EXPERIMENT – 1

TO PERFORM FOLLOWING LINUX COMMANDS

1. **To create and remove directory**
2. **To create file and delete file**
3. **File copy and moving**
4. **Listing of files with various options**

1. **Aim: - To create and remove directory**

Procedure:

a. **Creation**

Syntax:

```
mkdir <directory_name>
```

Example:

```
mkdir my_directory
```

b. **Deletion**

Syntax:

```
rmdir <directory_name>
```

Example:

```
rmdir my_directory
```

2. **Aim: - To create file and delete file**

Procedure:

a. **File Creation**

Syntax:

```
cat >  <file_name>
// File content
```

Example:

```
cat > file.txt
Hello World!
```

CTRL + D (save file)

b. File Deletion

Syntax:

```
rm <file_name>
```

Example:

```
rm my_file.txt
```

3. Aim: - File copy and moving

Procedure:

a. Copy file

Syntax:

```
cp <source_file> <new_file_name>
```

Example:

```
cp file.txt  file2.txt
```

b. Moving file

Syntax:

```
mv <source_file> <destination_path >
```

Example:

```
mv file.txt  /user/my_directory/
```

4. Aim: - Listing of files with various options

Procedure:

a. Basic listing files and directories

Command:

```
ls
```

Output:

```
file1.txt  file2.txt  subdirectory
```

b. Show all files included hidden files and directories

Command:

```
ls –a
```

Output:

```
. ..  file1.txt  file2.txt  .hidden_file  subdirectory
```

c. Show detailed information about files and directories

Command:

```
ls –l
```

Output:

```
total 12
-rw-r--r-- 1 user group 1024 Jan 1 10:00 file1.txt
-rw-r--r-- 1 user group 2048 Jan 1 11:30 file2.txt
drwxr-xr-x 2 user group 4096 Jan 1 09:45 subdirectory
```

Result: - Commands executed successfully.

EXPERIMENT - 2

TO PERFORM FOLLOWING HADOOP COMMANDS

1. **Directory creation or deletion**
2. **Putting file from local to Hadoop**
3. **Listing the file with various options**
4. **File copy and moving**

1. **Aim:- Directory creation and deletion**

Procedure:

a) **Creation**

Syntax:

```
hadoop fs -mkdir <directory_path>
```

Example:

```
hadoop fs -mkdir /user/myuser/new_directory
```

b) **Deletion**

Syntax:

```
hadoop fs -rmdir <directory_path>
```

Example:

```
hadoop fs -rmdir /user/myuser/new_directory
```

2. **Aim:- Putting file from local to Hadoop**

Procedure:

Syntax:

```
hadoop fs -put <local_file_path> <hdfs_destination_path>
```

Example:

```
hadoop fs -put /path/to/local/file.txt  /user/myuser/hdfs_directory/file.txt
```

3. **Aim:- Listing the file with various options**

Procedure:

i. Basic listing of files in a directory:

Syntax:

```
hadoop fs -ls /user/myuser/hdfs_directory
```

Output:

```
Found 3 items

-rw-r--r--   3 user supergroup    1234567890 2024-01-08 10:00
/user/myuser/hdfs_directory/file1.txt

-rw-r--r--   3 user supergroup    9876543210 2024-01-07 15:30
/user/myuser/hdfs_directory/file2.txt

drwxr-xr-x   - user supergroup             0 2024-01-06 09:45
/user/myuser/hdfs_directory/subdirectory
```

ii. Long format listing with human readable sizes:

Syntax:

```
hadoop fs -ls -l -h /user/myuser/hdfs_directory
```

Output:

```
Found 3 items

-rw-r--r--   3 user supergroup   1.1G 2024-01-08 10:00
/user/myuser/hdfs_directory/file1.txt

-rw-r--r--   3 user supergroup   9.2G 2024-01-07 15:30
/user/myuser/hdfs_directory/file2.txt

drwxr-xr-x   - user supergroup      0 2024-01-06 09:45
/user/myuser/hdfs_directory/subdirectory
```

iii. Recursive listing of files and subdirectories:

Syntax:

```
hadoop fs -ls -R /user/myuser/hdfs_directory
```

Output:

```
/user/myuser/hdfs_directory/file1.txt

/user/myuser/hdfs_directory/file2.txt

/user/myuser/hdfs_directory/subdirectory:

drwxr-xr-x   - user supergroup    0 2024-01-06 09:45
/user/myuser/hdfs_directory/subdirectory/inner_directory

-rw-r--r--   3 user supergroup  456 2024-01-05 12:00
/user/myuser/hdfs_directory/subdirectory/inner_file.txt
```

4. **Aim:- File copy and moving**

Procedure:

a) File Copy

Syntax:

```
hadoop fs -cp <source_path> <destination_path>
```

Example:

```
hadoop fs -cp /user/myuser/source_directory/file.txt
/user/myuser/destination_directory/file.txt
```

b) File Moving

Syntax:

```
hadoop fs -mv <source_path> <destination_path>
```

Example:

```
hadoop fs -mv /user/myuser/source_directory/file.txt
/user/myuser/destination_directory/new_file_name.txt
```

Result: Hadoop commands executed successfully.

EXPERIMENT – 3

TO PERFORM THE MONGODB QURIES AND TASK

1. **Create a database using MongoDB**
2. **Create a collection**
3. **Insert a document into collection**
4. **Display document**
5. **Deletion of document from collection**
6. **Updating of document**

1. **Aim: - Create a database using MongoDB**

   Procedure:

   Syntax:

   ```
   use <database name>
   ```

   Example:

   ```
   use my_db
   ```

   Output:

   ```
   mongodb> switched to db my_db
   my_db> {ok:1}
   ```

2. **Aim: - Create a collection**

   Procedure:

   Syntax:

   ```
   db.createCollection("<collection name>")
   ```

   Example:

   ```
   db.createCollection("Students")
   show collection
   ```

   Output:

   ```
   my_db> Students
   ```

3. **Aim: - Insert document into collection**

   Procedure:

Syntax:

```
        db.Students.insertMany([{“Reg No”:’01’, “Name”:’Rahul’,”Age”:’34’},
                                {“Reg No”:’02’, “Name”:’Rohit’,”Age”:’19’},
                                {“Reg No”:’03’, “Name”:’Amit’,”Age”:’24’}])
```

Example:

```
        my_db> … … … {
                Acknowledged : true,
                insertedIds : {
                ‘0’ : ObjectId(“934782392h3id9dun23r92389f8ajh”),
                ‘1’ : ObjectId(“934782392sfisfosii6n23r92389f8ajh”),
                ‘2’ : ObjectId(“79283798423hjskdfsnfwefw89fwfui”)
      }
}
```

4. Aim: - Display document

Procedure:

Syntax:

```
        db.<collection name>.find()
```

Example:

```
        db.Students.find()
```

Output:

```
        my_db> [
{
        _id : ObjectId(“934782392h3id9dun23r92389f8ajh”),
        “Reg No”:’01’,
        “Name”:’Rahul’,
        ”Age”:’34’
},
{
        _id : ObjectId(“934782392sfisfosii6n23r92389f8ajh”),
        “Reg No”:’02’,
        “Name”:’Rohit’,
        ”Age”:’19’
},
{
        _id : ObjectId(“79283798423hjskdfsnfwefw89fwfui”),
        “Reg No”:’03’,
        “Name”:’Amit’,
        ”Age”:’24’
}
]
```

5. Aim: - Deletion of document from collection

Procedure:

Syntax:

```
        db.<collection name>.remove(deletion criteria)
```

Example:

```
        db.Students.remove({“Reg No”:”03”})
        db.Students.find()
```

Output:

```
        my_db> [
{
        _id : ObjectId(“934782392h3id9dun23r92389f8ajh”),
        “Reg No”:’01’,
        “Name”:’Rahul’,
        ”Age”:’34’
},
{
        _id : ObjectId(“934782392sfisfosii6n23r92389f8ajh”),
        “Reg No”:’02’,
        “Name”:’Rohit’,
        ”Age”:’19’
}
]
```

6. Aim: - Updating of document

Procedure:

Syntax:

```
        db.<collection name>.update(criteria, updated_data)
```

Example:

```
        db.Students.update({“Reg No” : ”02”}, {$set : {“age” : “21”}})
        db.Students.find()
```

Output:

```
        my_db> [
{
        _id : ObjectId(“934782392h3id9dun23r92389f8ajh”),
        “Reg No”:’01’,
        “Name”:’Rahul’,
        ”Age”:’34’
},
{
        _id : ObjectId(“934782392sfisfosii6n23r92389f8ajh”),
        “Reg No”:’02’,
        “Name”:’Rohit’,
        ”Age”:’21’
}
]
```

Result: - MongoDB commands executed successfully.

EXPERIMENT – 4

WRITE A PROGRAM IN PYTHON TO PERFORM BASIC OPERATION ON IMAGES

1. **Display image**
2. **Display resized image**
3. **Display grayscale image**
4. **Display flipped image**

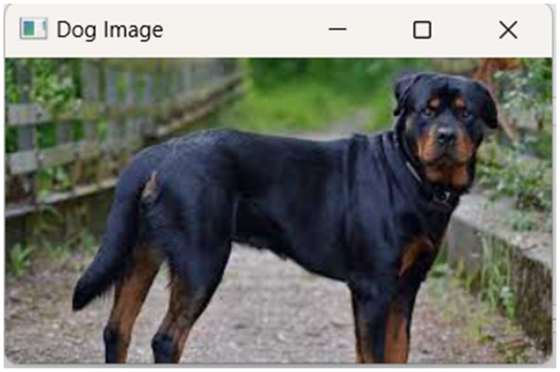
1. **Aim:- Display Image**

Procedure:

```
import cv2

img = cv2.imread('dog.jpeg')

cv2.imshow("Dog Image",img)

cv2.waitKey()

cv2.destroyAllWindows()
```

Input:

Output:



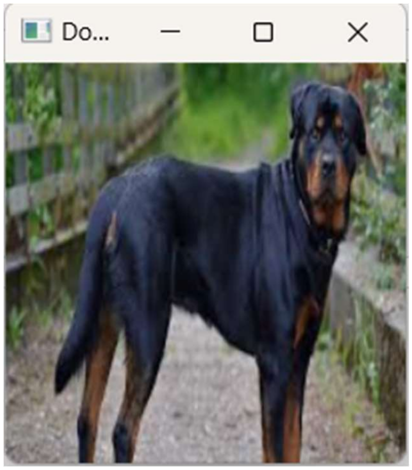
2. <u>Aim</u>:- Display resized image

<u>Procedure</u>:

```
import cv2
img = cv2.imread('dog.jpeg')
resize_img = cv2.resize(img,(200,200))
cv2.imshow("Dog Resized Image",resize_img)
cv2.waitKey()
cv2.destroyAllWindows()
```

<u>Input</u>:

<u>Output</u>:



3. <u>Aim</u>:- Display grayscale image

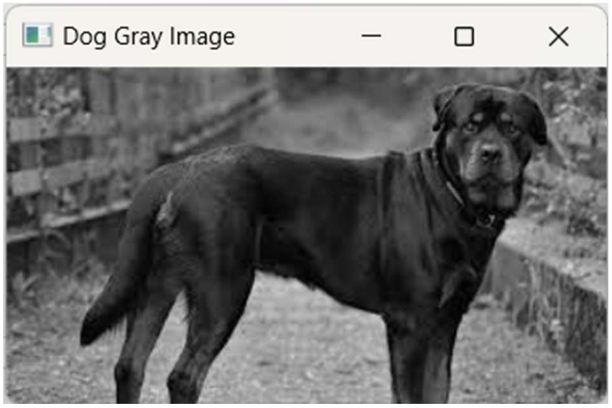
<u>Procedure</u>:

```
import cv2
img = cv2.imread('dog.jpeg')
```

```
gray_img = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)

cv2.imshow("Dog Gray Image",gray_img)

cv2.waitKey()

cv2.destroyAllWindows()
```

Input:

Output:



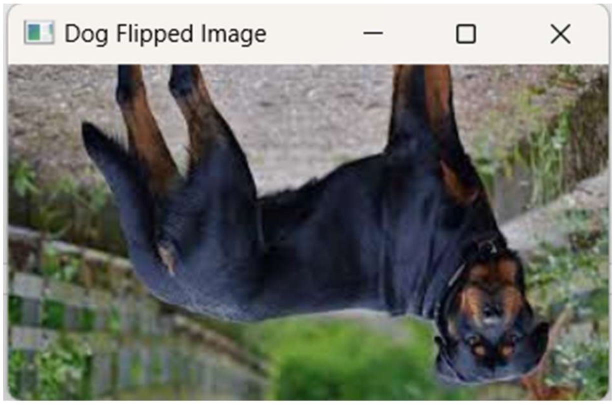
4. Aim:- Display flip image

Procedure:

```
import cv2

img = cv2.imread('dog.jpeg')

flip_img = cv2.flip(img,0)

cv2.imshow("Flipped Image",flip_img)

cv2.waitKey()

cv2.destroyAllWindows()
```

Input:

Output:



Result: Successfully performed basic operations on image.

EXPERIMENT – 5

WRITE A PROGRAM TO COUNT THE FREQUENCY OF WORDS IN A GIVEN PARAGRAPH OR SENTENCE

Aim: - The aim of this program is to count the frequency of words in a text.

Procedure:

```
def word_freq(sent):
    counts = dict()
    words = sent.split()
    for word in words:
        if word in counts:
            counts[word] += 1
        else:
            counts[word] = 1
    return counts
text = str(input('Enter the sentence:'))
print(word_freq(text.lower()))
```

Input:

```
Enter the sentence:This is a python program. this IS A Python code.
```

Output:

```
{'this': 2, 'is': 2, 'a': 2, 'python': 2, 'program.': 1, 'code.': 1}
```

Result: - We successfully executed the frequency counts program.