

1. Explain about file and directory permissions?

File and directory permissions are a fundamental part of maintaining security and proper functionality in an operating system. They determine who can read, write, or execute a file or directory

File permissions are based on a model that assigns different permissions to three classes of users:

- owner (user)
- group
- others

Permission Types:

Read (r):	Permission to read the file or list the contents of the directory
Write (w):	Permission to modify the contents of the file or directory.
Execute (x):	Permission to execute the file (if it is a script or binary) or access the directory (useful for navigating into the directory).

A common way to view file and directory permissions is with the `ls -l` command .This command displays a detailed listing, including permission information

```
[udaykiran@localhost user1]$ ls
py python shell tcl
[udaykiran@localhost user1]$ ls -l
total 0
-rw-rw-r--. 1 udaykiran udaykiran 0 Jul 10 19:22 py
drwxrwxr-x. 2 udaykiran udaykiran 24 Jul  5 12:18 python
drwxrwxr-x. 2 udaykiran udaykiran 24 Jul  5 11:50 shell
drwxrwxr-x. 2 udaykiran udaykiran 25 Jul  5 12:18 tcl
[udaykiran@localhost user1]$ 
```

2. What is the command used for getting a long listing format and explain the content of it?

A common way to view file for getting a long listing format `ls -l` command .This command displays a detailed listing, including permission information

```
[udaykiran@localhost user1]$ ls
py python shell tcl
[udaykiran@localhost user1]$ ls -l
total 0
-rw-rw-r--. 1 udaykiran udaykiran 0 Jul 10 19:22 py
drwxrwxr-x. 2 udaykiran udaykiran 24 Jul  5 12:18 python
drwxrwxr-x. 2 udaykiran udaykiran 24 Jul  5 11:50 shell
drwxrwxr-x. 2 udaykiran udaykiran 25 Jul  5 12:18 tcl
[udaykiran@localhost user1]$ 
```

Example: -rw-rw-r- .1 udaykiran udaykiran 0 Jul 10 19:22 py

1st Character – File Type: First character specifies the type of the file.

In the example above the hyphen (-) in the 1st character indicates that this is a normal file.

Field Explanation

-	normal file
d	directory
S	socket file
l	link file

Field 1 – File Permissions: Next characters specify the files permission. Every 3 characters specify read, write, and execute permissions for user (root), group and others respectively in order. Taking the above example, -rw-rw-r- indicates read-write permission for user (root), read permission for group, and no permission for others respectively. If all three permissions are given to user (root), group and others, the format looks like -rwxrwxrwx

Field 2 – Number of links: Second field specifies the number of links for that file. In this example, 1 indicates only one link to this file.

Field 3 – Owner: Third field specifies owner of the file. In this example, this file is owned by username ‘udaykiran’.

Field 4 – Group: Fourth field specifies the group of the file. In this example, this file belongs to” udaykiran”.

Field 5 – Size: Fifth field specifies the size of file in bytes. In this example, ‘0’ indicates the file size in bytes.

Field 6 – Last modified date and time: Sixth field specifies the date and time of the last modification of the file. In this example, ‘Jul 10 19:22 specifies the last modification time of the file.

Field 7 – File name: The last field is the name of the file. In this example, the file name is py.

3. Provide two ways to change permissions i.e., by using the octal method and the symbolic method ?

Representation:

Permissions can be represented in two ways:

- **Octal notation:** Uses numbers (0-7) to represent each permission type (read, write, execute) for each user category (owner, group, others).
- **Symbolic notation:** Uses letters (r, w, x) to represent permissions, which can be easier to understand for some users.

A common way to view file and directory permissions is with the ls -l command .This command displays a detailed listing, including permission information

Notations:

(u):user	(r):read
(g):group	(w):write
(o):other	(x):execute
(+):add permission	(-): remove permission

decimal	binary			permissions
	r	w	x	
0	0	0	0	No permission
1	0	0	1	Execute
2	0	1	0	Write
3	0	1	1	Write & Execute
4	1	0	0	Read
5	1	0	1	Read & Execute
6	1	1	0	Read & Write
7	1	1	1	Read & Write & Execute

Command used to change permissions for files and directories is

Chmod : change mode

chmod u=w+x,g=r+w,o=r+w+x table → {symbolic or text method}

```
[udaykiran@localhost ~]$ ls -l table
-r-x---wx. 1 udaykiran udaykiran 677 Jul 10 21:29 table
[udaykiran@localhost ~]$ chmod u=w+x,g=r+w,o=r+w+x table
[udaykiran@localhost ~]$ ls -l table
--wxrw-rwx. 1 udaykiran udaykiran 677 Jul 10 21:29 table
[udaykiran@localhost ~]$ █
```

For the above command the permissions for the table has been changed as (user = write & execute, group=read & write, others=Read & write & execute)

This command will change the permissions of user, group and others note + and- are used to add permission and remove permissions respectively.

chmod 457 table → {using mode number} 777

given to all user,group and others full permissions

```
[udaykiran@localhost ~]$ ls
Desktop Documents Downloads Home Music Pictures Public table Templates uday Videos
[udaykiran@localhost ~]$ ls -l table
-rwxrwxrwx. 1 udaykiran udaykiran 677 Jul 10 21:29 table
[udaykiran@localhost ~]$ chmod 457 table
[udaykiran@localhost ~]$ ls -l
total 4
drwxr-xr-x. 2 udaykiran udaykiran 6 Jul 3 10:10 Desktop
drwxr-xr-x. 2 udaykiran udaykiran 6 Jul 3 10:10 Documents
drwxr-xr-x. 2 udaykiran udaykiran 6 Jul 3 10:10 Downloads
drwxrwxr-x. 5 udaykiran udaykiran 45 Jul 5 11:46 Home
drwxr-xr-x. 2 udaykiran udaykiran 6 Jul 3 10:10 Music
drwxr-xr-x. 2 udaykiran udaykiran 6 Jul 9 16:19 Pictures
drwxr-xr-x. 2 udaykiran udaykiran 6 Jul 3 10:10 Public
-r--r-xrwx. 1 udaykiran udaykiran 677 Jul 10 21:29 table
drwxr-xr-x. 2 udaykiran udaykiran 6 Jul 3 10:10 Templates
drwxrwxr-x. 5 udaykiran udaykiran 72 Jul 8 12:20 uday
drwxr-xr-x. 2 udaykiran udaykiran 6 Jul 3 10:10 Videos
[udaykiran@localhost ~]$ ls -l table
-r--r-xrwx. 1 udaykiran udaykiran 677 Jul 10 21:29 table
[udaykiran@localhost ~]$ █
```

For chmod 457 table gives (for user – read, Group-Read and execute, others-Read,write and Execute) permission will be given

4.Explain the following operators with examples.

a. pipe “|”

b. asterisk “*”

c. Question mark “?”

The operators pipe (|), asterisk (*), and question mark (?)

Pipe (|)

The pipe operator (|) is used to pass the output of one command as the input to another command. This allows you to chain commands.

Example

- Using pipe to filter and count lines:**

Suppose you have a file named `file.txt` and you want to count the number of lines that contain the word "example".

```
grep "example" file.txt | wc -l
```

- grep "example" file.txt: Searches for the word "example" in `file.txt`.
- |: Passes the output of `grep` to the `wc` command.
- `wc -l`: Counts the number of lines in the output.

Asterisk (*)

The asterisk * is a wildcard character that represents zero or more characters in filenames or directory names. It is commonly used in file manipulation commands to operate on multiple files at once.

Example

- **Using * to list all files:**

```
ls *.txt
```

This command lists all files in the current directory that have a .txt extension.

- **Using * to delete multiple files:**

```
rm *.log
```

This command deletes all files in the current directory that have a .log extension.

Question Mark (?)

The question mark (?) is a wildcard character that represents exactly one character in filenames or directory names. It is used to match files with specific patterns.

Example

- **Using? to list files with single-character differences:**

```
ls file?.txt
```

This command lists all files in the current directory that match the pattern `file?.txt`, such as `file1.txt`, `file2.txt`, `fileA.txt`, but not `file10.txt` (because? matches only one character).

- **Using? to delete specific files:**

```
rm log?.txt
```

This command deletes files like `log1.txt`, `log2.txt`, but not `log10.txt`.

These operators are powerful tools in the Linux command line that help in managing and manipulating files and data efficiently.

5. The `sort` command is a versatile tool in Linux for organizing the contents of text files. Here's a breakdown of its functionalities and common usage:

Sorting Basics:

- Sorts lines of text in a file by default.
- Sorts alphabetically (A-Z) by default.
- Can sort numerically or based on specific columns.

Common Options:

- `-n`: Sorts data numerically.
- `-r`: Sorts in reverse order (descending).
- `-k <column number>`: Sorts based on a specific column (useful for files with tabular data).
- `-o <output file>`: Saves the sorted output to a specified file.
- `-c`: Checks if the file is already sorted. No output if sorted, displays unsorted lines otherwise.

6.

[udaykiran@localhost ~]\$ cat table				
S.No	Name	Student ID	Email Id	Mobile
1.	Revathi	SEAI11	Revathi@seai.com	123456
2.	Srujana	SEAI12	Srujana@gmail.com	456565
3.	Leela	SEAI13	Leela@seai.com	555659
4.	Hemalatha	SEAI14	Hemalatha@seai.com	9659656
5.	Sneha	SEAI15	Sneha@gmail.com	59598666
6.	Naga Koushik	SEAI16	NagaKoushik@yahoo.com	8956556
7.	Uday Kiran	SEAI17	Uday Kiran@gmail.com	6569995
8.	Hemadri	SEAI18	Hemadri@seai.com	5656555
9.	Apsa	SEAI19	Apsa@yahoo.com	55559555
10.	Harshini	SEAI20	Harshini@yahoo.com	5625665

[udaykiran@localhost ~]\$ sort -k1 table				
S.No	Name	Student ID	Email Id	Mobile
10.	Harshini	SEAI20	Harshini@yahoo.com	5625665
1.	Revathi	SEAI11	Revathi@seai.com	123456
2.	Srujana	SEAI12	Srujana@gmail.com	456565
3.	Leela	SEAI13	Leela@seai.com	555659
4.	Hemalatha	SEAI14	Hemalatha@seai.com	9659656
5.	Sneha	SEAI15	Sneha@gmail.com	59598666
6.	Naga Koushik	SEAI16	NagaKoushik@yahoo.com	8956556
7.	Uday Kiran	SEAI17	Uday Kiran@gmail.com	6569995
8.	Hemadri	SEAI18	Hemadri@seai.com	5656555
9.	Apsa	SEAI19	Apsa@yahoo.com	55559555

[udaykiran@localhost ~]\$ sort -k2 table				
S.No	Name	Student ID	Email Id	Mobile
9.	Apsa	SEAI19	Apsa@yahoo.com	55559555
10.	Harshini	SEAI20	Harshini@yahoo.com	5625665
8.	Hemadri	SEAI18	Hemadri@seai.com	5656555
4.	Hemalatha	SEAI14	Hemalatha@seai.com	9659656
3.	Leela	SEAI13	Leela@seai.com	555659
6.	Naga Koushik	SEAI16	NagaKoushik@yahoo.com	8956556
S.No	Name	Student ID	Email Id	Mobile
1.	Revathi	SEAI11	Revathi@seai.com	123456
5.	Sneha	SEAI15	Sneha@gmail.com	59598666
2.	Srujana	SEAI12	Srujana@gmail.com	456565
7.	Uday Kiran	SEAI17	Uday Kiran@gmail.com	6569995

```
[udaykiran@localhost ~]$ sort -k3 table
1. Revathi      SEAI11      Revathi@seai.com      123456
2. Srujana       SEAI12      Srujana@gmail.com    456565
3. Leela         SEAI13      Leela@seai.com      555659
4. Hemalatha    SEAI14      Hemalatha@seai.com   9659656
5. Sneha         SEAI15      Sneha@gmail.com     59598666
6. NagaKoushik  SEAI16      NagaKoushik@yahoo.com 8956556
7. Udaykiran    SEAI17      Uday Kiran@gmail.com 6569995
8. Hemadri       SEAI18      Hemadri@seai.com    5656555
9. Apsa          SEAI19      Apsa@yahoo.com     55559555
10. Harshini     SEAI20      Harshini@yahoo.com 5625665
S.No Name        Student_ID Email_Id           Mobile
```

```
[udaykiran@localhost ~]$ sort -k4 table
9. Apsa          SEAI19      Apsa@yahoo.com     55559555
S.No Name        Student_ID Email_Id           Mobile
10. Harshini     SEAI20      Harshini@yahoo.com 5625665
8. Hemadri       SEAI18      Hemadri@seai.com    5656555
4. Hemalatha    SEAI14      Hemalatha@seai.com   9659656
3. Leela         SEAI13      Leela@seai.com      555659
6. NagaKoushik  SEAI16      NagaKoushik@yahoo.com 8956556
1. Revathi       SEAI11      Revathi@seai.com    123456
5. Sneha         SEAI15      Sneha@gmail.com     59598666
2. Srujana       SEAI12      Srujana@gmail.com   456565
7. Udaykiran    SEAI17      Uday Kiran@gmail.com 6569995
```

```
[udaykiran@localhost ~]$ sort -k5 table
1. Revathi      SEAI11      Revathi@seai.com      123456
2. Srujana       SEAI12      Srujana@gmail.com    456565
9. Apsa          SEAI19      Apsa@yahoo.com     55559555
3. Leela         SEAI13      Leela@seai.com      555659
10. Harshini     SEAI20      Harshini@yahoo.com 5625665
8. Hemadri       SEAI18      Hemadri@seai.com    5656555
5. Sneha         SEAI15      Sneha@gmail.com     59598666
6. NagaKoushik  SEAI16      NagaKoushik@yahoo.com 8956556
4. Hemalatha    SEAI14      Hemalatha@seai.com   9659656
7. Udaykiran    SEAI17      Uday Kiran@gmail.com 6569995
S.No Name        Student ID Email Id           Mobile
```

b.

```
[udaykiran@localhost ~]$ grep -v '@seai.com' table |awk '{print $2}'
Name
Srujana
Sneha
NagaKoushik
Udaykiran
Apsa
Harshini
[udaykiran@localhost ~]$ █
```

c.

```
[udaykiran@localhost ~]$ sed -E 's/gmail|yahoo/seai/g' table
S.No Name Student_ID Email_Id Mobile
1. Revathi SEAI11 Revathi@seai.com 123456
2. Srujana SEAI12 Srujana@seai.com 456565
3. Leela SEAI13 Leela@seai.com 555659
4. Hemalatha SEAI14 Hemalatha@seai.com 9659656
5. Sneha SEAI15 Sneha@seai.com 59598666
6. NagaKoushik SEAI16 NagaKoushik@seai.com 8956556
7. Udaykiran SEAI17 Uday Kiran@seai.com 6569995
8. Hemadri SEAI18 Hemadri@seai.com 5656555
9. Apsa SEAI19 Apsa@seai.com 55559555
10. Harshini SEAI20 Harshini@seai.com 5625665
[udaykiran@localhost ~]$
```

d.

```
[udaykiran@localhost ~]$ grep '6$' table
1. Revathi SEAI11 Revathi@seai.com 123456
4. Hemalatha SEAI14 Hemalatha@seai.com 9659656
5. Sneha SEAI15 Sneha@gmail.com 59598666
6. NagaKoushik SEAI16 NagaKoushik@yahoo.com 8956556
```

7.Explain the use of “grep” command in file processing. Explain all options with examples

The grep command in Unix/Linux is a powerful tool used for searching and filtering text within files. It stands for "Global Regular Expression Print." grep searches through the contents of files for lines that match a specified pattern and prints the matching lines. Here's a detailed explanation of the grep command, its options, and examples.

Basic Syntax

grep [options] pattern [file...]

Common Options:

- i or --ignore-case: Perform a case-insensitive search.
- v or --invert-match: Invert the match, i.e., show lines that do not match the pattern.
- n or --line-number: Display line numbers along with the matched lines.
- c or --count: Display only the count of matching lines.
- r or --recursive: Recursively search subdirectories.
- w or --word-regexp: Match whole words only.
- l or --files-with-matches: Display only the names of files with matching lines.
- L or --files-without-match: Display only the names of files without matching lines.
- e simple regex pattern

```
[udaykiran@localhost ~]$ grep -n 'seai' table
2:1. Revathi    SEAI11    Revathi@seai.com      123456
4:3. Leela       SEAI13    Leela@seai.com        555659
5:4. Hemalatha   SEAI14    Hemalatha@seai.com     9659656
9:8. Hemadri     SEAI18    Hemadri@seai.com      5656555
[udaykiran@localhost ~]$ grep -ne 'seai' table
2:1. Revathi    SEAI11    Revathi@seai.com      123456
4:3. Leela       SEAI13    Leela@seai.com        555659
5:4. Hemalatha   SEAI14    Hemalatha@seai.com     9659656
9:8. Hemadri     SEAI18    Hemadri@seai.com      5656555
[udaykiran@localhost ~]$ grep -v 'seai' table
S.No Name      Student_ID Email_Id           Mobile
2. Srujana    SEAI12    Srujana@gmail.com  456565
5. Sneha      SEAI15    Sneha@gmail.com    59598666
6. NagaKoushik SEAI16    NagaKoushik@yahoo.com 8956556
7. UdayKiran   SEAI17    Uday Kiran@gmail.com 6569995
9. Apsa        SEAI19    Apsa@yahoo.com     55559555
10. Harshini   SEAI20   Harshini@yahoo.com  5625665
[udaykiran@localhost ~]$ grep -r 'seai' table
1. Revathi    SEAI11    Revathi@seai.com      123456
3. Leela       SEAI13    Leela@seai.com        555659
4. Hemalatha   SEAI14    Hemalatha@seai.com     9659656
8. Hemadri     SEAI18    Hemadri@seai.com      5656555
[udaykiran@localhost ~]$ grep -i 'seai' table
1. Revathi    SEAI11   Revathi@seai.com      123456
2. Srujana    SEAI12   Srujana@gmail.com  456565
3. Leela       SEAI13   Leela@seai.com        555659
4. Hemalatha   SEAI14   Hemalatha@seai.com     9659656
5. Sneha      SEAI15   Sneha@gmail.com    59598666
6. NagaKoushik SEAI16   NagaKoushik@yahoo.com 8956556
7. UdayKiran   SEAI17   Uday Kiran@gmail.com 6569995
8. Hemadri     SEAI18   Hemadri@seai.com      5656555
9. Apsa        SEAI19   Apsa@yahoo.com     55559555
10. Harshini   SEAI20   Harshini@yahoo.com  5625665
```

8.Differentiate between less and more commands.

The less and more commands are both used for viewing the contents of a file one screen at a time.

more Command

- **Basic Functionality:** The more command allows you to view the contents of a file one screen at a time.
- **Navigation:** Primarily forward navigation. Limited backward navigation in some versions.
- **Commands:**

Spacebar	Move to the next screen
Enter	Move to the next line
b	Move back one screen
q	Quit the viewer
/pattern	Search for a pattern. Press n to find the next occurrence

less Command

- **Basic Functionality:** The less command is more advanced, allowing both forward and backward navigation through the file.
- **Navigation:** Both forward and backward navigation.
- **Commands:**

Spacebar	Move to the next screen.
Enter	Move to the next line
b	Move back one screen
/pattern	Search for a pattern. Press n to find the next occurrence.
?pattern	Search backward for a pattern
G	Go to the end of the file.
g	Go to the beginning of the file
h	Display help
q	Quit the viewer

9.Explain the various shortcut keys to use in less command for controlling the movement of files.

The less command in Unix/Linux is a powerful text file viewer that allows you to navigate through files efficiently. Here are the various shortcut keys used in less for controlling the movement within files:

Basic Navigation

- **Spacebar:** Move forward one screen.
- **b:** Move backward one screen.
- **Enter:** Move forward one line.
- **y:** Move backward one line.
- **d:** Move forward half a screen.
- **u:** Move backward half a screen.

Jump to Specific Locations

- **G:** Go to the end of the file.
- **g:** Go to the beginning of the file.
- **<:** Go to the first line of the file.
- **>:** Go to the last line of the file.
- **N:** Go to line number N (replace N with the line number).

Exiting and Help

- **q:** Quit less.
- **h:** Display help screen with all commands.