**AIM:** To write a map reduce program to implement search a word in a file.

**PROGRAM:**

**/\*MAPPER\*/**

import java.util.\*;

import java.io.\*;

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

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

import org.apache.hadoop.conf.Configuration;

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

{

int count;

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

{

count=0;

Configuration conf = contex.getConfiguration();

String search\_key=conf.get("key");

String line1=value.toString();

String line=value.toString();

line=line.replaceAll("[+-;\",]"," ");

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

for(int i=0;i<word.length;i++)

{

if((search\_key.toUpperCase()).equals(word[i].toUpperCase()))

{

count++;

}

}

contex.write(new Text(search\_key),new Text(line1+"~"+count));

}

}

**/\*REDUCER\*/**

import java.util.\*;

import java.io.\*;

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

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

import org.apache.hadoop.conf.Configuration;

public class SearchReducer extends Reducer<Text, Text, Text, Text>

{

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

{

int times=0,t;

for(Text value:values)

{

String p=value.toString();

String[] l=p.split("~");

t=Integer.parseInt(l[1]);

if(t!=0)

{

//x=x+l[0]+"\n";

context.write(new Text("key:"+key+"\n"+"found in line:"),new Text(l[0]));

times=1;

}

}

if(times==0)

context.write(new Text("key:"+key+"\n"+"not found"),new Text(""));

}

}

**/\*DRIVER\*/**

import java.util.\*;

import java.io.\*;

import org.apache.hadoop.util.Tool;

import org.apache.hadoop.util.ToolRunner;

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

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

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.conf.Configured;

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 SearchDriver extends Configured implements Tool

{

public int run(String args[]) throws Exception

{

Configuration conf = this.getConf();

conf.set("key",args[2]);

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

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(Text.class);

job.setMapOutputKeyClass(Text.class);

job.setMapOutputValueClass(Text.class);

job.setMapperClass(SearchMapper.class);

job.setReducerClass(SearchReducer.class);

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

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

job.setJarByClass(SearchDriver.class);

return job.waitForCompletion(true) ? 0 : 1;

}

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

{

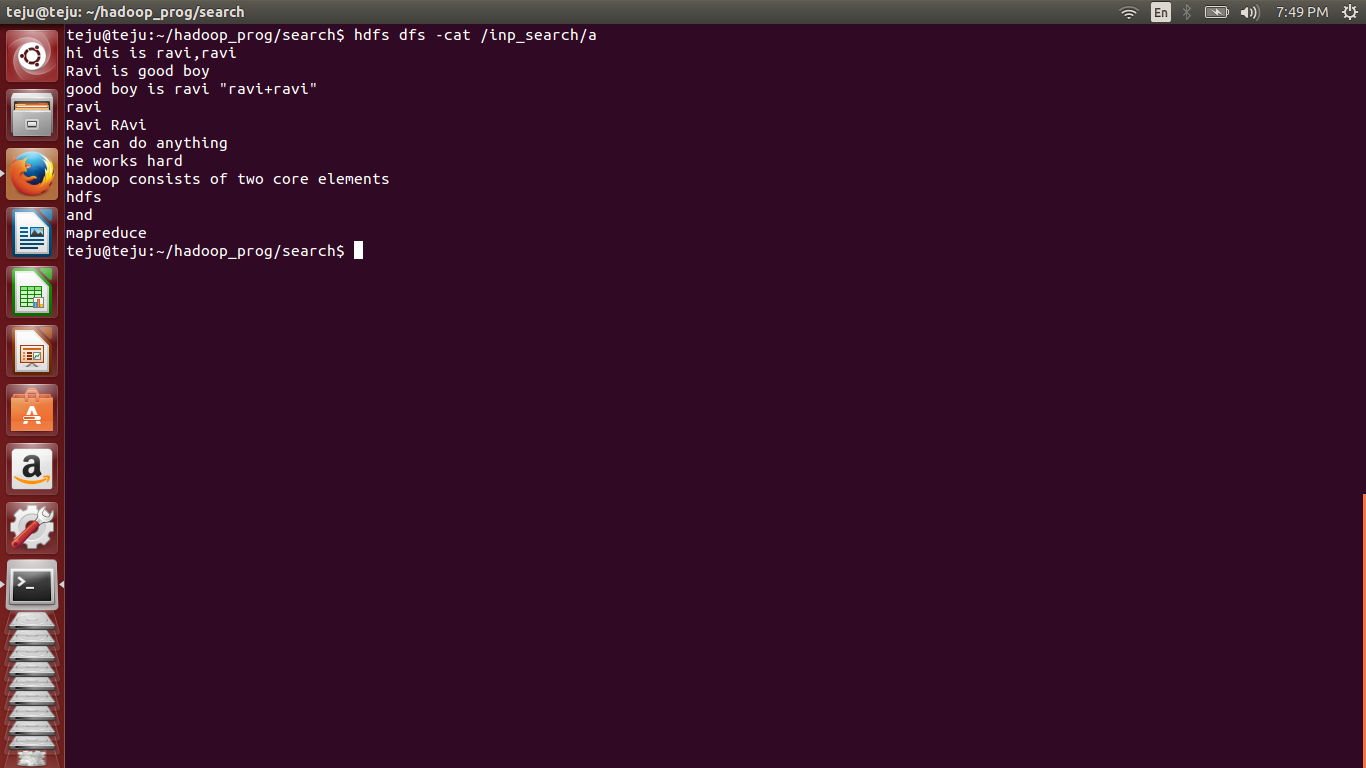
int res= ToolRunner.run(new Configuration(), new SearchDriver(), args);

System.exit(res);

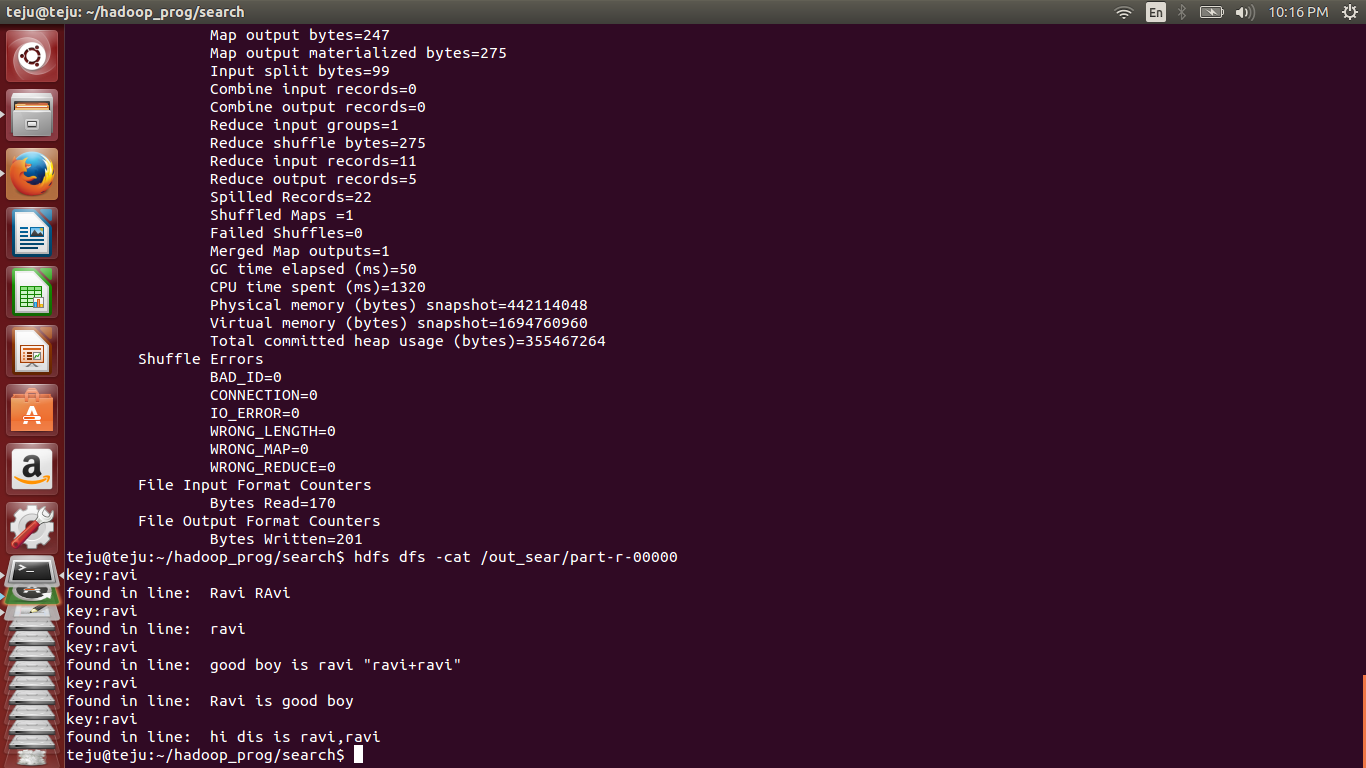
}

}

**INPUT:**

****

**OUTPUT:**

****