking
7. CLI as well as GUI
8. Better File System

File system of Linux
/ is root directory

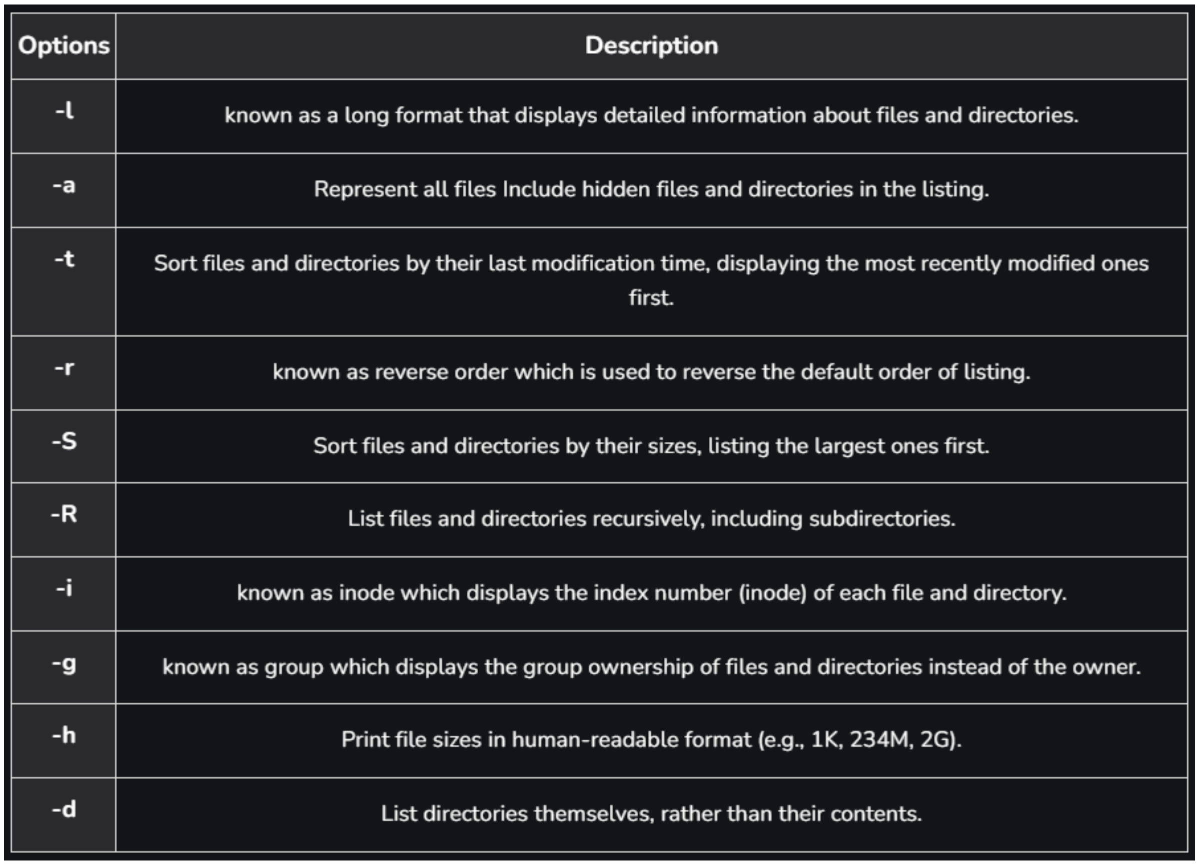
1. /bin: User Bineries
2. /sbin: System Bineries
3. /etc: Configuration Files
4. /home: Parent directory of user friendly directory
5. /lib: System Libraries

6. /proc: Process Information
7. /dev: Device Files
8. /var: Variables Files
9. /tmp: Temporary Files
10. /usr: User Programs
11. /boot: Boot Loader Files
12. /opt: Apps

Commands associated with files/directories

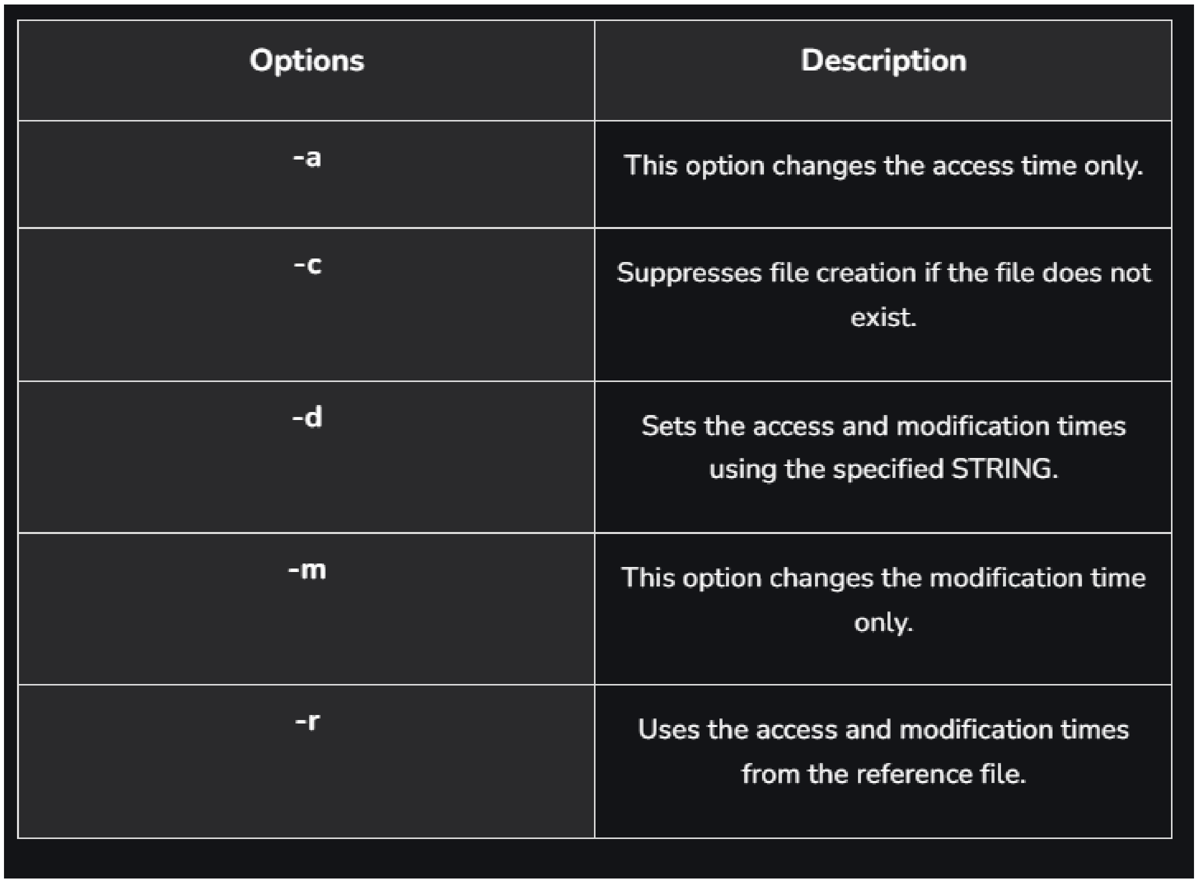
1. pwd: Print Working Directory
2. ls: it list out all the files and directory of current working directory

   commonly used options in ls command in linux



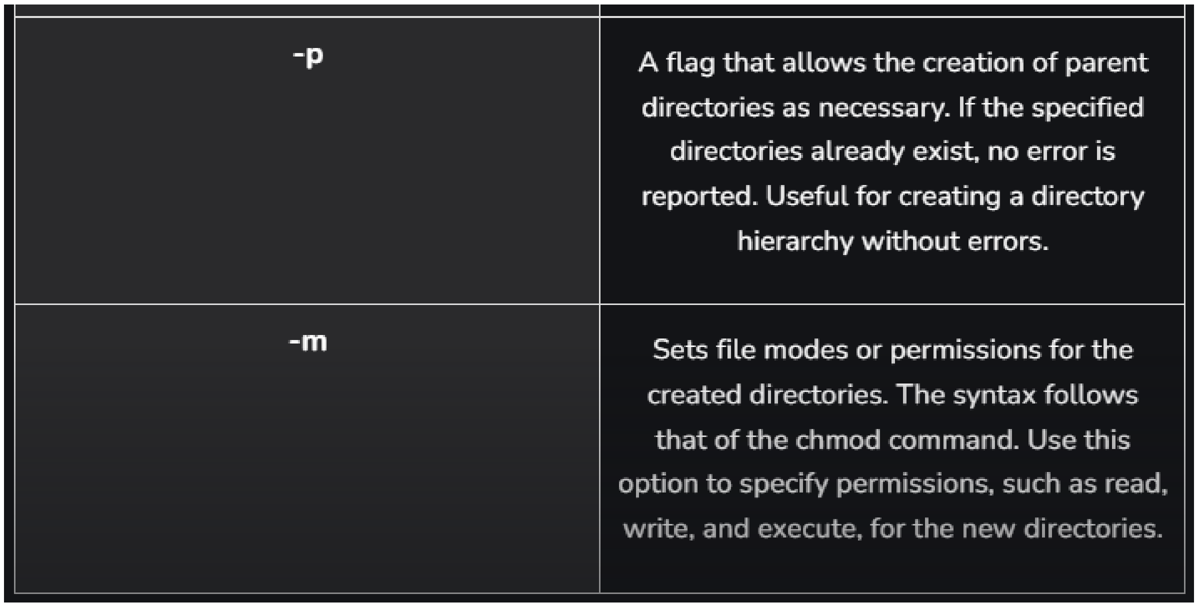
Options	Description
-l	known as a long format that displays detailed information about files and directories.
-a	Represent all files Include hidden files and directories in the listing.
-t	Sort files and directories by their last modification time, displaying the most recently modified ones first.
-r	known as reverse order which is used to reverse the default order of listing.
-S	Sort files and directories by their sizes, listing the largest ones first.
-R	List files and directories recursively, including subdirectories.
-i	known as inode which displays the index number (inode) of each file and directory.
-g	known as group which displays the group ownership of files and directories instead of the owner.
-h	Print file sizes in human-readable format (e.g., 1K, 234M, 2G).
-d	List directories themselves, rather than their contents.

3. nano: it actually run the nano editor and open the specified file.
4. touch: It is used to create a new file



Options	Description
-a	This option changes the access time only.
-c	Suppresses file creation if the file does not exist.
-d	Sets the access and modification times using the specified STRING.
-m	This option changes the modification time only.
-r	Uses the access and modification times from the reference file.

5. mkdir: To create a new directory.



-p	A flag that allows the creation of parent directories as necessary. If the specified directories already exist, no error is reported. Useful for creating a directory hierarchy without errors.
-m	Sets file modes or permissions for the created directories. The syntax follows that of the chmod command. Use this option to specify permissions, such as read, write, and execute, for the new directories.

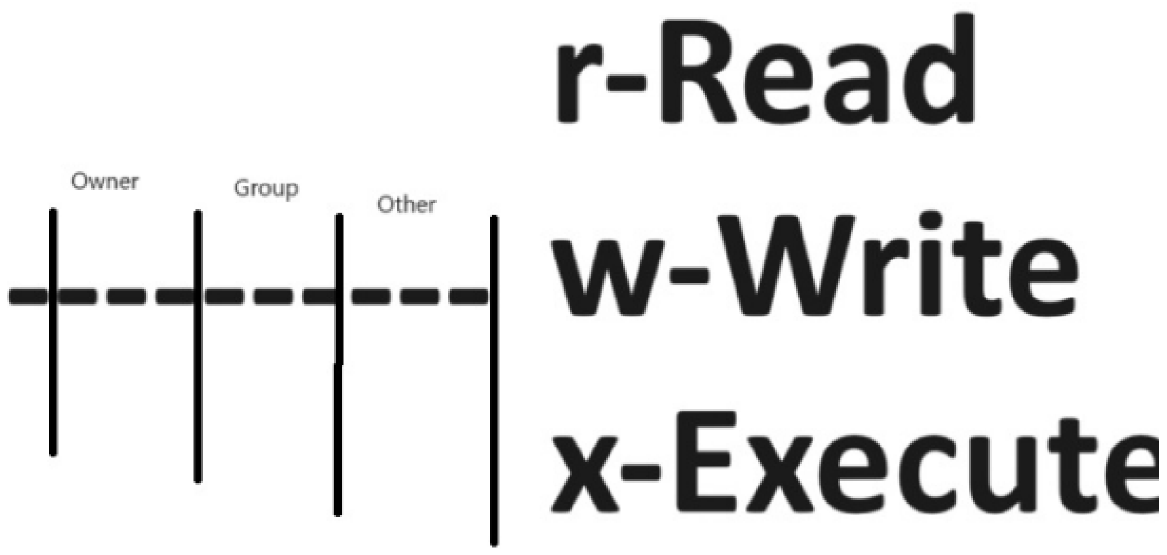
6. chmod: to give and revoke the file or directory permissions

   u: Owner

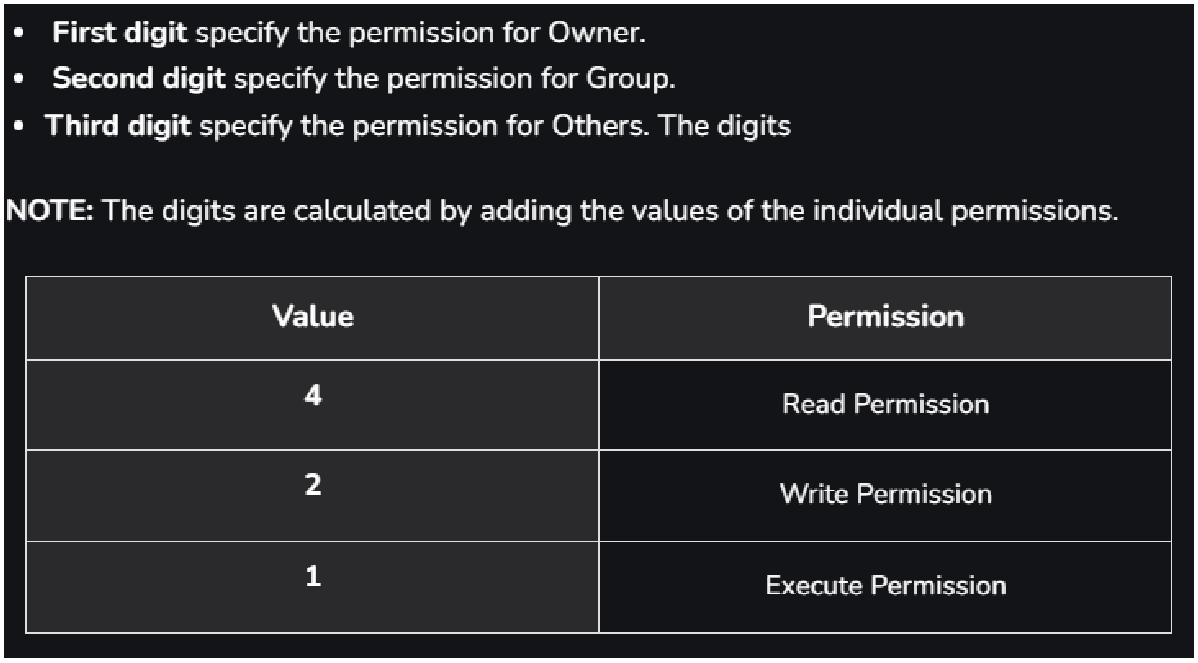
   g: Group

   o: Others

   a: All



Octal Representation in chmod:



- **First digit** specify the permission for Owner.
- **Second digit** specify the permission for Group.
- **Third digit** specify the permission for Others. The digits

NOTE: The digits are calculated by adding the values of the individual permissions.

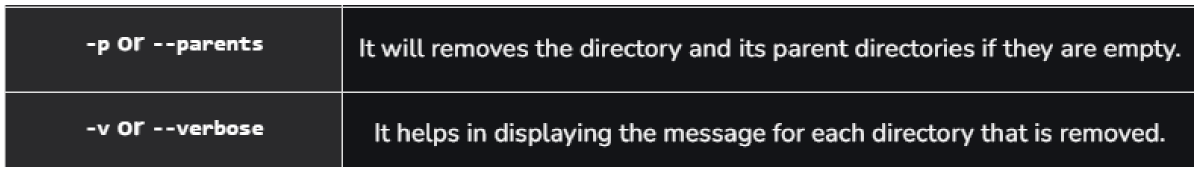
Value	Permission
4	Read Permission
2	Write Permission
1	Execute Permission

- **6** represent permission of file Owner which are **(rw).**
- **7** represent permission of Group which are **(rwx).**
- **4** represent permission of Other which is **(r).**

7. rm: to remove file and recursive directory

```
rm <filename>
rm -r <directory> #Will delete all directories and sub ds
rm -i <file>      #ask the user for confirmation before remov
rm -f <file>      #confirmation removal if a file is write pr
```

8. rmdir: to remove a perticuler directory



`-p` or `--parents`	It will removes the directory and its parent directories if they are empty.
`-v` or `--verbose`	It helps in displaying the message for each directory that is removed.

9. cd: to change directory
10. chown: to change owner of the file

```
chown <username> <filename>
-c #will report options
```

11. su: to switch the user from current to any specified user
12. cat: to display content of the file on console

```
cat file_name    #Display content of file
cat -n file_name  #Adds line number before
cat > file        #Add fresh content to file
cat >> file       #Appends content
cat file1 file2 > merge    #merges 2 files
cat -s file       #supress repeated multiple lines
tac file          #Used to show content of file in reverse o
cat -E file       #Highlight end of the line with $
cat *.txt         #Display content of all text files in fold
```

13. head: to display top n lines of the file on console. By default it will print first 10 lines

```
head -n num file    #prints first n lines
head -c num file    #prints first n chars
head -n 20 state.txt | tail -10
#To print data between 2 lines. here: data between 10 and 20
```

14. tail: to display bottom n lines of the file on console. By default it will print last 10 lines

```
tail -n 3 state.txt    ##prints first n lines
tail +25 state.txt     #start printing from line number ‘n’ 
tail -c -7 state.txt   #prints last n chars
tail -f logfile        # to monitor the growth of the log fi
```

15. adduser: to add new user into the system
16. sudo: to give some specific privilages to the user's other than root.

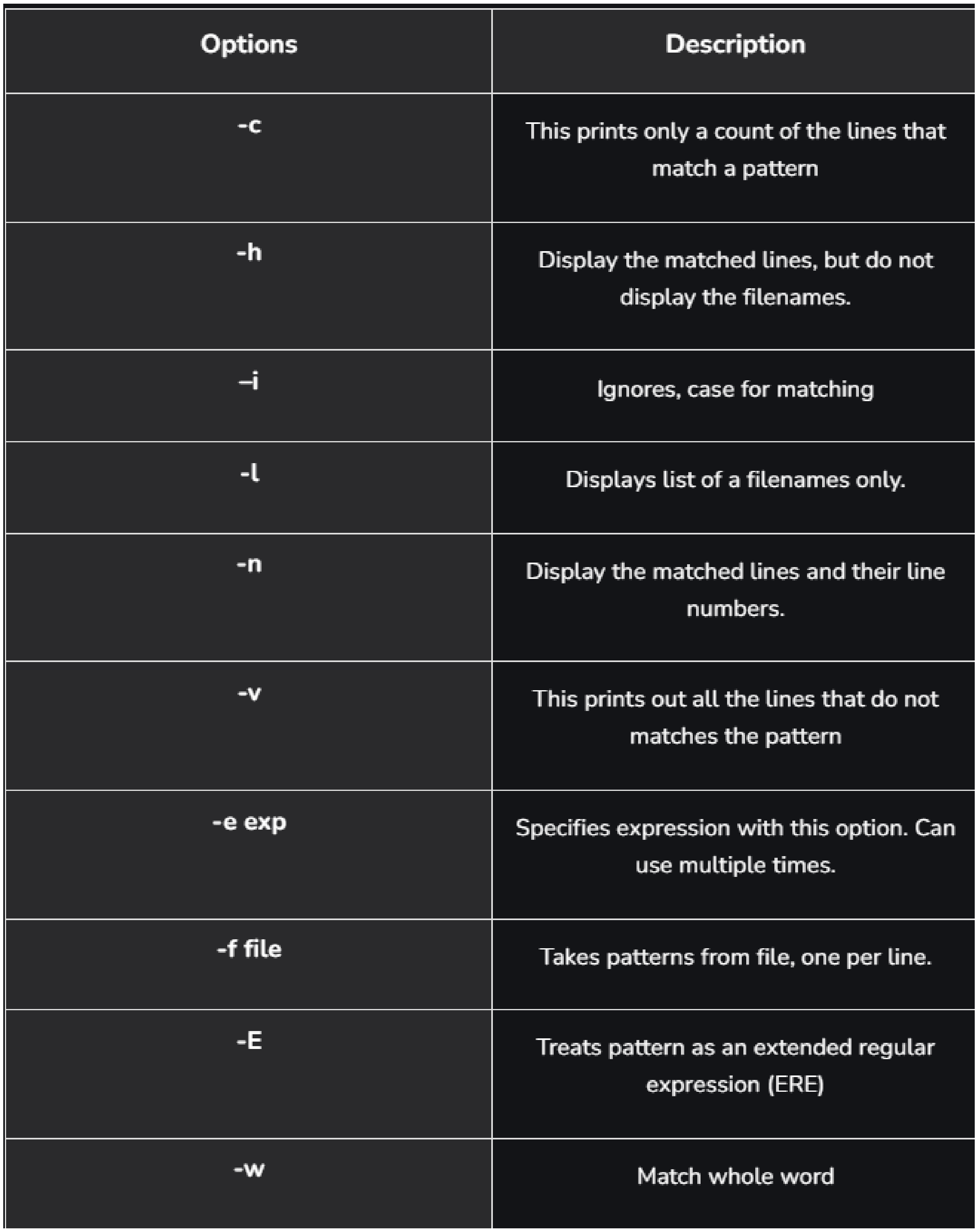
17. mv: Used to rename and move file or directories
18. cp: used to copy data from one destination to other
19. echo: used to print a string on command line

```
echo string
echo -e "\nsumant"    # enables the interpretation of backsla
echo *                # will act as a ls command
echo -n string        # avoid newline
echo "Welcome GFG" > output.txt    #output of echo is redire
```

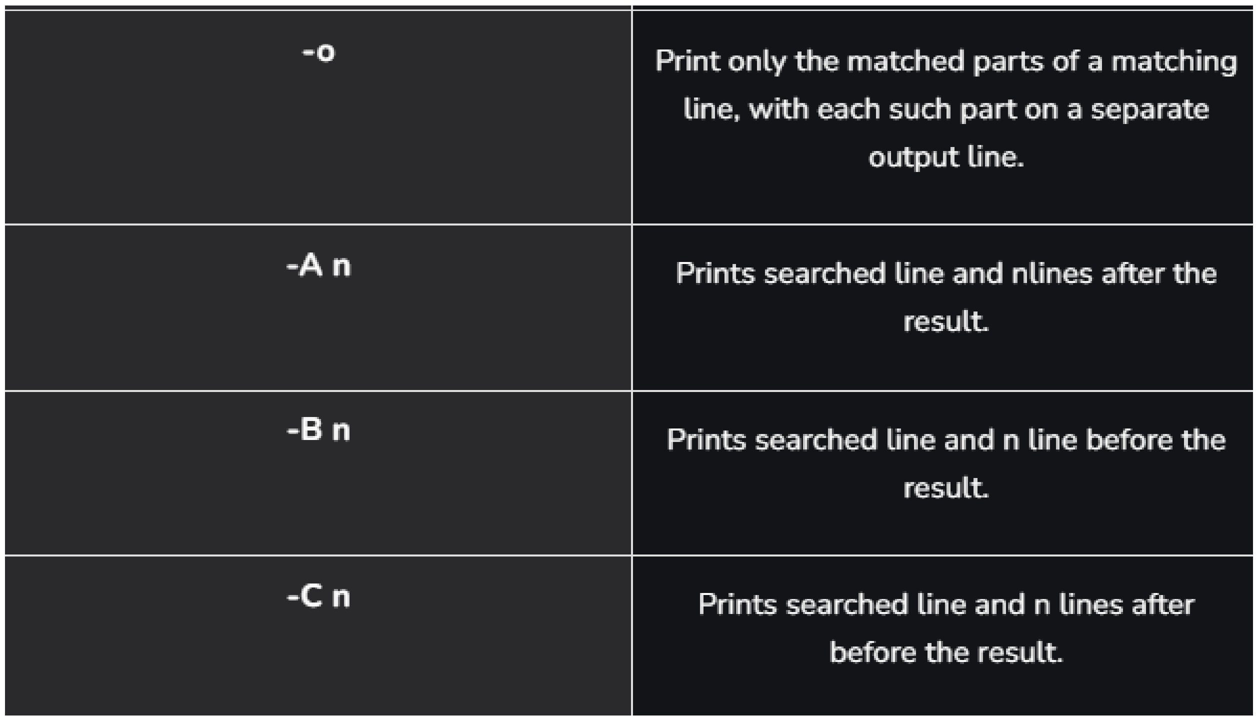
20. man: to access maual of any command

```
man <command name>
-f : one line descp
-k : command for given keyword
```

21. uname: This command '**uname**' displays the information about the system.
22. grep: command is used to search for patterns in file

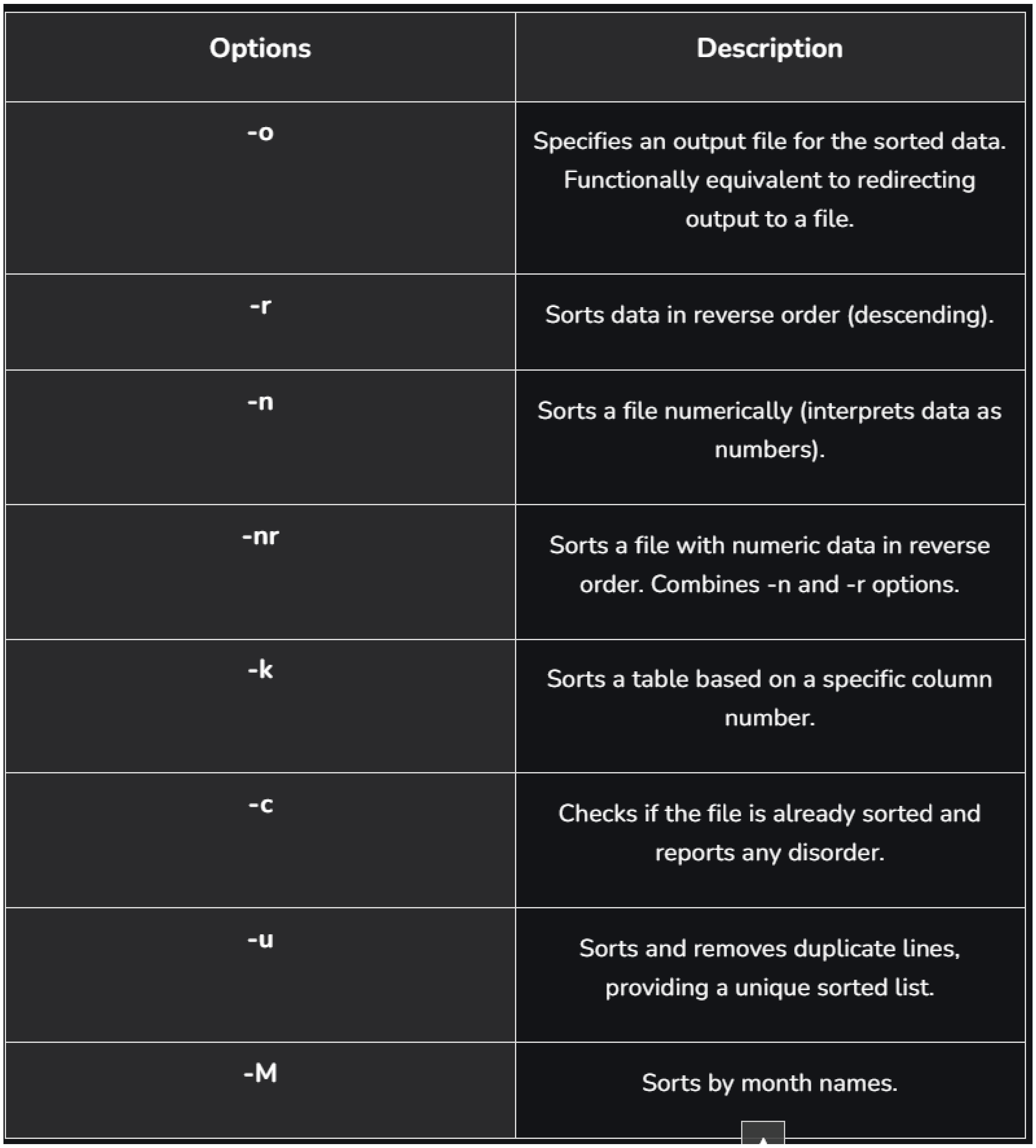


Options	Description
-c	This prints only a count of the lines that match a pattern
-h	Display the matched lines, but do not display the filenames.
–i	Ignores, case for matching
-l	Displays list of a filenames only.
-n	Display the matched lines and their line numbers.
-v	This prints out all the lines that do not matches the pattern
-e exp	Specifies expression with this option. Can use multiple times.
-f file	Takes patterns from file, one per line.
-E	Treats pattern as an extended regular expression (ERE)
-w	Match whole word



-o	Print only the matched parts of a matching line, with each such part on a separate output line.
-A n	Prints searched line and nlines after the result.
-B n	Prints searched line and n line before the result.
-C n	Prints searched line and n lines after before the result.

23. diff: command is used to differentiate the files
24. cmp: reports the location of the first mismatch to the screen
25. sort: sorts the data in file



Options	Description
-o	Specifies an output file for the sorted data. Functionally equivalent to redirecting output to a file.
-r	Sorts data in reverse order (descending).
-n	Sorts a file numerically (interprets data as numbers).
-nr	Sorts a file with numeric data in reverse order. Combines -n and -r options.
-k	Sorts a table based on a specific column number.
-c	Checks if the file is already sorted and reports any disorder.
-u	Sorts and removes duplicate lines, providing a unique sorted list.
-M	Sorts by month names.

26. SED command in UNIX stands for stream editor and it can perform lots of functions on file like searching, find and replace, insertion or deletion

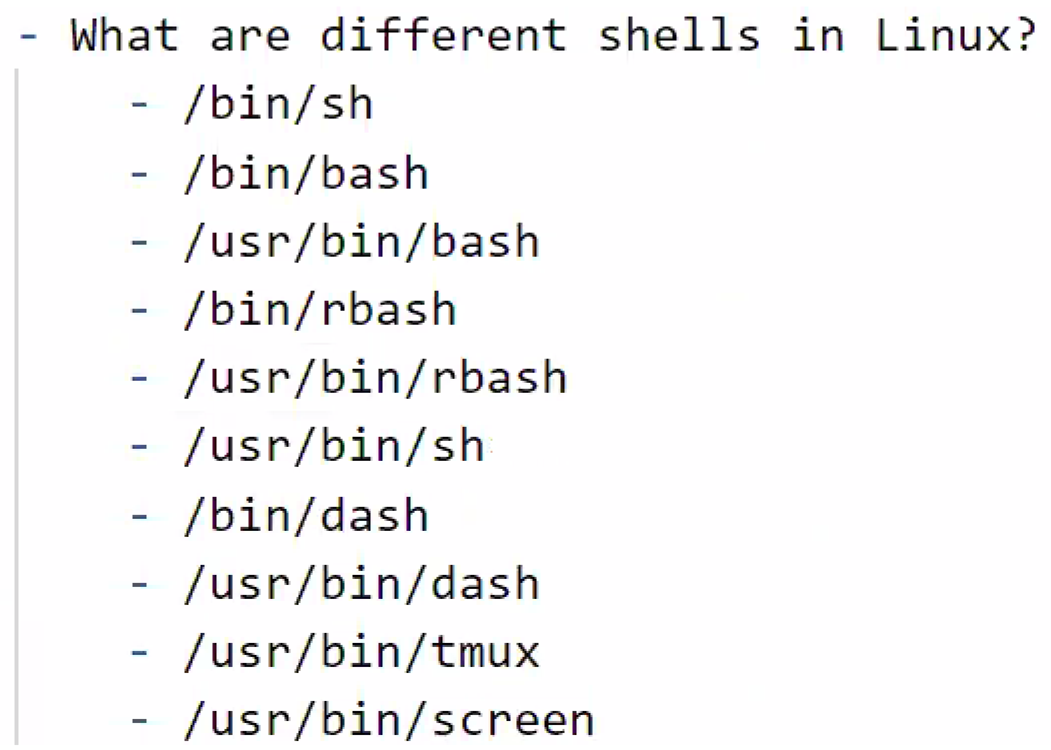
```

```

Shell Programming:

Shell is basically interface between user and kernel

An user can interact with kernel by using shell commands or shell script/program

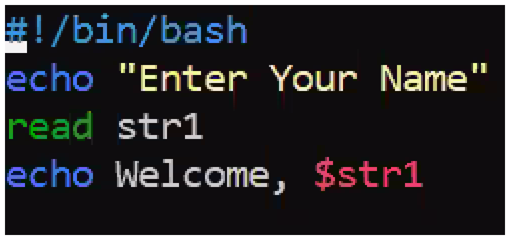


```
- What are different shells in Linux?
    - /bin/sh
    - /bin/bash
    - /usr/bin/bash
    - /bin/rbash
    - /usr/bin/rbash
    - /usr/bin/sh
    - /bin/dash
    - /usr/bin/dash
    - /usr/bin/tmux
    - /usr/bin/screen
```

In Unix, True = 0, False = 1.

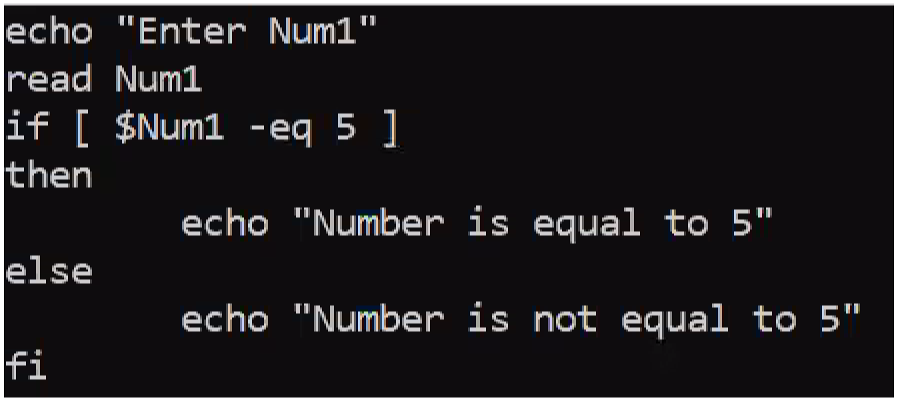
Variables are stored in shell until the shell is alive

1. Address of the shell
2. to print prompt messages : echo command
3. To take input: read <variable name>
4. To use variable’s value, ‘$’ is used

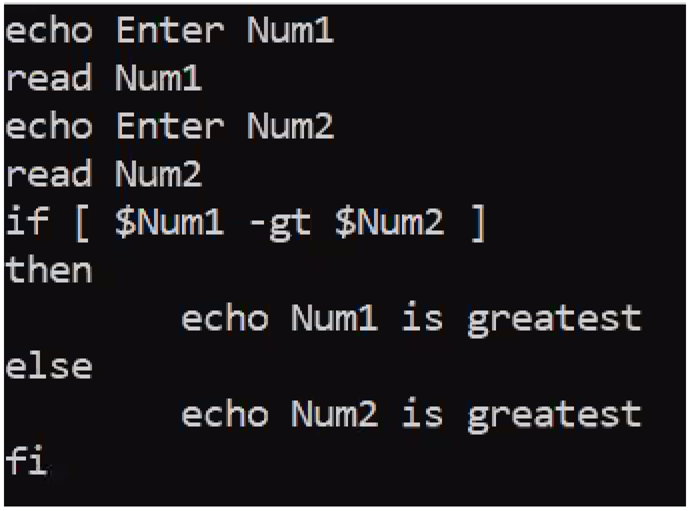


```
#!/bin/bash
echo "Enter Your Name"
read str1
echo Welcome, $str1
```

5. if-then-else-fi block to apply conditions

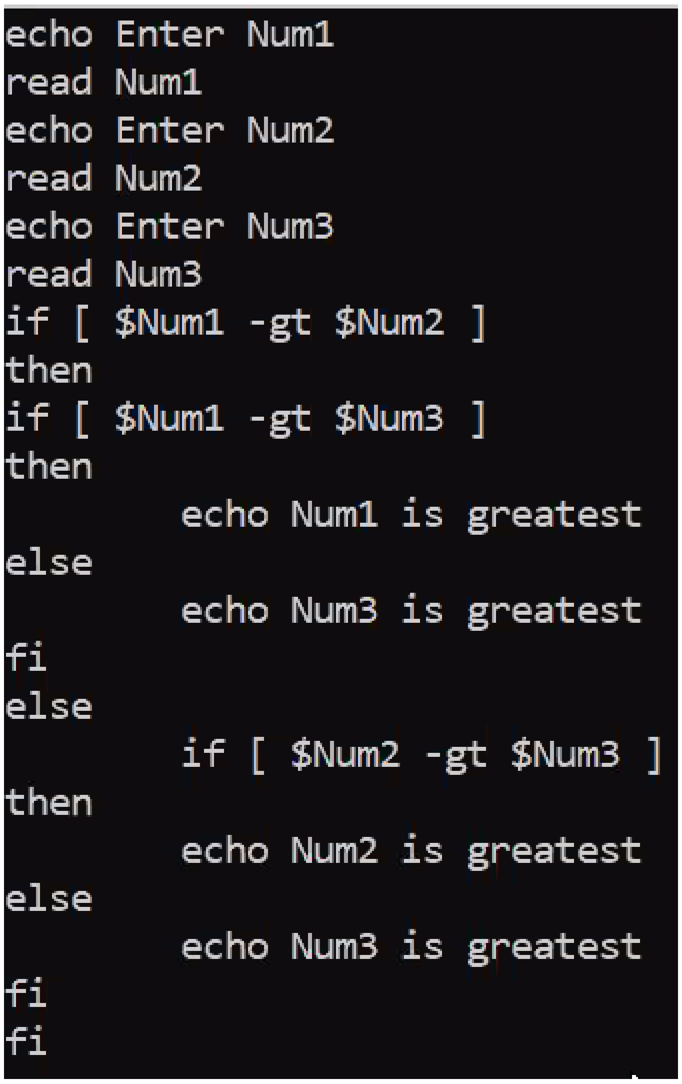


```
echo "Enter Num1"
read Num1
if [ $Num1 -eq 5 ]
then
        echo "Number is equal to 5"
else
        echo "Number is not equal to 5"
fi
```



```
echo Enter Num1
read Num1
echo Enter Num2
read Num2
if [ $Num1 -gt $Num2 ]
then
        echo Num1 is greatest
else
        echo Num2 is greatest
fi
```

6. nested if else



```
echo Enter Num1
read Num1
echo Enter Num2
read Num2
echo Enter Num3
read Num3
if [ $Num1 -gt $Num2 ]
then
if [ $Num1 -gt $Num3 ]
then
        echo Num1 is greatest
else
        echo Num3 is greatest
fi
else
        if [ $Num2 -gt $Num3 ]
then
        echo Num2 is greatest
else
        echo Num3 is greatest
fi
fi
```

7. Loops
    a. for loop

```
for a in 1 2 3 4 5 6
do
    echo $a
done
```

b. while loop

```
a=0
while [ $a -le 6 ]
```

```
do
    echo $a
    a=`expr $a + 1`
done
```

c. until loop

```
a=0
until [ $a -gt 6 ]
do
    echo $a
    a=`expr $a + 1`
done
```

→ while loop runs until the condition goes false

→ until loop runs until the condition goes true

Integer Comparison Operators

-gt: Greater than

-lt: Less than

-ge: Greater than or equal to

-le: Less than or equal to

-eq: Equal to

String Operators: >,<, ==, empty

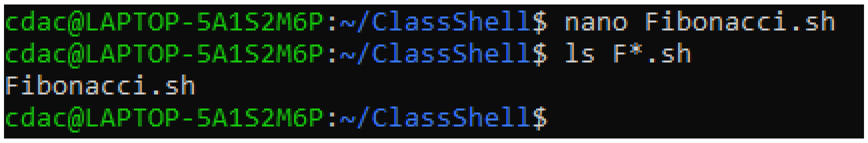
3. Wildcard Symbols: Special characters in shell scripting to give different characteristics to different commands

   a. Pipe (|) : Piping allows us to filteration among the results of command

   ```
   cat sumant.txt | grep ^S
   #It filters out lines starting from S from all lines in ca
   ```

b. Redirection (>) : Output of one command to input to other command

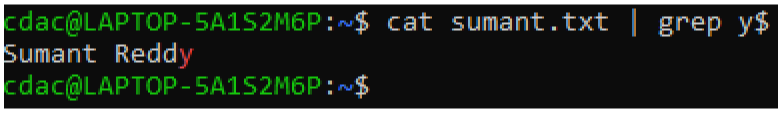
c. Asterisk (*) : is like a filler of words



```
cdac@LAPTOP-5A1S2M6P:~/ClassShell$ nano Fibonacci.sh
cdac@LAPTOP-5A1S2M6P:~/ClassShell$ ls F*.sh
Fibonacci.sh
cdac@LAPTOP-5A1S2M6P:~/ClassShell$
```

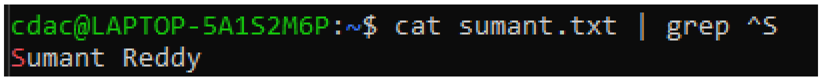
d. Tilde (~) : Denotes user's home directory

e. Doller Symbol ($): It is also used to match the string by it's end



```
cdac@LAPTOP-5A1S2M6P:~$ cat sumant.txt | grep y$
Sumant Reddy
cdac@LAPTOP-5A1S2M6P:~$
```

f. Caret (^) : If you want to find a line starting with or ending with a character, carat (^) symbol is used.



```
cdac@LAPTOP-5A1S2M6P:~$ cat sumant.txt | grep ^S
Sumant Reddy
```

4. Command Line Arguments

   The argument which are provided while writing the command

   1. $#: will return numbers of parameter passed
   2. $0: will return script name
   3. $i: where return ith parameter
   4. $*: will return all parameters

5. Arithmatic Expressions

   - $a + $b

   - $a - $b

   - $a * $b

- $a / $b
- $a % $b

→ Test Command

to test status of something

```
x=60
y=60

if test $x -eq $y                      #For String: test $x =
then
        echo both are equal
else
        echo not equal
fi
```

6. Network Command
    a. Telenet: it is used to take control of remote server
    b. ftp: (File Transfer Protocol) it allows user to send or recieve file from remote server / machine
    c. ssh: Also used to take control of remote server or machine
    d. stfp: (Secure File Transfer Protocol)
    e. finger:

7. If we want to access the application from anywhere then the path of the application is given in environment variable