SHRI G.S .INSTITUTE OF TECHNOLOGY AND SCIENCE

INDORE (M.P)

Department of Computer Technology and Applications

MCA First Year Semester-II

Session:- January 2026 – April 2026

Subject:- Operating System

Assignment -2

Submitted To:-

Prof. Shweta Gupta

Prof. Priyanka Kabra

Submitted By:-

Samarth thombre

0801CA251123

S.NO.	ASSIGNMENT -2	Page No.	Remark
	Tasks:		
1	**Case Study : Student Records Analysis** **Creating a file and a file name students.txt contains student records in the following format:** **RollNo:Name:Department:Marks 101:Rahul:CSE:78** **102:Asha:IT:85** **103:Ravi:CSE:65** **104:Neha:ECE:90** **105:Karan:CSE:78** **Tasks :** ❖ **Display all records of CSE students.** ❖ **Count the total number of students.** ❖ **Display names of students who scored more than 75 marks.** ❖ **Sort the file based on marks (descending order).** ❖ **Find students having duplicate marks.** **Using the grep, wc, awk, sort, uniq commands for this task**	1	
2.	**Case Study : Payroll Processing** **Creating an employee salary file name salary.txt and contains the following records in the following format:** **EmpID Name Department Salary** **E101 Amit HR 30000** **E102 Rina CTA 45000** **E103 Suresh CTA 50000** **E104 Meena HR 35000** **Perform the following task :** ❖ **Display employees belonging to the IT department.** ❖ **Print only Name and Salary columns.** ❖ **Increase salary by 10% for IT employees.** ❖ **Calculate the average salary.** ❖ **Count employees in each department.** **Using the awk ,cut ,grep commands**	2	
3.	**Case study :Automated Monitoring Report** **A daily monitoring report is required for system health.** **Tasks** ❖ **Collect CPU, memory, and disk usage information.** ❖ **Save the report with date and time.** ❖ **Schedule the report generation automatically.** ❖ **View historical monitoring data.** **Using top, free, df, date, cron, sar for complete these tasks.**	3	
4	**Case Study :File Copy, Move, and Rename** **Scenario** **Project files need to be organized into appropriate directories.** **Tasks** ❖ **Create a directory project.** ❖ **Copy report.txt into the project directory.** ❖ **Rename report.txt to final_report.txt.** ❖ **Move final_report.txt to a directory named backup.** ❖ **Verify the file location.** **Using mkdir, cp, mv, ls commands for this .**	6	
5	**Case Study :File Deletion and Recovery Awareness** **Scenario** **Old and unused files must be removed from the system**	6	

	Tasks ❖ **Delete a file named old.txt.** ❖ **Remove an empty directory.** ❖ **Remove a directory containing files.** ❖ **List files before and after deletion.** ❖ **Explain why deleted files cannot be easily recovered.** **Using rm, rmdir, ls for this task .**		
6	**Check Current User and Group:** Open a terminal and run the command to check your current user and group. **Setting Up the Environment:** ❖ Create a directory named file_permission_lab. Inside this directory, create three files: file1.txt, file2.txt, and file3.txt.	7	
7	**Viewing File Permissions:** Use the command to view the permissions of the files you created. Document the output.	8	
8	**Changing File Permissions:** ❖ Change the permissions of file1.txt to allow the owner to read and write, the group to read, and others to have no permissions. Use both symbolic and numeric methods. ❖ Change the permissions of file2.txt to allow everyone to read and execute, but not write. Change the permissions of file3.txt to allow only the owner to read, write, and execute.	8	
9	**Documenting Changes:** After each permission change, use ls -l to document the new permissions for each file.	9	
10	**File Handling Tasks:** 1. Create a script called file_handling.sh that: o Creates a directory named test_dir. o Creates a file named test_file.txt inside test_dir. o Writes "This is a test file." to the file. o Displays the contents of the file. o Deletes the file and the directory. Make the script executable and run it with different arguments.	9	
11	**Change Ownership:** ❖ If you have another user account on the system, change the ownership of file1.txt to that user (replace otheruser with the actual username)	10	
12	**Cleanup:** After completing the tasks, remove the file_permission_lab directory and its contents	11	
13	**File commands :** **Create a Directory:** ❖ Create a new directory named file_commands_lab in your home directory. **Navigate to the Directory:** ❖ Change to the newly created directory ❖ Create five text files named file1.txt, file2.txt, file3.txt, file4.txt, and file5.txt. Use the command to list the files in the directory.	11	
14	**Write to Files:**	12	

	❖ Use the echo command to write "This is file 1" text into file1.txt on the terminal not in the directly writing text file. Append text to file2.txt.write the contents "This is file 2" in the file2.txt and that append.		
15	❖ Use the cat command to display the contents of file1.txt. ❖ Use the less command to view the contents of 'file2.txt'	12	
16	**Copy Files:** ❖ Copy 'file1.txt' to create a new file named 'file1_copy.txt' ❖ Rename 'file2.txt ' to 'file2_renamed.txt' ❖ Delete 'file3.txt'.: Change to the 'backup directory and list the files:	13	
17	**Create a Compressed Archive:** Go back to the 'file_commands_lab' directory and create a compressed archive of the backup directory.	13	
18	**Extract the Archive:** Create a new directory named 'extracted_backup' and extract the contents of 'backup.tar.gz' into it:	13	
19	**Verify Extraction:** List the contents of the 'extracted_backup' directory to verify that the files were extracted correctly	14	
20	Create a text file named " count.txt " and write the 10 line sentences and use the command to count the word ,lines , character and also use command for highlighting a patten in this file .	14	

Q1.Case Study : Student Records Analysis
Creating a file and a file name students.txt contains student records in the following format:
RollNo:Name:Department:Marks
101:Rahul:CSE:78
102:Asha:IT:85
103:Ravi:CSE:65
104:Neha:ECE:90
105:Karan:CSE:78
Tasks :
1. Display all records of CSE students.
→ **grep ":CSE:" students.txt**

```
sam@ps:~/Desktop/collage/Assisment/DBMS$ grep ":CSE:" students.txt
101:Rahul:CSE:78
103:Ravi:CSE:65
105:Karan:CSE:78
```

2. Count the total number of students.
→ **wc -l students.txt**

```
sam@ps:~/Desktop/collage/Assisment/DBMS$ wc -l students.txt
6 students.txt
sam@ps:~/Desktop/collage/Assisment/DBMS$ awk -F: '$4 > 75 { print $
Name
```

→ **awk -F: '$4 > 75 { print $2 }' students.txt**

```
sam@ps:~/Desktop/collage/Assisment/DBMS$ awk -F: '$4 > 75 { print $2 }' students.txt
Name
Rahul
Asha
Neha
Karan
```

4. Sort the file based on marks (descending order).
→ **sort -t: -k4,4nr students.txt**

```
sam@ps:~/Desktop/collage/Assisment/DBMS$ sort -t: -k4,4nr students.txt
104:Neha:ECE:90
102:Asha:IT:85
101:Rahul:CSE:78
105:Karan:CSE:78
103:Ravi:CSE:65
RollNo:Name:Department:Marks
```

5. Find students having duplicate marks.
→ **awk -F: '{print $4}' students.txt | sort | uniq -d**

```
sam@ps:~/Desktop/collage/Assisment/DBMS$ awk -F: '{print $4}' students.txt | sort | uniq -d
78
```

Q2. Use Study : Payroll Processing
Creating an employee salary file name salary.txt and contains the following records in the following format:
EmpID Name Department Salary
E101 Amit HR 30000
E102 Rina IT 45000
E103 Suresh IT 50000
E104 Meena HR 35000

Perform the following task :
- Display employees belonging to the IT department.

→ **grep " IT " salary.txt**

```
sam@ps:~/Desktop/collage/Assisment/DBMS$ grep " IT " salary.txt
E102 Rina IT 45000
E103 Suresh IT 50000
```

- Print only Name and Salary columns.

→ **awk '{ print $2, $4 }' salary.txt**

```
sam@ps:~/Desktop/collage/Assisment/DBMS$ awk '{ print $2, $4 }' salary.txt
Name Salary
Amit 30000
Rina 45000
Suresh 50000
Meena 35000
```

- Increase salary by 10% for IT employees.

→ **awk '{ if ($3 == "IT")**
printf "%s %s %s %.0f\n", $1, $2, $3, $4 * 1.10
else print }' salary.txt

```
sam@ps:~/Desktop/collage/Assisment/DBMS$ awk '{ if ($3 == "IT")
        printf "%s %s %s %.0f\n", $1, $2, $3, $4 * 1.10
        else print  }' salary.txt
EmpID Name Department Salary
E101 Amit HR 30000
E102 Rina IT 49500
E103 Suresh IT 55000
E104 Meena HR 35000
```

🚌 Calculate the average salary.
→ awk '{ sum += $4 } END { print "Average Salary =", sum/NR }' salary.txt

```
sam@ps:~/Desktop/collage/Assisment/DBMS$ awk '{ sum += $4 } END { print "Average Salary =", sum/NR }'   salary.txt
Average Salary = 32000
```

🚌 Count employees in each department.
→ awk '{ count[$3]++ } END { for (d in count) print d, count[d] }' salary.txt

```
sam@ps:~/Desktop/collage/Assisment/DBMS$ awk '{ count[$3]++ } END { for (d in count) print d, count[d] }' salary.txt
IT 2
Department 1
HR 2
```

Q3.Case study :Automated Monitoring Report
A daily monitoring report is required for system health.
Tasks
🚌 Collect CPU, memory, and disk usage information.
→ top -b -n 1 | head -5 && free -h && df -h

```
sam@pc:~$ top -b -n 1 | head -5
top - 21:02:28 up  1:26,  1 user,  load average: 0.21, 1.20, 3.24
Tasks: 333 total,   1 running, 331 sleeping,   0 stopped,   1 zombie
%Cpu(s):  0.0 us,  1.6 sy,  0.0 ni, 98.4 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :   7836.5 total,   2746.0 free,   2333.8 used,   3092.4 buff/cache
MiB Swap:   4096.0 total,   4096.0 free,      0.0 used.   5502.7 avail Mem
sam@pc:~$ free -h
               total        used        free      shared  buff/cache   available
Mem:           7.7Gi       2.3Gi       2.7Gi        68Mi       3.0Gi       5.4Gi
Swap:          4.0Gi          0B       4.0Gi
sam@pc:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
tmpfs           784M  2.4M  782M   1% /run
/dev/nvme0n1p1  116G   75G   36G  68% /
tmpfs           3.9G  1.5M  3.9G   1% /dev/shm
tmpfs           5.0M   16K  5.0M   1% /run/lock
efivarfs        256K   46K  206K  19% /sys/firmware/efi/efivars
/dev/nvme0n1p2  1.1G   38M  1.1G   4% /boot/efi
tmpfs           784M  2.6M  782M   1% /run/user/1000
sam@pc:~$
```

🚌 Save the report with date and time.

→ touch report.txt

→ echo "System Monitoring Report - $(date)" > report.txt

echo "----------" >> report.txt
echo "CPU Usage:" >> report.txt
top -b -n 1 | head -5 >> report.txt
echo "Memory Usage:" >> report.txt
free -h >> report.txt
echo "Disk Usage:" >> report.txt
df -h >> report.txt

→ cat report.txt

```
sam@ps:~/Desktop/collage/Assisment/DBMS$ date
Sat Jan 24 09:49:01 PM IST 2026
sam@ps:~/Desktop/collage/Assisment/DBMS$ ./report.sh
sam@ps:~/Desktop/collage/Assisment/DBMS$ cat report.txt
System Monitoring Report - Sat Jan 24 09:49:05 PM IST 2026
----------
CPU Usage:
top - 21:49:06 up  2:17,  1 user,  load average: 0.83, 1.66, 1.46
Tasks: 355 total,   1 running, 354 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.8 us,  1.6 sy,  3.2 ni, 93.5 id,  0.8 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :   7836.5 total,    578.0 free,   3566.0 used,   3941.2 buff/cache
MiB Swap:   4096.0 total,   4096.0 free,      0.0 used.   4270.5 avail Mem
Memory Usage:
               total        used        free      shared  buff/cache   available
Mem:           7.7Gi       3.5Gi       577Mi       155Mi       3.8Gi       4.2Gi
Swap:          4.0Gi          0B       4.0Gi
Disk Usage:
Filesystem      Size  Used Avail Use% Mounted on
tmpfs           784M  2.4M  782M   1% /run
/dev/nvme0n1p1  116G   89G   22G  81% /
tmpfs           3.9G   18M  3.9G   1% /dev/shm
tmpfs           5.0M   16K  5.0M   1% /run/lock
efivarfs        256K   46K  206K  19% /sys/firmware/efi/efivars
/dev/nvme0n1p2  1.1G   38M  1.1G   4% /boot/efi
tmpfs           784M  148K  784M   1% /run/user/1000
```

🚌 Schedule the report generation automatically.

→ nano monitor.sh

→ #!/bin/bash

report="report_$(date +'%Y-%m-%d_%H-%M-%S').txt"

echo "System Monitoring Report - $(date)" > $report
echo "----------" >> $report
echo "CPU Usage:" >> $report
top -b -n 1 | head -5 >> $report

echo "Memory Usage:" >> $report
free -h >> $report

echo "Disk Usage:" >> $report
df -h >> $report

→ **chmod +x monitor.sh**
→ **crontab -e**
→ **0 9 * * * /home/sam/monitor.sh**

```
sam@ps:~/Desktop/collage/Assisment/DBMS$ chmod +x monitor.sh
sam@ps:~/Desktop/collage/Assisment/DBMS$ nano monitor.sh
sam@ps:~/Desktop/collage/Assisment/DBMS$ chmod +x monitor.sh
sam@ps:~/Desktop/collage/Assisment/DBMS$ crontab -e
no crontab for sam - using an empty one

Select an editor.  To change later, run 'select-editor'.
  1. /bin/nano        <---- easiest
  2. /usr/bin/vim.tiny
  3. /usr/bin/cursor
  4. /bin/ed

Choose 1-4 [1]: 1
crontab: installing new crontab
```

🚌 View historical monitoring data.
→ **ls report_*.txt**
→ **cat report_2026-01-24_22-10-31.txt**

```
sam@ps:~/Desktop/collage/Assisment/DBMS$ cat report_2026-01-24_22-10-31.txt
System Monitoring Report - Sat Jan 24 10:10:31 PM IST 2026
------------------------------
CPU Usage:
top - 22:10:32 up  2:39,  1 user,  load average: 1.43, 1.80, 1.83
Tasks: 358 total,   1 running, 357 sleeping,   0 stopped,   0 zombie
%Cpu(s):  1.6 us,  1.6 sy,  0.8 ni, 96.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :   7836.5 total,    313.9 free,   3611.8 used,   4123.0 buff/cache
MiB Swap:   4096.0 total,   4096.0 free,      0.0 used.   4224.8 avail Mem
Memory Usage:
               total        used        free      shared  buff/cache   available
Mem:           7.7Gi       3.5Gi       315Mi       129Mi       4.0Gi       4.1Gi
Swap:          4.0Gi          0B       4.0Gi
Disk Usage:
Filesystem      Size  Used Avail Use% Mounted on
tmpfs           784M  2.4M  782M   1% /run
/dev/nvme0n1p1  116G   89G   22G  81% /
tmpfs           3.9G   18M  3.9G   1% /dev/shm
tmpfs           5.0M   16K  5.0M   1% /run/lock
efivarfs        256K   46K  206K  19% /sys/firmware/efi/efivars
/dev/nvme0n1p2  1.1G   38M  1.1G   4% /boot/efi
tmpfs           784M  144K  784M   1% /run/user/1000
```

Q4.Case Study :File Copy, Move, and Rename
Scenario
Project files need to be organized into appropriate directories.
Tasks
- Create a directory project.

→mkdir project

- Copy report.txt into the project directory.

→cp report.txt project/

- Rename report.txt to final_report.txt.

→mv project/report.txt project/final_report.txt

- Move final_report.txt to a directory named backup.

→mkdir backup
→ mv project/final_report.txt backup/

- Verify the file location.

→ls project
→ ls backup

```
sam@ps:~/Desktop/collage/Assisment/DBMS$ mkdir project
sam@ps:~/Desktop/collage/Assisment/DBMS$ cp report.txt project/
sam@ps:~/Desktop/collage/Assisment/DBMS$ mv project/report.txt project/final_report.txt
sam@ps:~/Desktop/collage/Assisment/DBMS$ mkdir backup
sam@ps:~/Desktop/collage/Assisment/DBMS$ mv project/final_report.txt backup/
sam@ps:~/Desktop/collage/Assisment/DBMS$ ls project/
sam@ps:~/Desktop/collage/Assisment/DBMS$ ls backup/
final_report.txt
```

Q5.Case Study :File Deletion and Recovery Awareness
Scenario
Old and unused files must be removed from the system.
Tasks
- Delete a file named old.txt.

→ rm old.txt

- Remove an empty directory.

→ rmdir temp

- Remove a directory containing files.

→ rm -r file

- List files before and after deletion.

→ ls
→ rm -r file
→ ls

```
sam@ps:~/Desktop/collage/Assisment/DBMS$ ls
 backup                os_assiment_2_ROLLNO-123.pdf            report.sh
 file                  project                                 report.txt
 monitor.sh            report_2026-01-24_22-07-58.txt          salary.txt
 os_assiment_2.docx    report_2026-01-24_22-10-31.txt          students.txt
 os_assiment_2.odt    'report=report_2026-01-24_22-09-35.txt'
sam@ps:~/Desktop/collage/Assisment/DBMS$ rm -r file
sam@ps:~/Desktop/collage/Assisment/DBMS$ ls
 backup                          project                                 report.txt
 monitor.sh                      report_2026-01-24_22-07-58.txt          salary.txt
 os_assiment_2.docx              report_2026-01-24_22-10-31.txt          students.txt
 os_assiment_2.odt              'report=report_2026-01-24_22-09-35.txt'
 os_assiment_2_ROLLNO-123.pdf    report.sh
```

🚍 Explain why deleted files cannot be easily recovered.
→ When a file is deleted using rm, Linux does not move it to a recycle bin.

- The file's directory entry and inode references are removed.
- The actual data blocks are marked as free space.
- New data can overwrite these blocks at any time.
- Once overwritten, recovery becomes nearly impossible.

Q6.Check Current User and Group:
Open a terminal and run the command to check your current user and group.
→ **whoami**
→ **group**

Setting Up the Environment:
🚍 Create a directory named file_permission_lab.
→ **mkdir file_permission_lab**

🚍 Inside this directory, create three files: file1.txt, file2.txt, and file3.txt.
→ **cd file_permission_lab**
→ **touch file1.txt file2.txt file3.txt**

```
sam@ps:~/Desktop/collage/Assisment/DBMS$ whoami
sam
sam@ps:~/Desktop/collage/Assisment/DBMS$ groups
sam adm cdrom sudo dip plugdev users lpadmin
sam@ps:~/Desktop/collage/Assisment/DBMS$ cd file_permission_lab/
bash: cd: file_permission_lab/: No such file or directory
sam@ps:~/Desktop/collage/Assisment/DBMS$ whoami
sam
sam@ps:~/Desktop/collage/Assisment/DBMS$ groups
sam adm cdrom sudo dip plugdev users lpadmin
sam@ps:~/Desktop/collage/Assisment/DBMS$ mkdir file_permission_lab
sam@ps:~/Desktop/collage/Assisment/DBMS$ cd file_permission_lab/
sam@ps:~/Desktop/collage/Assisment/DBMS/file_permission_lab$ touch file1.txt file2.txt file3.txt
sam@ps:~/Desktop/collage/Assisment/DBMS/file_permission_lab$ ls
file1.txt  file2.txt  file3.txt
```

Q7. Viewing File Permissions:
🚌 Use the command to view the permissions of the files you created. Document the output.
→ **ls -l**

```
sam@ps:~/Desktop/collage/Assisment/DBMS$ ls -l
total 240
drwxrwxr-x 2 sam sam   4096 Jan 24 22:14  backup
drwxrwxr-x 2 sam sam   4096 Jan 24 22:31  file_permission_lab
-rwxrwxr-x 1 sam sam    349 Jan 24 22:10  monitor.sh
-rw-rw-r-- 1 sam sam  11844 Jan 24 21:08  os_assiment_2.docx
-rw-rw-r-- 1 sam sam  42997 Jan 24 21:10  os_assiment_2.odt
-rw-rw-r-- 1 sam sam 146810 Jan 24 21:10  os_assiment_2_ROLLNO-123.pdf
drwxrwxr-x 2 sam sam   4096 Jan 24 22:14  project
-rw-rw-r-- 1 sam sam   1098 Jan 24 22:07  report_2026-01-24_22-07-58.txt
-rw-rw-r-- 1 sam sam   1098 Jan 24 22:10  report_2026-01-24_22-10-31.txt
-rw-rw-r-- 1 sam sam      0 Jan 24 22:09 'report=report_2026-01-24_22-09-35.txt'
-rwxrwxr-x 1 sam sam    346 Jan 24 21:36  report.sh
-rw-rw-r-- 1 sam sam   1098 Jan 24 21:49  report.txt
-rw-rw-r-- 1 sam sam    108 Jan 24 22:20  salary.txt
-rw-rw-r-- 1 sam sam    110 Jan 24 19:37  students.txt
```

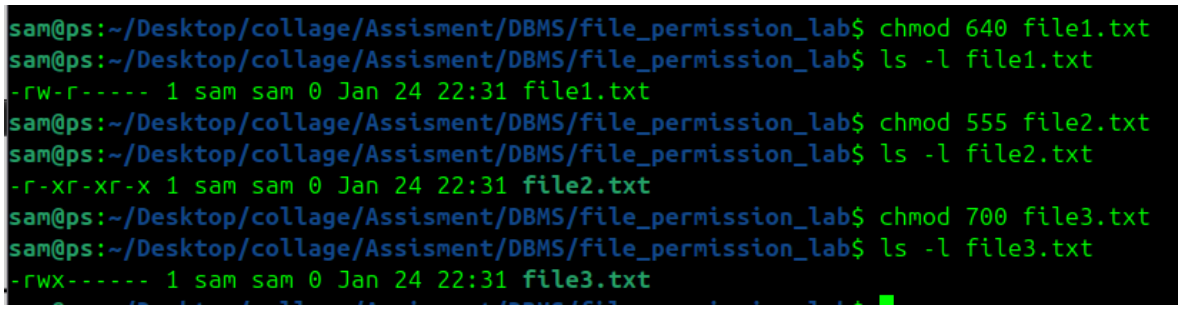
Q8.Changing File Permissions:
🚌 Change the permissions of file1.txt to allow the owner to read and write, the group to read, and others to have no permissions. Use both symbolic and numeric methods.
→ **chmod u=rw,g=r,o= file1.txt**
→ **chmod 640 file1.txt**

🚌 Change the permissions of file2.txt to allow everyone to read and execute, but not write.
→ **chmod a=rx file2.txt**
→ **chmod 555 file2.txt**

🚌 Change the permissions of file3.txt to allow only the owner to read, write, and execute.
→ **chmod u=rwx,g=,o= file3.txt**
→ **chmod 700 file3.txt**

```
sam@ps:~/Desktop/collage/Assisment/DBMS/file_permission_lab$ ls -l
total 0
-rw-r----- 1 sam sam 0 Jan 24 22:31 file1.txt
-rw-rw-r-- 1 sam sam 0 Jan 24 22:31 file2.txt
-rw-rw-r-- 1 sam sam 0 Jan 24 22:31 file3.txt
sam@ps:~/Desktop/collage/Assisment/DBMS/file_permission_lab$ chmod a=rx file2.txt
sam@ps:~/Desktop/collage/Assisment/DBMS/file_permission_lab$ chmod 700 file3.txt
sam@ps:~/Desktop/collage/Assisment/DBMS/file_permission_lab$ ls -l
total 0
-rw-r----- 1 sam sam 0 Jan 24 22:31 file1.txt
-r-xr-xr-x 1 sam sam 0 Jan 24 22:31 file2.txt
-rwx------ 1 sam sam 0 Jan 24 22:31 file3.txt
```

Q9.Documenting Changes:
🚌 After each permission change, use ls -l to document the new permissions for each file
→ **chmod 640 file1.txt**
→ **ls -l file1.txt**
→ **chmod 555 file2.txt**
→ **ls -l file2.txt**
→ **chmod 700 file3.txt**
→ **ls -l file3.txt**
→ **ls -l**



Q10.File Handling Tasks:
1. Create a script called file_handling.sh that:
   - o Creates a directory named test_dir.
   - o Creates a file named test_file.txt inside test_dir.
   - o Writes "This is a test file." to the file.
   - o Displays the contents of the file.
   - o Deletes the file and the directory.

→ **nano file_handling.sh**
→

```
#!/bin/bash
DIR_NAME=${1:-test_dir}
FILE_NAME=${2:-test_file.txt}

echo "Creating directory: $DIR_NAME"
mkdir $DIR_NAME

echo "Creating file: $FILE_NAME inside $DIR_NAME"
touch $DIR_NAME/$FILE_NAME

echo "Writing content to file..."
echo "This is a test file." > $DIR_NAME/$FILE_NAME

echo "Displaying file contents:"
cat $DIR_NAME/$FILE_NAME
```

```
echo "Deleting file and directory..."
rm $DIR_NAME/$FILE_NAME
rmdir $DIR_NAME

echo "Task completed successfully."
```

2. Make the script executable and run it with different arguments.

→ **chmod +x file_handling.sh**

```
sam@pc:~/Desktop/collage/Assisment$ nano file_handling.sh
sam@pc:~/Desktop/collage/Assisment$ chmod +x file_handling.sh
sam@pc:~/Desktop/collage/Assisment$ ls -l
total 8
drwxrwxr-x 2 sam sam 4096 Feb  1 20:54 DBMS
-rwxrwxr-x 1 sam sam  461 Feb  1 21:11 file_handling.sh
sam@pc:~/Desktop/collage/Assisment$ ./file_handling.sh
Creating directory: test_dir
Creating file: test_file.txt inside test_dir
Writing content to file...
Displaying file contents:
This is a test file.
Deleting file and directory...
Task completed successfully.
```

Q11.Change Ownership:

🚌 If you have another user account on the system, change the ownership of file1.txt to that user (replace otheruser with the actual username)

→ **sudo chown otheruser file1.txt**

```
sam@pc:~/Desktop/collage/Assisment$ sudo chown sam file1.txt
sam@pc:~/Desktop/collage/Assisment$ ls -l
total 8
drwxrwxr-x 2 sam sam 4096 Feb  1 20:54 DBMS
-rw-rw-r-- 1 sam sam    0 Feb  1 21:14 file1.txt
-rwxrwxr-x 1 sam sam  461 Feb  1 21:11 file_handling.sh
```

Q12.Cleanup:

- After completing the tasks, remove the file_permission_lab directory and its contents

→ **rm -r file_permission_lab**

```
sam@pc:~/Desktop/collage/Assisment$ ls -l
total 12
drwxrwxr-x 2 sam sam 4096 Feb  1 20:54 DBMS
-rw-rw-r-- 1 sam sam    0 Feb  1 21:14 file1.txt
-rwxrwxr-x 1 sam sam  461 Feb  1 21:11 file_handling.sh
drwxrwxr-x 2 sam sam 4096 Feb  1 21:16 file_permission_lab
sam@pc:~/Desktop/collage/Assisment$ rm -r file_permission_lab
sam@pc:~/Desktop/collage/Assisment$ ls -l
total 8
drwxrwxr-x 2 sam sam 4096 Feb  1 20:54 DBMS
-rw-rw-r-- 1 sam sam    0 Feb  1 21:14 file1.txt
-rwxrwxr-x 1 sam sam  461 Feb  1 21:11 file_handling.sh
```

Q13.File commands :

Create a Directory:

- Create a new directory named file_commands_lab in your home directory.

→ **mkdir ~/file_commands_lab**

Navigate to the Directory:

- Change to the newly created directory

→ **cd ~/file_commands_lab**

- Create five text files named file1.txt, file2.txt, file3.txt, file4.txt, and file5.txt.

→ **touch file1.txt file2.txt file3.txt file4.txt file5.txt**

- Use the command to list the files in the directory.

→ **ls**

```
sam@ps:~/Desktop/collage/Assisment/DBMS/file_permission_lab$ mkdir ~/file_commands_lab
sam@ps:~/Desktop/collage/Assisment/DBMS/file_permission_lab$ cd ~/file_commands_lab
sam@ps:~/file_commands_lab$ touch file1.txt file2.txt file3.txt file4.txt file5.txt
sam@ps:~/file_commands_lab$ ls
file1.txt  file2.txt  file3.txt  file4.txt  file5.txt
```

Q14.Write to Files:
🚌 Use the echo command to write “This is file 1” text into file1.txt on the terminal not in the directly writing text file.
→ **echo "This is file 1" > file1.txt**

🚌 Append text to file2.txt.write the contents "This is file 2” in the file2.txt and that append.
→ **echo "This is file 2" >> file2.txt**

Q15.🚌 Use the cat command to display the contents of file1.txt.
→ **cat file1.txt**

🚌 Use the less command to view the contents of ‘file2.txt’
→ **less file2.txt**

```
sam@ps:~/file_commands_lab$ echo "This is file 1" > file1.txt
sam@ps:~/file_commands_lab$ echo "This is file 2" > file2.txt
sam@ps:~/file_commands_lab$ cat file1.txt
This is file 1
sam@ps:~/file_commands_lab$ less file2.txt

[1]+  Stopped                 less file2.txt
```

Q16.Copy Files:
🚍 Copy ‘file1.txt’ to create a new file named ‘file1_copy.txt’
→ **cp file1.txt file1_copy.txt**

🚍 Rename ‘file2.txt ‘ to ‘file2_renamed.txt’
→ **mv file2.txt file2_renamed.txt**

🚍 Delete ‘file3.txt’.:
→ **rm file3.txt**

🚍 Change to the ‘backup directory and list the files:
→ **cd backup**
→ **ls**

```
sam@ps:~/file_commands_lab$ cp file1.txt file1_copy.txt
sam@ps:~/file_commands_lab$ mv file2.txt file2_renamed.txt
sam@ps:~/file_commands_lab$ rm file3.txt
sam@ps:~/file_commands_lab$ cd backup
bash: cd: backup: No such file or directory
sam@ps:~/file_commands_lab$ ls
file1_copy.txt  file1.txt  file2_renamed.txt  file4.txt  file5.txt
```

Q17. Create a Compressed Archive:
🚍 Go back to the ‘file_commands_lab’ directory and create a compressed archive of the backup directory.
→ **cd ~/file_commands_lab**
→ **tar -czf backup.tar.gz backup/**

```
sam@pc:~/Desktop/collage/Assisment$ cd file_commands_lab/
sam@pc:~/Desktop/collage/Assisment/file_commands_lab$ tar -czf backup.tar.gz backup
sam@pc:~/Desktop/collage/Assisment/file_commands_lab$ ls
backup  backup.tar.gz
```

Q18.Extract the Archive:
🚍 Create a new directory named ‘extracted_backup’ and extract the contents of ‘backup.tar.gz’ into it
→ **mkdir extracted_backup**
→ **tar -xzvf backup.tar.gz -C extracted_backup/**

```
sam@pc:~/Desktop/collage/Assisment/file_commands_lab$ mkdir extracted_backup
sam@pc:~/Desktop/collage/Assisment/file_commands_lab$ tar -xzvf backup.tar.gz -C extra
backup/
sam@pc:~/Desktop/collage/Assisment/file_commands_lab$ ls
backup  backup.tar.gz  extracted_backup
```

Q19.Verify Extraction:
🚌 List the contents of the 'extracted_backup' directory to verify that the files were extracted correctly
→ **ls extracted_backup**
→ **ls extracted_backup/backup**

```
sam@ps:~/file_commands_lab$ cd ~/file_commands_lab
sam@ps:~/file_commands_lab$ tar -czvf backup.tar.gz backup/
backup/
sam@ps:~/file_commands_lab$ mkdir extracted_backup
sam@ps:~/file_commands_lab$ tar -xzvf backup.tar.gz -C extracted_back
backup/
sam@ps:~/file_commands_lab$ ls extracted_backup
backup
sam@ps:~/file_commands_lab$ ls extracted_backup/backup
```

Q20.Create a text file named " count.txt " and write the 10 line sentences and use the command to count the word ,lines , character and also use command for highlighting a patten in this file .
→ **cat > count.txt**

This is line one.
This is line two.
This is line three.
This is line four.
This is ling five.
This is line six.
This is line seven.
This is line eight.
This is line nine.
This is line ten.

→ **wc -l count.txt**
→ **wc -w count.txt**
→ **wc -c count.txt**

```
sam@ps:~/file_commands_lab$ wc -l count.txt
9 count.txt
sam@ps:~/file_commands_lab$ wc -w count.txt
36 count.txt
sam@ps:~/file_commands_lab$  wc -c count.txt
179 count.txt
```