**AIM:** To write a map reduce program to implement PARTITIONER in word count program.

**PROGRAM:**

**/\*MAPPER\*/**

import java.util.\*;

import java.io.\*;

import org.apache.hadoop.io.\*;

import org.apache.hadoop.mapreduce.\*;

public class WordMapper extends Mapper<LongWritable, Text, Text, LongWritable>

{

LongWritable one = new LongWritable(1);

int i;

public void map(LongWritable key, Text value, Context contex) throws IOException, InterruptedException

{

String line = value.toString();

String[] wordsinline = line.split(" ");

for( i=0 ; i<wordsinline.length ; i++)

contex.write(new Text(wordsinline[i]),one);

}

}

**/\*REDUCER\*/**

import java.util.\*;

import java.io.\*;

import org.apache.hadoop.io.\*;

import org.apache.hadoop.mapreduce.\*;

public class WordReducer extends Reducer<Text, LongWritable, Text, LongWritable>

{

LongWritable totalWC = new LongWritable();

public void reduce(Text key, Iterable<LongWritable> values, Context context) throws IOException, InterruptedException

{

long wordcount = 0;

for(LongWritable val:values)

wordcount = wordcount + val.get();

totalWC.set(wordcount);

context.write(key,totalWC);

}

}

**/\*PARTITIONER\*/**

import org.apache.hadoop.io.Text;

import org.apache.hadoop.io.LongWritable;

import org.apache.hadoop.mapreduce.Partitioner;

public class WordPartitioner extends Partitioner<Text,LongWritable>

{

public int getPartition(Text key,LongWritable value,int numReduceTasks)

{

if(numReduceTasks==0)

{

return 0;

}

else

{

return(key.toString().charAt(0)%numReduceTasks);

}

}

}

**/\*DRIVER\*/**

import java.util.\*;

import java.io.\*;

import org.apache.hadoop.io.\*;

import org.apache.hadoop.mapreduce.\*;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;

import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;

import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;

public class WordCount

{

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

{

Configuration conf = new Configuration();

Job job = Job.getInstance(conf,"xyz");

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(LongWritable.class);

job.setMapperClass(WordMapper.class);

job.setReducerClass(WordReducer.class);

job.setPartitionerClass(WordPartitioner.class);

job.setNumReduceTasks(2);

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

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

job.setJarByClass(WordCount.class);

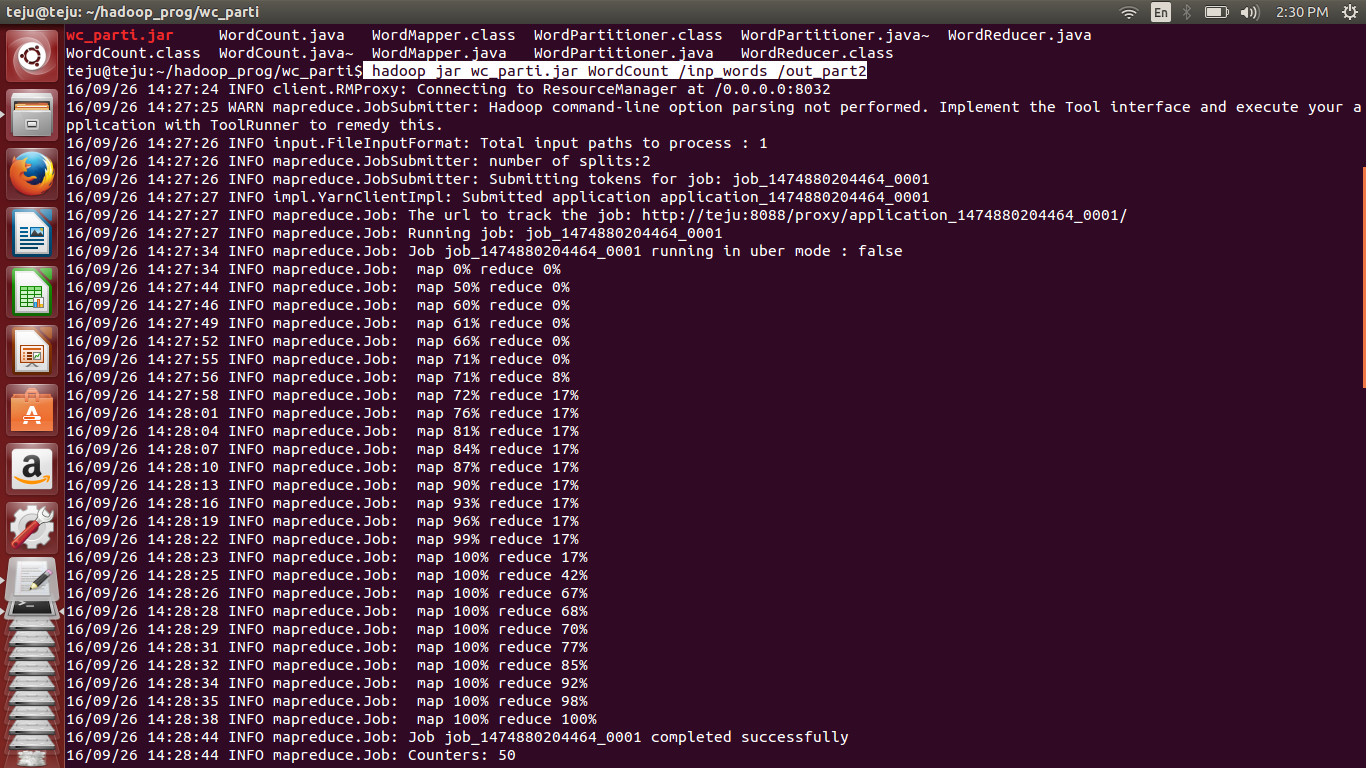
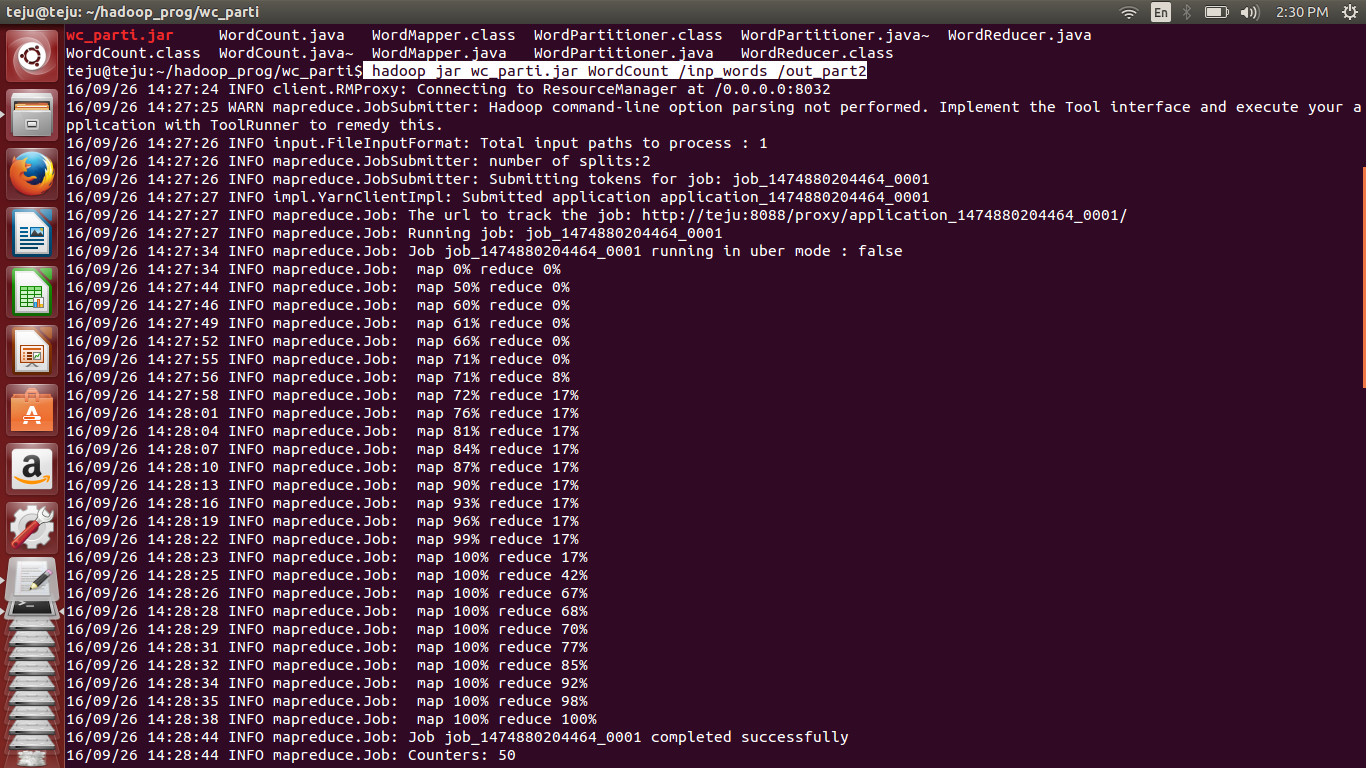
job.waitForCompletion(true);

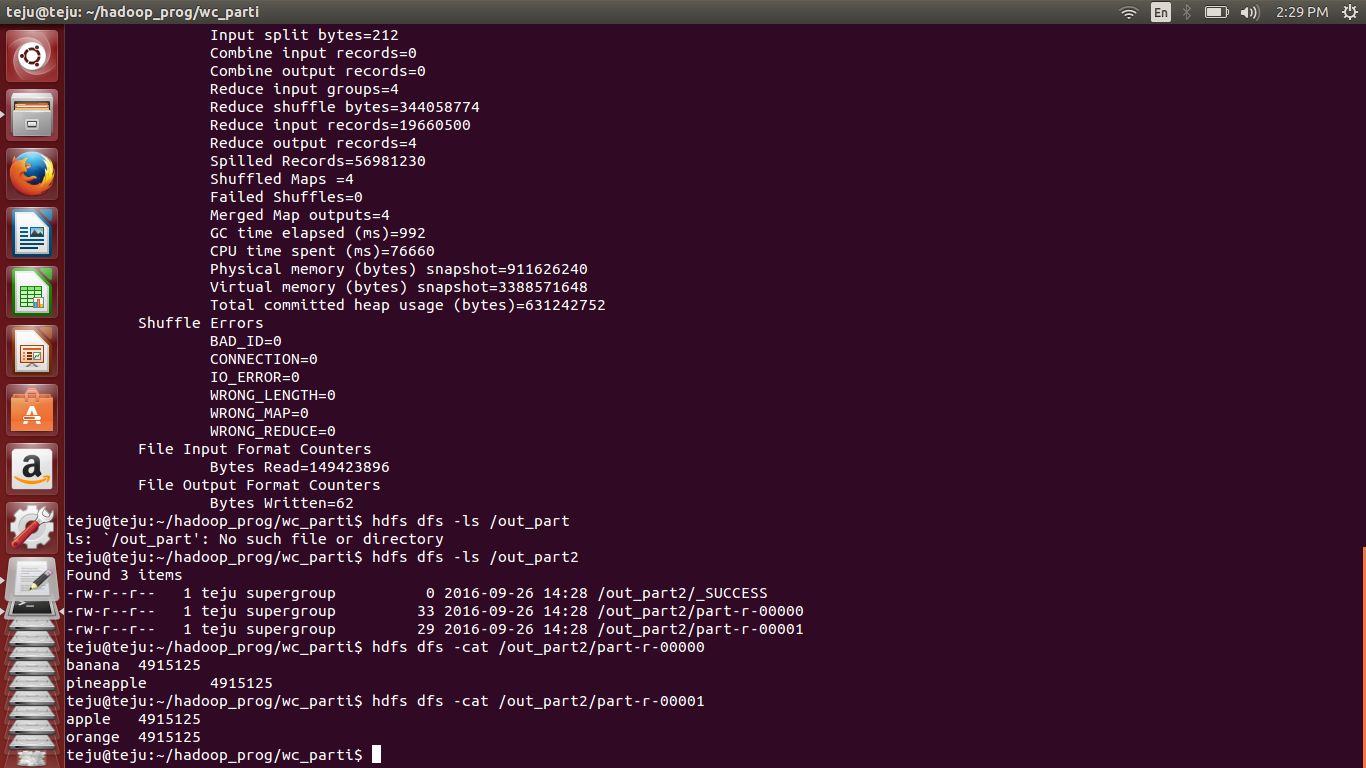
}

}

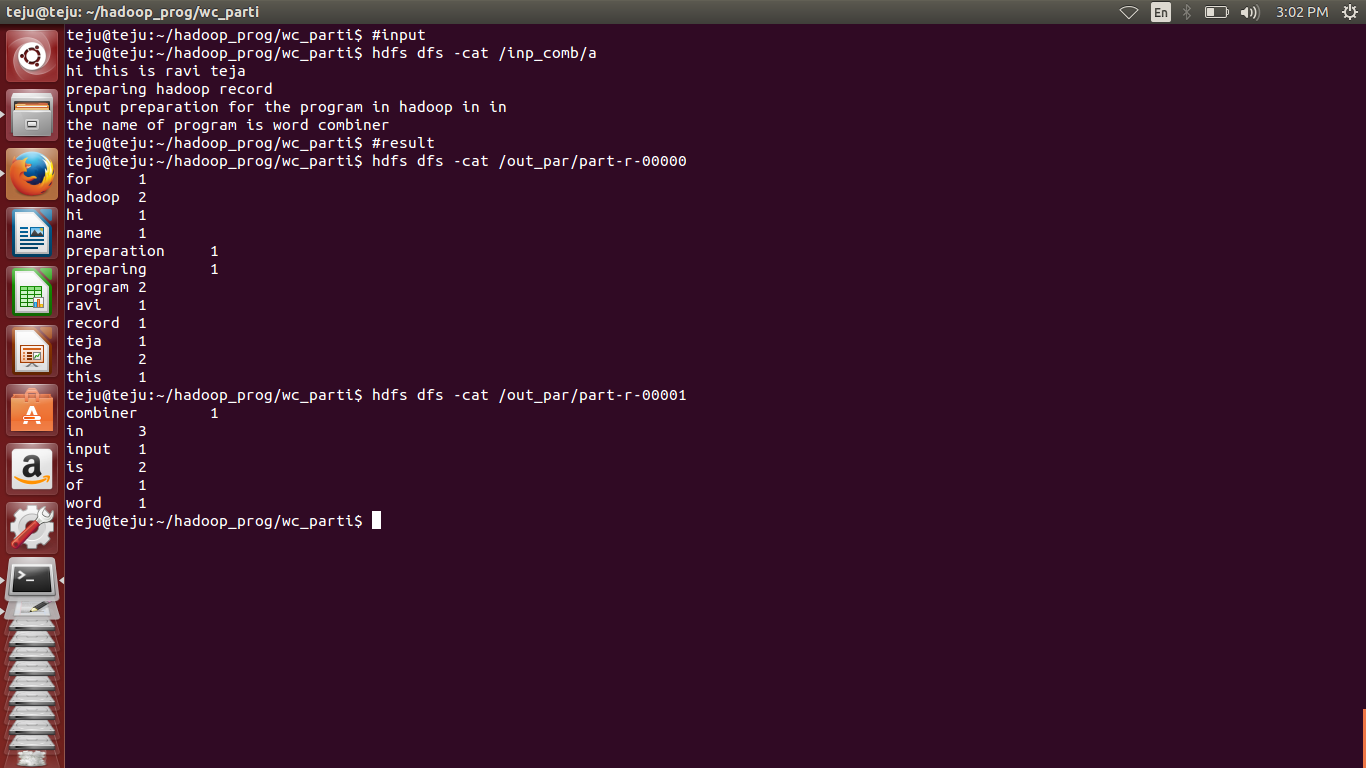
**OUTPUT 1:**

**Input.txt generated by c program is given as input.**

****

****

**OUTPUT 2:**

****