**Exp.No: 2**

**Run a basic Word Count Map Reduce program to understand Map Reduce**

**Paradigm**

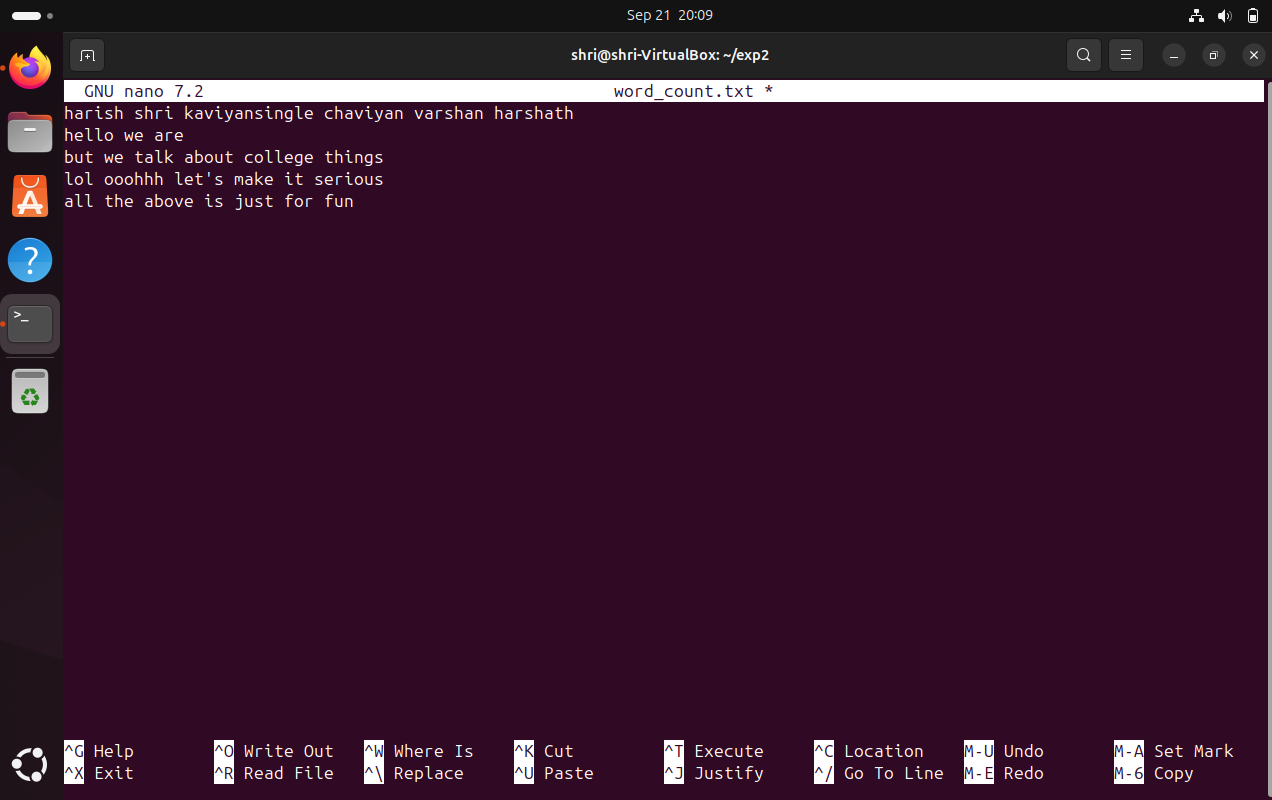
**AIM:**

To run a basic Word Count MapReduce program.

**PROCEDURE:**

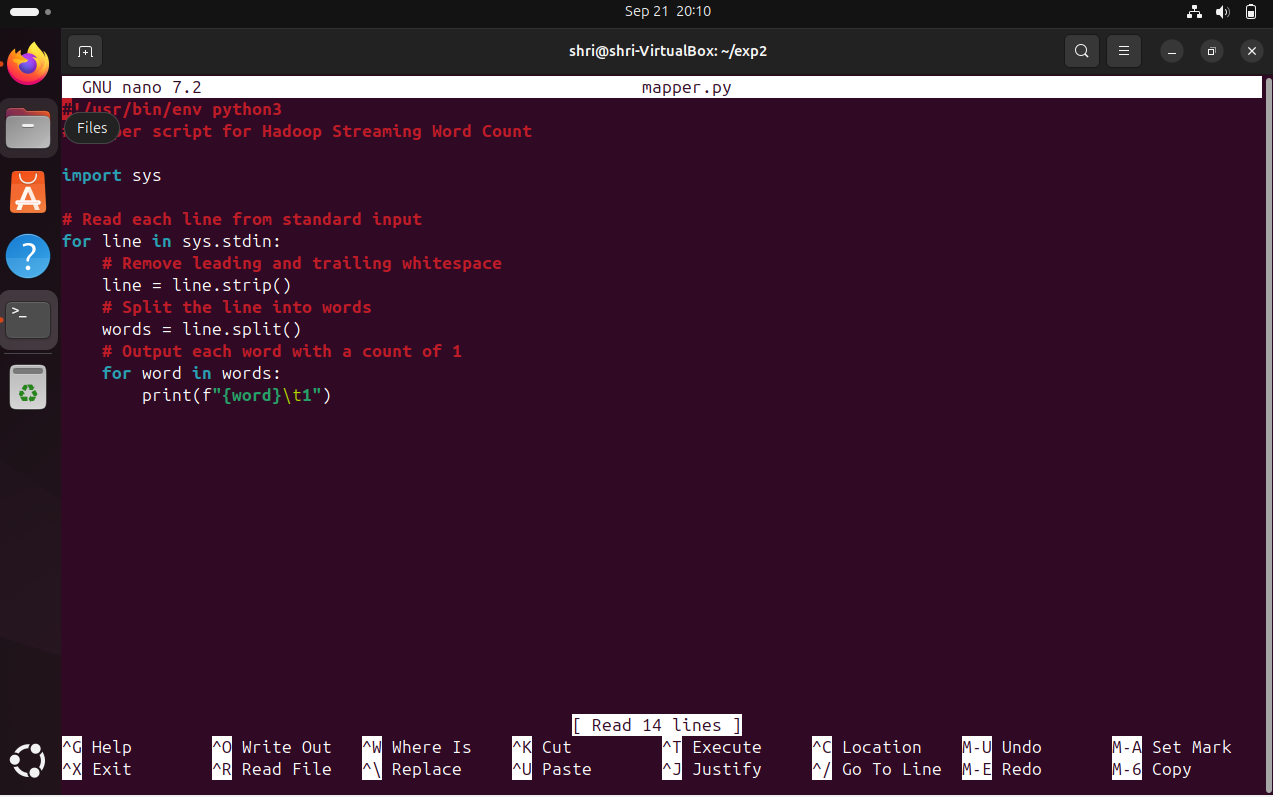
**Step 1: Create Data File:** Create a file named "word\_count\_data.txt" and populate it with text data that you wish to analyse. Login with your hadoop user.

**nano word\_count.txt**

****

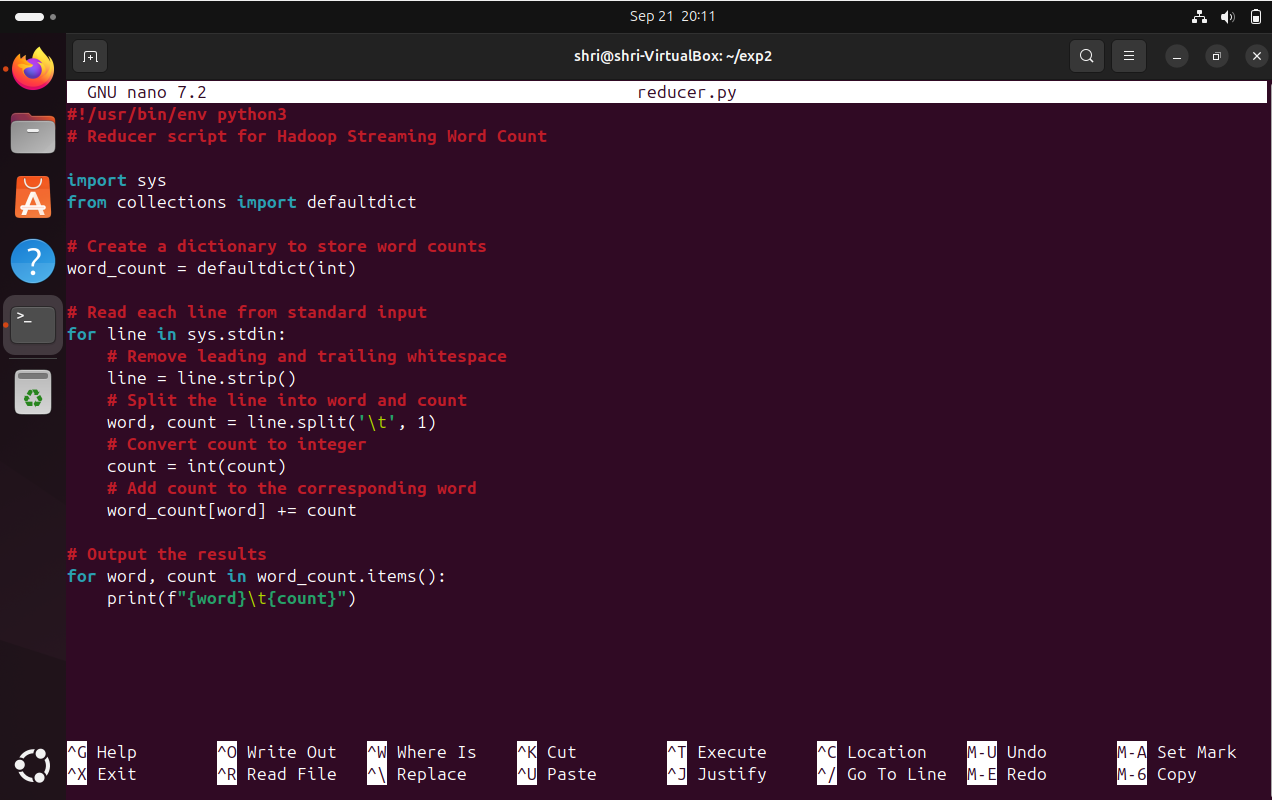
**Step 2: Mapper Logic - mapper.py:** Create a file named "mapper.py" to implement the logic for the mapper. The mapper will read input data from STDIN, split lines into words, and output each word with its count.

**nano mapper.py**

****

**Step 3: Reducer Logic - reducer.py:** Create a file named "reducer.py" to implement the logic for the reducer. The reducer will aggregate the occurrences of each word and generate the final output.

**nano reducer.py**

****

**Step 4: Prepare Hadoop Environment:** Start the Hadoop daemons and create a directory in HDFS to store your data.

**start-all.sh**

**hdfs dfs -mkdir /word\_count\_in\_python**

**hdfs dfs -copyFromLocal /path/to/word\_count.txt /word\_count\_in\_python**

**Step 5: Make Python Files Executable:** Give executable permissions to your mapper.py and reducer.py files.

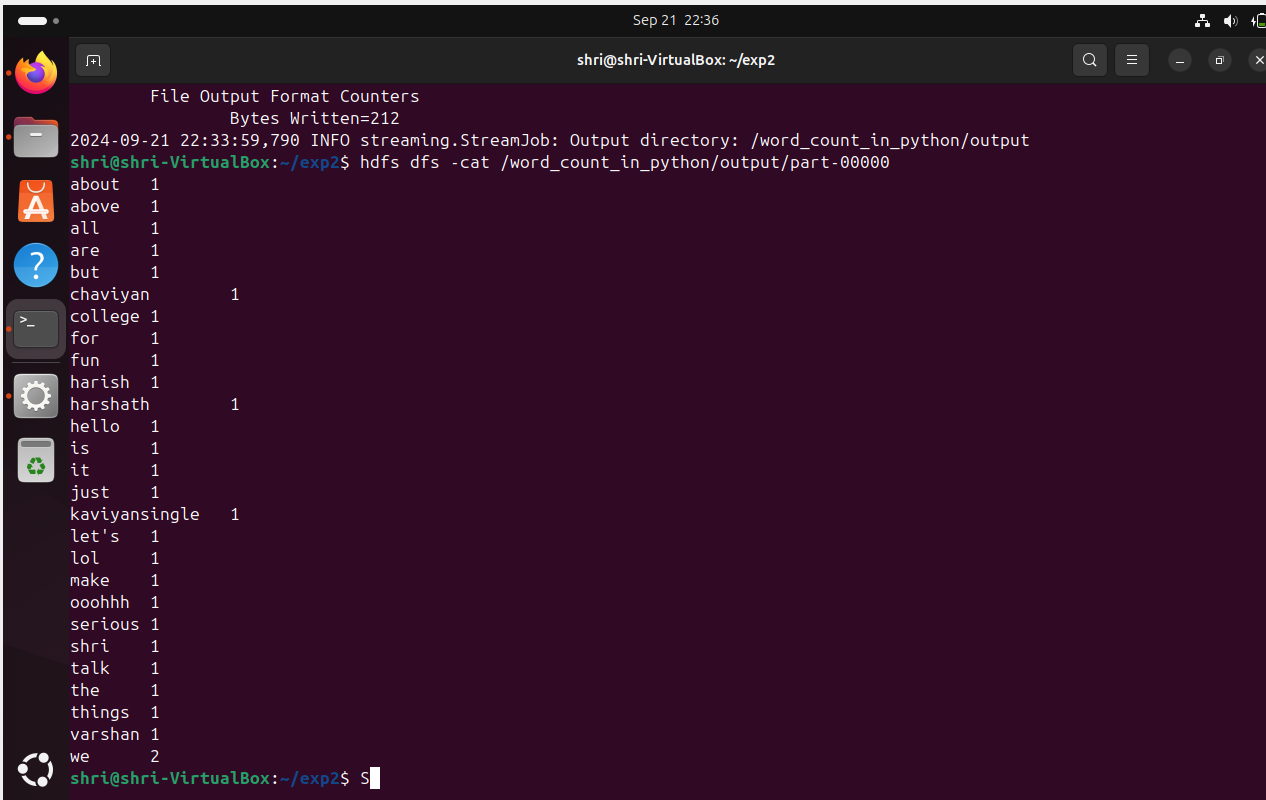
**chmod 777 mapper.py reducer.py**

**Step 6: Run Word Count using Hadoop Streaming:** Download the latest hadoop-streaming jar file and place it in a location you can easily access. Then run the Word Count program using Hadoop Streaming.

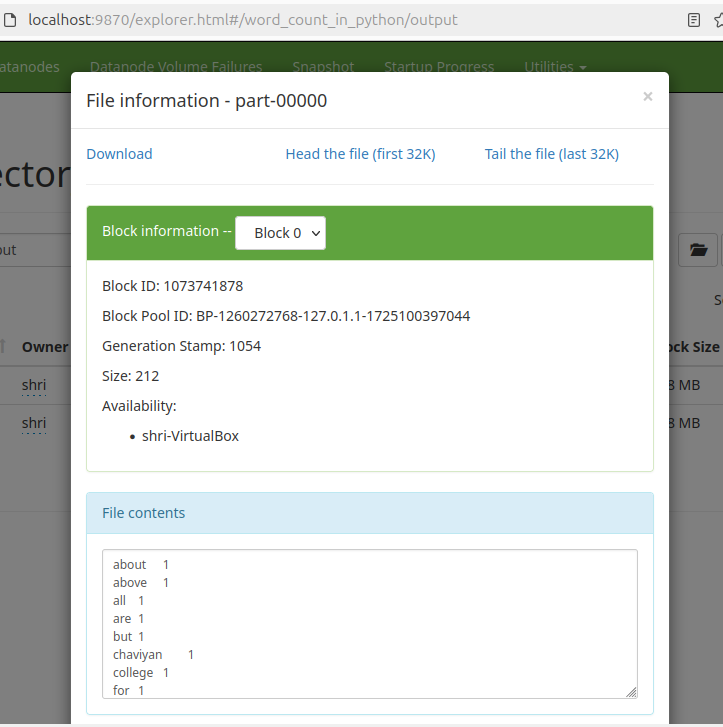
**hadoop jar /path/to/hadoop-streaming-3.3.6.jar \ -input /word\_count\_in\_python/word\_count.txt \ -output /word\_count\_in\_python/output \ -mapper /path/to/mapper.py \ -reducer /path/to/reducer.py**

**Step 7: Check Output:** Check the output of the Word Count program in the specified HDFS output directory.

**hdfs dfs -cat /word\_count\_in\_python/output/part-00000**

****

**Step-8: Check the Output in the browser.**

****

**RESULT:**

Thus, the program for basic Word Count Map Reduce has been executed successfully.