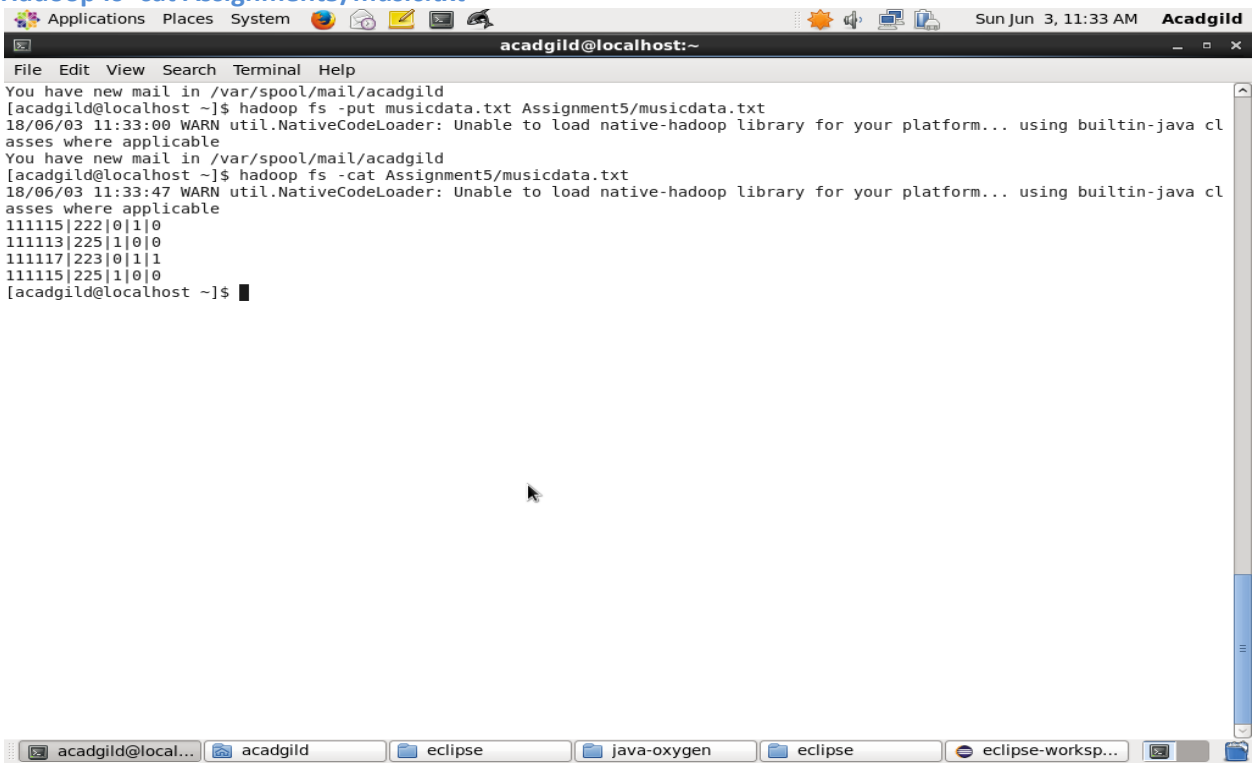
Assignment 5

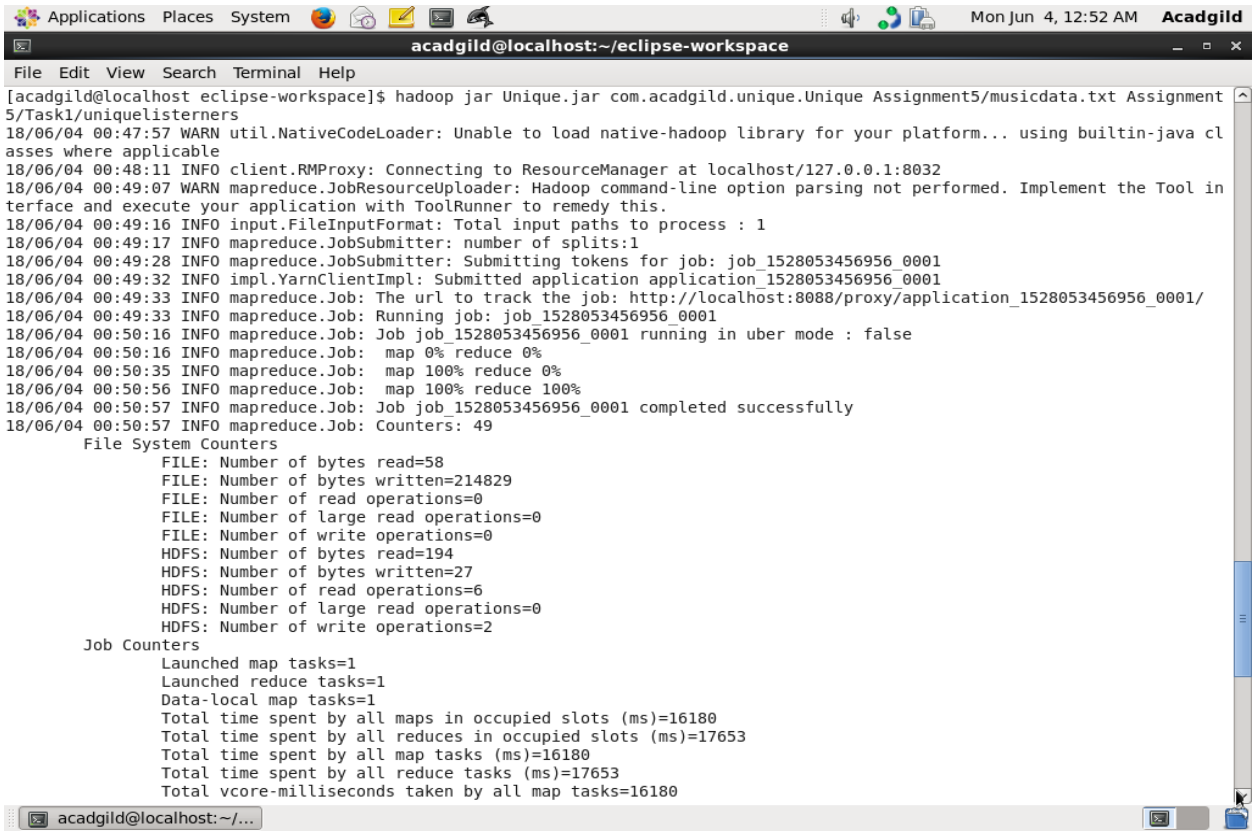
- **Create music.txt in local File system and put the file to Hadoop file system.**
  Command:
  Hadoop fs -put music.txt Assignment5/music.txt.txt
- **Display the contents of music.txt**
  Command:
  Hadoop fs -cat Assignment5/music.txt

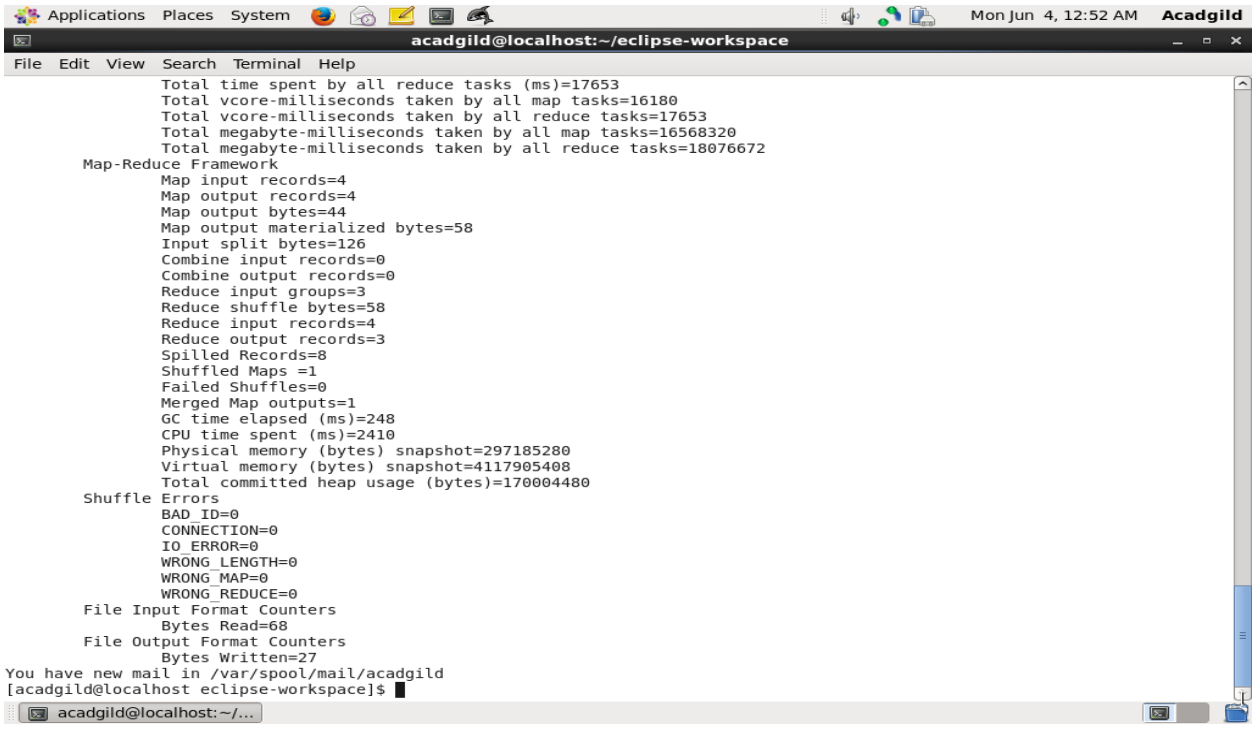


Task 1:
Find the number of unique listeners in the data set.

Step 1:
Create jar file "Unique.jar"

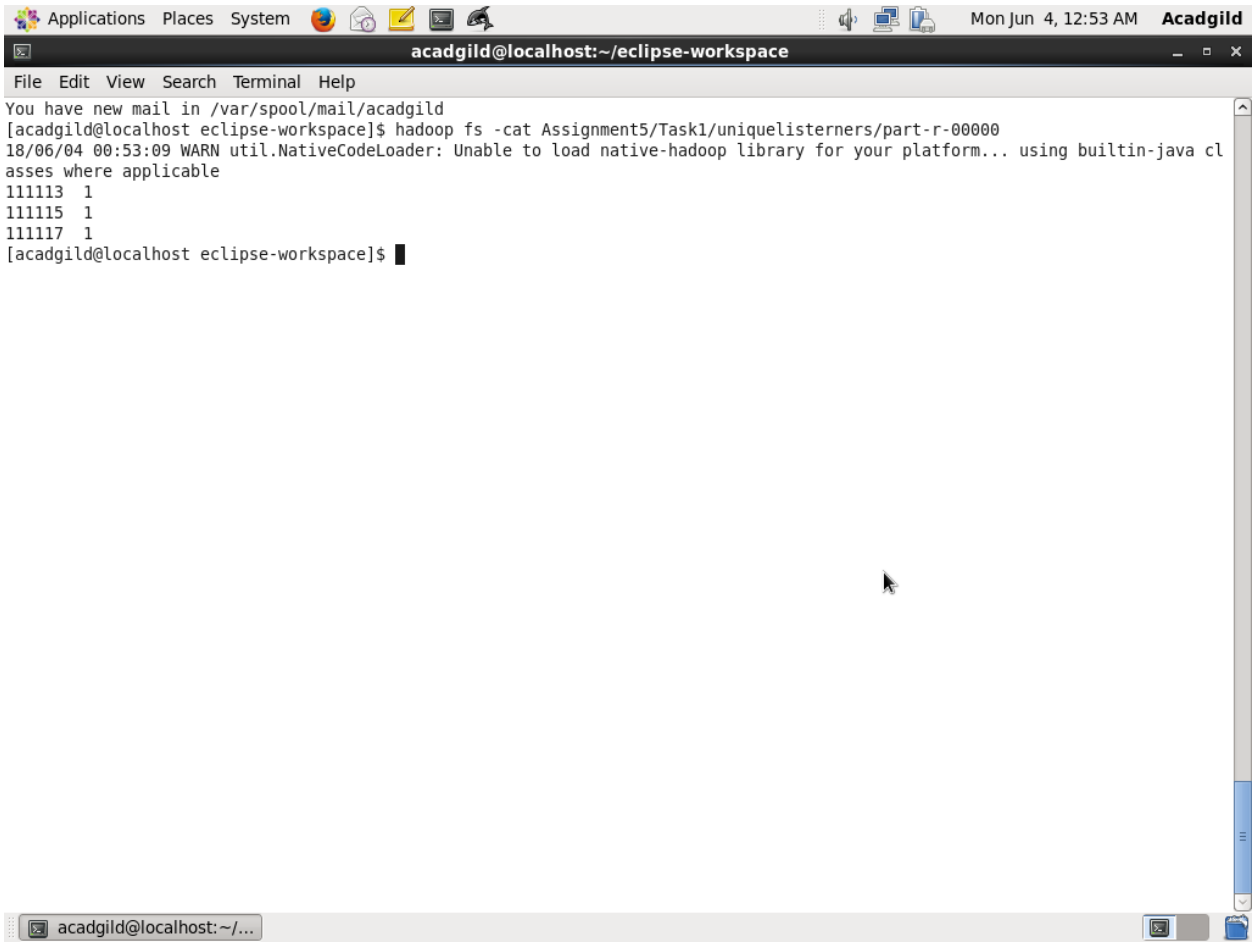
Step 2:
Execute the jar file using the below command
Hadoop jar Unique.jar com.acadgild.unique.Unique Assignment5/music.txt Assignmnet5/Task1/Uniquelisteners


Applications Places System
Mon Jun 4, 12:52 AM Acadgild
acadgild@localhost:~/eclipse-workspace
File Edit View Search Terminal Help
[acadgild@localhost eclipse-workspace]$ hadoop jar Unique.jar com.acadgild.unique.Unique Assignment5/musicdata.txt Assignment
5/Task1/uniquelisterners
18/06/04 00:47:57 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
18/06/04 00:48:11 INFO client.RMProxy: Connecting to ResourceManager at localhost/127.0.0.1:8032
18/06/04 00:49:07 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool in
terface and execute your application with ToolRunner to remedy this.
18/06/04 00:49:16 INFO input.FileInputFormat: Total input paths to process : 1
18/06/04 00:49:17 INFO mapreduce.JobSubmitter: number of splits:1
18/06/04 00:49:28 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1528053456956_0001
18/06/04 00:49:32 INFO impl.YarnClientImpl: Submitted application application_1528053456956_0001
18/06/04 00:49:33 INFO mapreduce.Job: The url to track the job: http://localhost:8088/proxy/application_1528053456956_0001/
18/06/04 00:49:33 INFO mapreduce.Job: Running job: job_1528053456956_0001
18/06/04 00:50:16 INFO mapreduce.Job: Job job_1528053456956_0001 running in uber mode : false
18/06/04 00:50:16 INFO mapreduce.Job: map 0% reduce 0%
18/06/04 00:50:35 INFO mapreduce.Job: map 100% reduce 0%
18/06/04 00:50:56 INFO mapreduce.Job: map 100% reduce 100%
18/06/04 00:50:57 INFO mapreduce.Job: Job job_1528053456956_0001 completed successfully
18/06/04 00:50:57 INFO mapreduce.Job: Counters: 49
File System Counters
FILE: Number of bytes read=58
FILE: Number of bytes written=214829
FILE: Number of read operations=0
FILE: Number of large read operations=0
FILE: Number of write operations=0
HDFS: Number of bytes read=194
HDFS: Number of bytes written=27
HDFS: Number of read operations=6
HDFS: Number of large read operations=0
HDFS: Number of write operations=2
Job Counters
Launched map tasks=1
Launched reduce tasks=1
Data-local map tasks=1
Total time spent by all maps in occupied slots (ms)=16180
Total time spent by all reduces in occupied slots (ms)=17653
Total time spent by all map tasks (ms)=16180
Total time spent by all reduce tasks (ms)=17653
Total vcore-milliseconds taken by all map tasks=16180
acadgild@localhost:~/...


Applications Places System
Mon Jun 4, 12:52 AM Acadgild
acadgild@localhost:~/eclipse-workspace
File Edit View Search Terminal Help
Total time spent by all reduce tasks (ms)=17653
Total vcore-milliseconds taken by all map tasks=16180
Total vcore-milliseconds taken by all reduce tasks=17653
Total megabyte-milliseconds taken by all map tasks=16568320
Total megabyte-milliseconds taken by all reduce tasks=18076672
Map-Reduce Framework
Map input records=4
Map output records=4
Map output bytes=44
Map output materialized bytes=58
Input split bytes=126
Combine input records=0
Combine output records=0
Reduce input groups=3
Reduce shuffle bytes=58
Reduce input records=4
Reduce output records=3
Spilled Records=8
Shuffled Maps =1
Failed Shuffles=0
Merged Map outputs=1
GC time elapsed (ms)=248
CPU time spent (ms)=2410
Physical memory (bytes) snapshot=297185280
Virtual memory (bytes) snapshot=4117905408
Total committed heap usage (bytes)=170004480
Shuffle Errors
BAD_ID=0
CONNECTION=0
IO_ERROR=0
WRONG_LENGTH=0
WRONG_MAP=0
WRONG_REDUCE=0
File Input Format Counters
Bytes Read=68
File Output Format Counters
Bytes Written=27
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost eclipse-workspace]$
acadgild@localhost:~/...

Step 3:

Display the contents of the output file.



Code:

```
/**
 *
 */
package com.acadgild.unique;
import java.io.IOException;
import java.util.Iterator;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
```

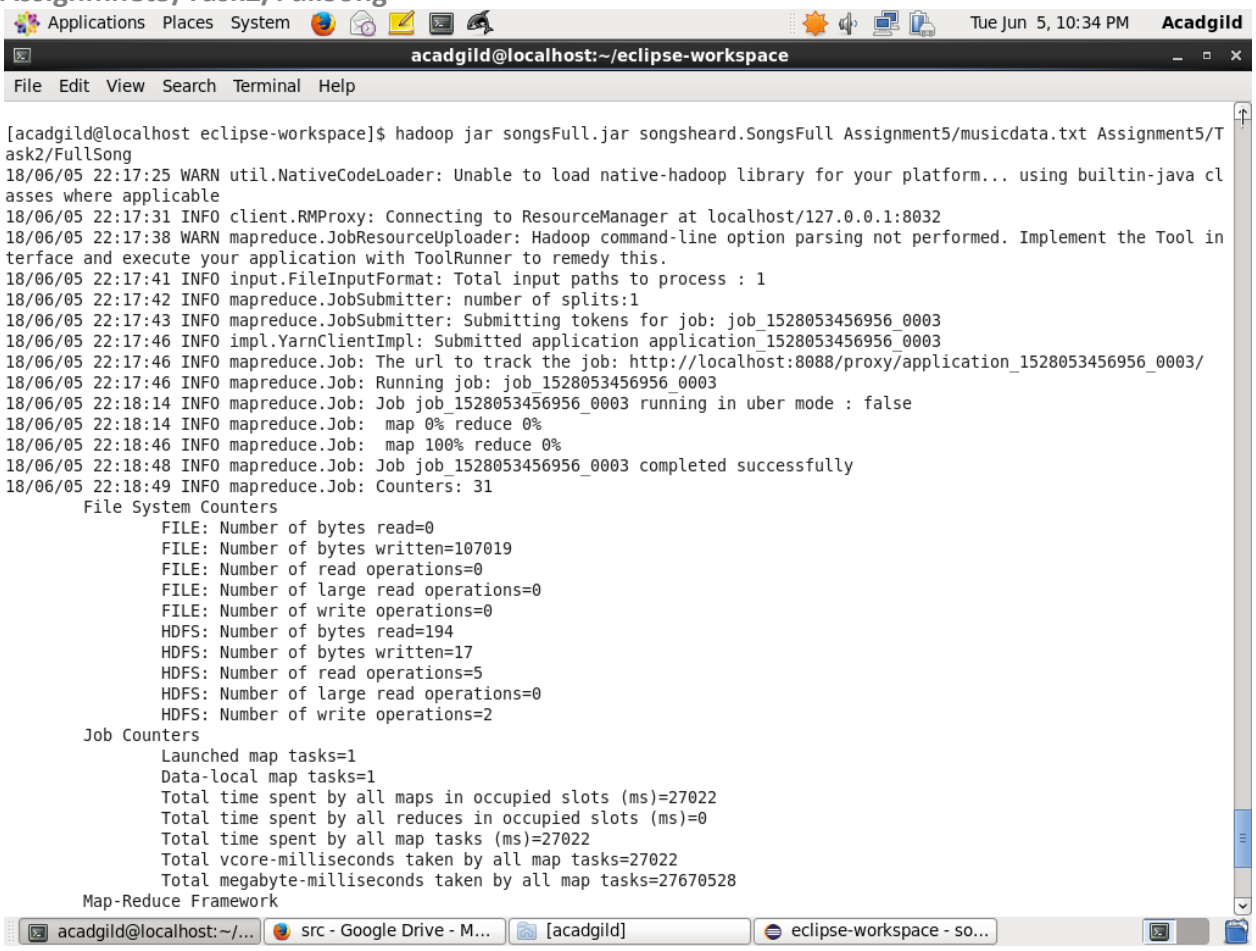
```
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;

/**
 * @author acadgild
 *
 */
public class Unique {

    /**
     *
     */
    //public Unique() {
        // TODO Auto-generated constructor stub
    //}

    public static class Uniquemapper extends
            Mapper<LongWritable, Text, Text, IntWritable>
            {

                private final static IntWritable one = new IntWritable(1);

                public void map(LongWritable key, Text value,
                Context context) throws IOException,InterruptedException
                {
                    String music = value.toString();
                    String[] userid = music.split("\\|");

                    context.write(new Text(userid[0]), one);
                }
            }

            public static class Uniquereducer extends
            Reducer<Text, IntWritable, Text, IntWritable>
            {
            public void reduce(Text key, Iterable<IntWritable> values,
            Context context)
            throws
            IOException,InterruptedException
            {
                int cname = 0;


                cname = cname + 1 ;



                context.write(key,new IntWritable(cname));
            }
```

```
        }

        public static void main(String[] args)
        throws IllegalArgumentException, IOException,
        ClassNotFoundException, InterruptedException
        {
            // TODO Auto-generated method stub
            Configuration conf = new Configuration();
            Job job = new Job(conf, "unique");

            job.setJarByClass(Unique.class);
            job.setMapperClass(Uniquemapper.class);
            job.setOutputKeyClass(Text.class);
            job.setOutputValueClass(IntWritable.class);
            job.setReducerClass(Uniquereducer.class);


            FileInputFormat.addInputPath(job, new Path(args[0]));
            FileOutputFormat.setOutputPath(job, new Path(args[1]));

  boolean result = job.waitForCompletion(true);

            System.exit(result ? 0 : 1);

        }

}
```

Task 2:
What are the number of times a song was heard fully.

Step 1:
Create jar file “songsFull.jar”

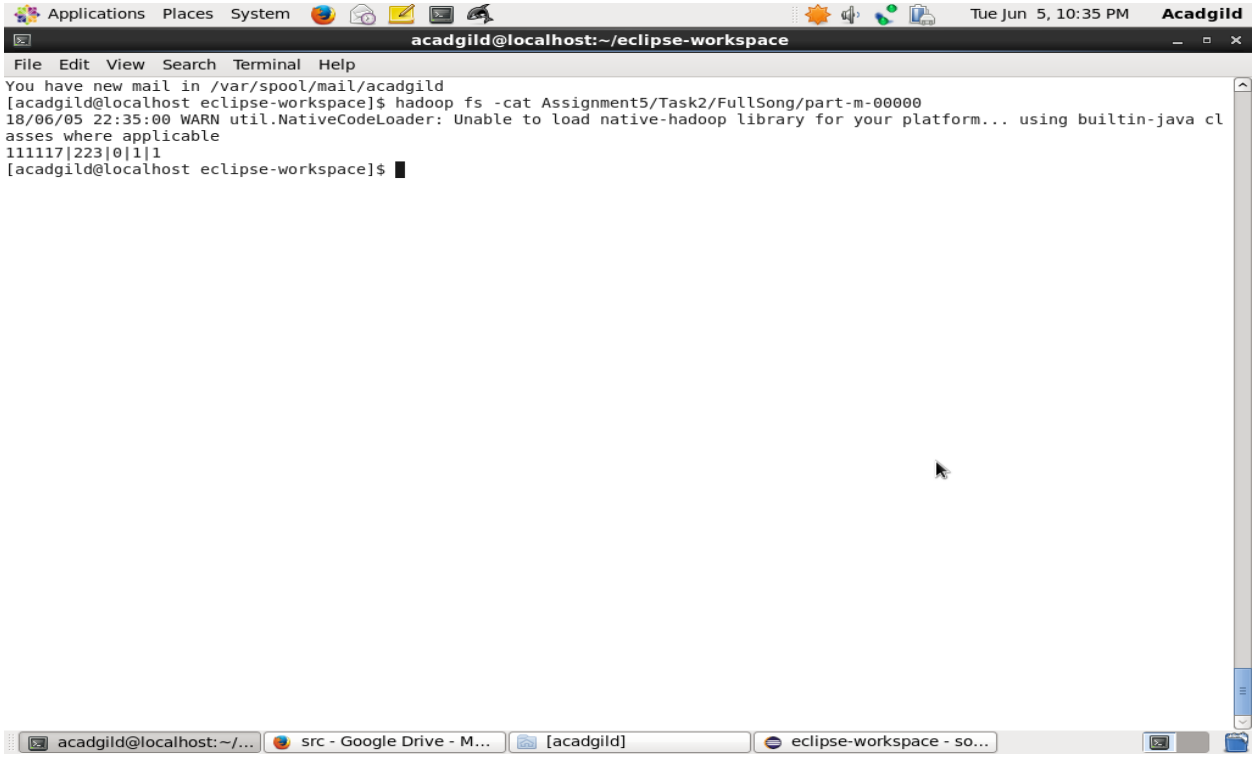
Step 2:
Execute the jar file using the below command
Hadoop jar songsFull.jar songsheard.SongFull Assignment5/music.txt Assignmnet5/Task2/FullSong



```
                FILE: Number of bytes written=107019
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=194
                HDFS: Number of bytes written=17
                HDFS: Number of read operations=5
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=1
                Data-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=27022
                Total time spent by all reduces in occupied slots (ms)=0
                Total time spent by all map tasks (ms)=27022
                Total vcore-milliseconds taken by all map tasks=27022
                Total megabyte-milliseconds taken by all map tasks=27670528
        Map-Reduce Framework
                Map input records=4
                Map output records=1
                Input split bytes=126
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=192
                CPU time spent (ms)=2820
                Physical memory (bytes) snapshot=91889664
                Virtual memory (bytes) snapshot=2056761344
                Total committed heap usage (bytes)=32571392
        File Input Format Counters
                Bytes Read=68
        File Output Format Counters
                Bytes Written=17
        songsheard.SongsFull$songcounter
                FULL=1
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost eclipse-workspace]$
```

Step 3:

Display the contents of the output file.



Code:

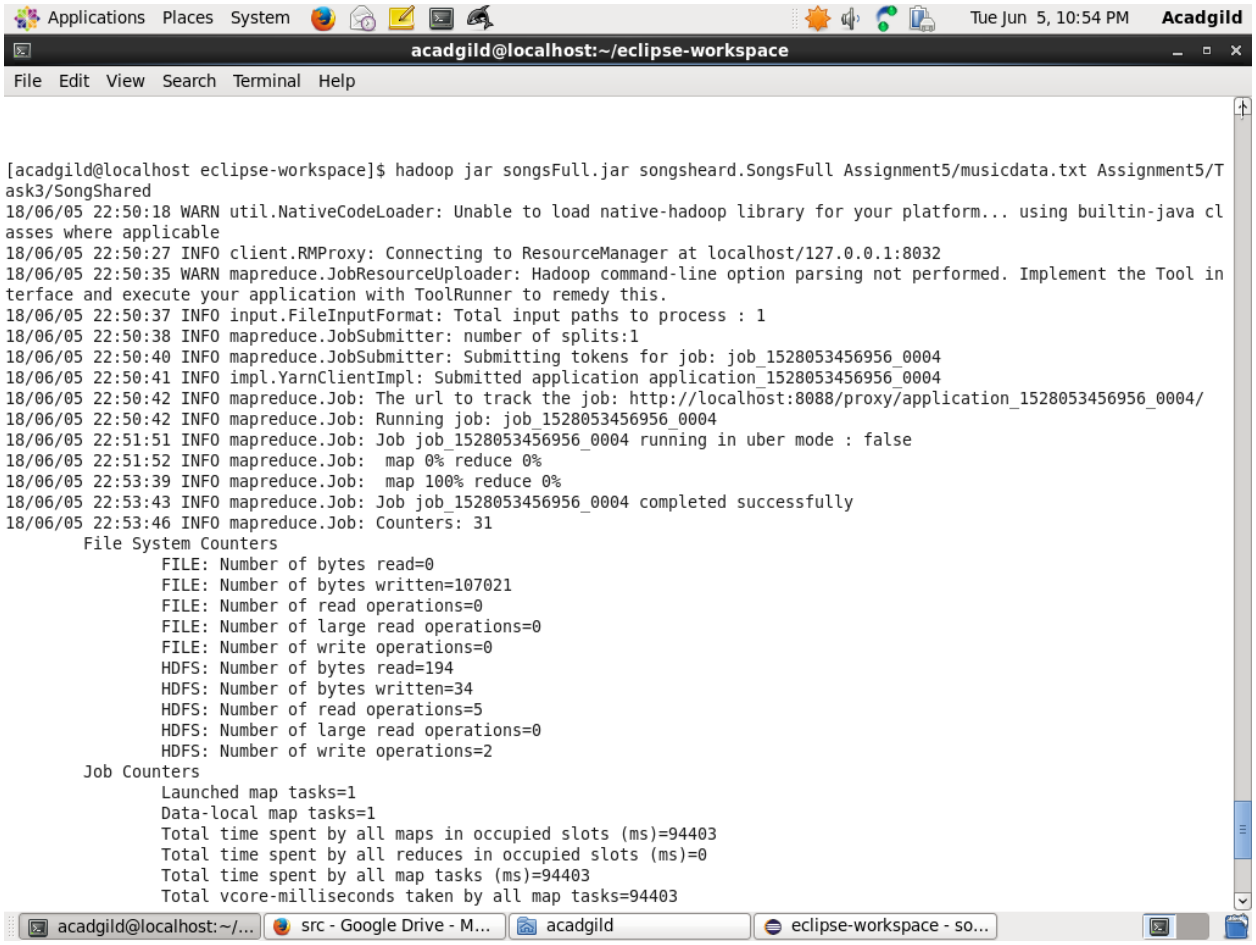
```
package songsheard;
import org.apache.hadoop.fs.Path;

import java.io.IOException;

import org.apache.hadoop.conf.*;
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

import org.apache.hadoop.mapreduce.Counter;
import org.apache.hadoop.mapreduce.Counters;

public class SongsFull {


   public static enum songcounter{
      FULL,

   };


   public static class songMapper extends Mapper<LongWritable, Text, Text, IntWritable>
   {


            public void map(LongWritable key, Text value, Context context) throws IOException,
InterruptedException
        {
               String valueString = value.toString();
               String[] finstr= valueString.split("\\|");
               String songhe = finstr[2].trim();


            if(songhe.equals("1")) {
               context.write(new Text(value),null);
               context.getCounter(songcounter.FULL).increment(1);
            }

         }

   }
```

```
    public static void main(String[] args) throws Exception {


            //Job Related Configurations
            Configuration conf = new Configuration();
            Job job = new Job(conf, "Songs");
            job.setJarByClass(SongsFull.class);


            job.setMapperClass(songMapper.class);
            job.setOutputKeyClass(Text.class);
            job.setOutputValueClass(Text.class);


            // Sets reducer tasks to 0
            job.setNumReduceTasks(0);

            FileInputFormat.addInputPath(job, new Path(args[0]));
            FileOutputFormat.setOutputPath(job, new Path(args[1]));

            boolean result = job.waitForCompletion(true);

            System.exit(result ? 0 : 1);

            Counters counters = job.getCounters();

            Counter sCounter = counters.findCounter(songcounter.FULL);
            System.out.println(sCounter.getDisplayName()+ " : " + sCounter.getValue());



    }


}
```

Task 3:
What are the number of times a song was shared.

Step 1:
Create jar file “songsFull.jar”

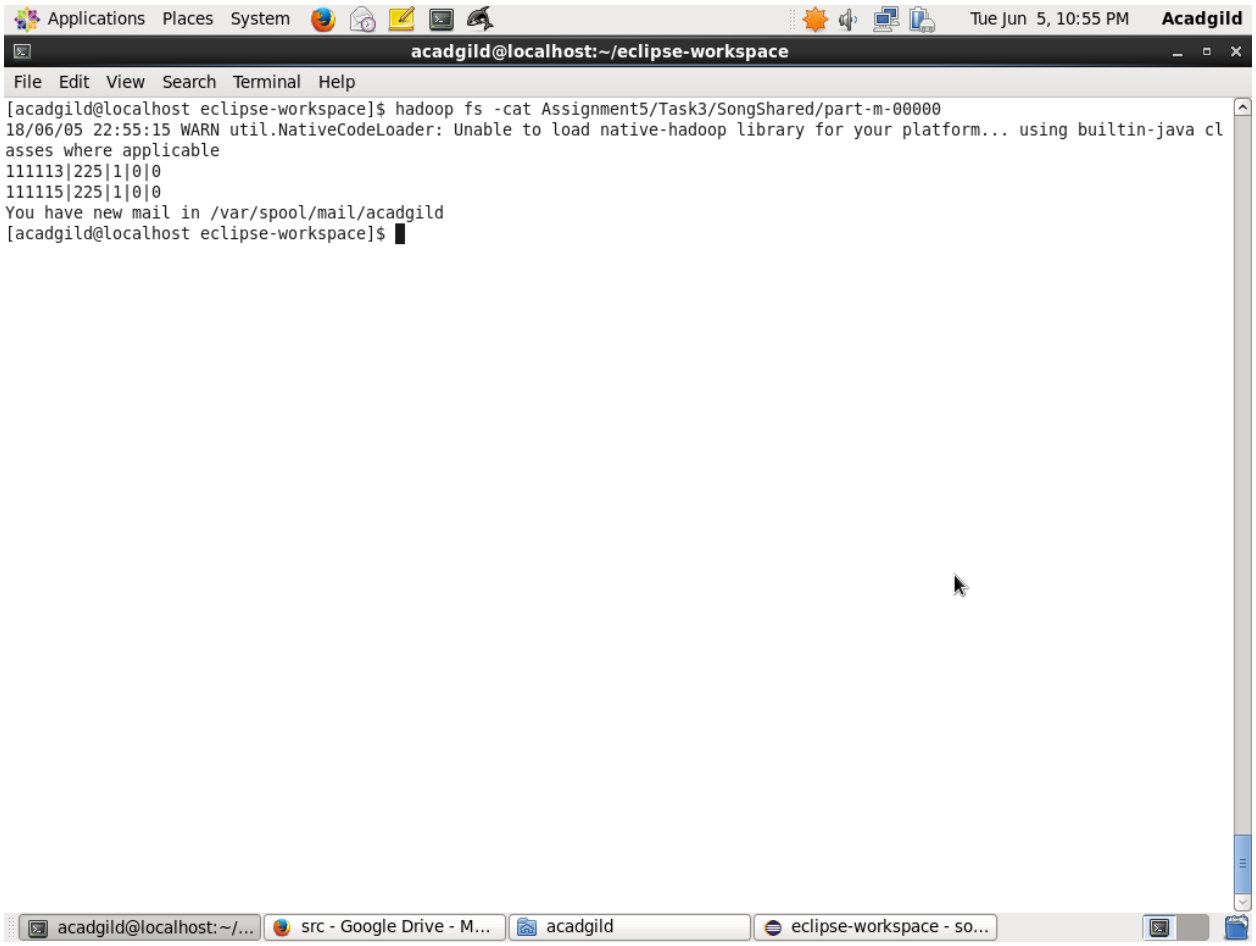
Step 2:
Execute the jar file using the below command
Hadoop jar songsFull.jar songsheard.SongFull Assignment5/music.txt Assignmnet5/Task2/SongShared



```
                FILE: Number of bytes written=107021
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=194
                HDFS: Number of bytes written=34
                HDFS: Number of read operations=5
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=1
                Data-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=94403
                Total time spent by all reduces in occupied slots (ms)=0
                Total time spent by all map tasks (ms)=94403
                Total vcore-milliseconds taken by all map tasks=94403
                Total megabyte-milliseconds taken by all map tasks=96668672
        Map-Reduce Framework
                Map input records=4
                Map output records=2
                Input split bytes=126
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=486
                CPU time spent (ms)=2690
                Physical memory (bytes) snapshot=83181568
                Virtual memory (bytes) snapshot=2056761344
                Total committed heap usage (bytes)=32571392
        File Input Format Counters
                Bytes Read=68
        File Output Format Counters
                Bytes Written=34
        songsheard.SongsFull$songcounter
                SHARE=2
have new mail in /var/spool/mail/acadgild
adgild@localhost eclipse-workspace]$
```

Step 3:

Display the contents of the output file.



Code:

```
package songsheard;
import org.apache.hadoop.fs.Path;

import java.io.IOException;

import org.apache.hadoop.conf.*;
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
```

```
import org.apache.hadoop.mapreduce.Counter;
import org.apache.hadoop.mapreduce.Counters;

public class SongsFull {


    public static enum songcounter{
        SHARE,

    };


    public static class songMapper extends Mapper<LongWritable, Text, Text, IntWritable>
    {


            public void map(LongWritable key, Text value, Context context) throws IOException,
InterruptedException
        {
                String valueString = value.toString();
                String[] finstr= valueString.split("\\|");
                String songhe = finstr[2].trim();


            if(songhe.equals("1")) {
                context.write(new Text(value),null);
                context.getCounter(songcounter.SHARE).increment(1);
            }

        }

    }




    public static void main(String[] args) throws Exception {


            //Job Related Configurations
            Configuration conf = new Configuration();
            Job job = new Job(conf, "Songs");
            job.setJarByClass(SongsFull.class);
```

```
        job.setMapperClass(songMapper.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(Text.class);


        // Sets reducer tasks to 0
        job.setNumReduceTasks(0);

        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));

        boolean result = job.waitForCompletion(true);

        System.exit(result ? 0 : 1);

        Counters counters = job.getCounters();

        Counter sCounter = counters.findCounter(songcounter.SHARE);
        System.out.println(sCounter.getDisplayName()+ " : " + sCounter.getValue());



    }


}
```