Bigdata Assignment 1.6

We have a dataset of sales of different TV sets across different locations.
Records look like:
Samsung|Optima|14|Madhya Pradesh|132401|14200
The fields are arranged like:
Company Name|Product Name|Size in inches|State|Pin Code|Price
There are some invalid records which contain 'NA' in either Company Name or Product Name.
1. Write a Map Reduce program to filter out the invalid records. Map only job will fit for this
context.

Answer

Driver code-

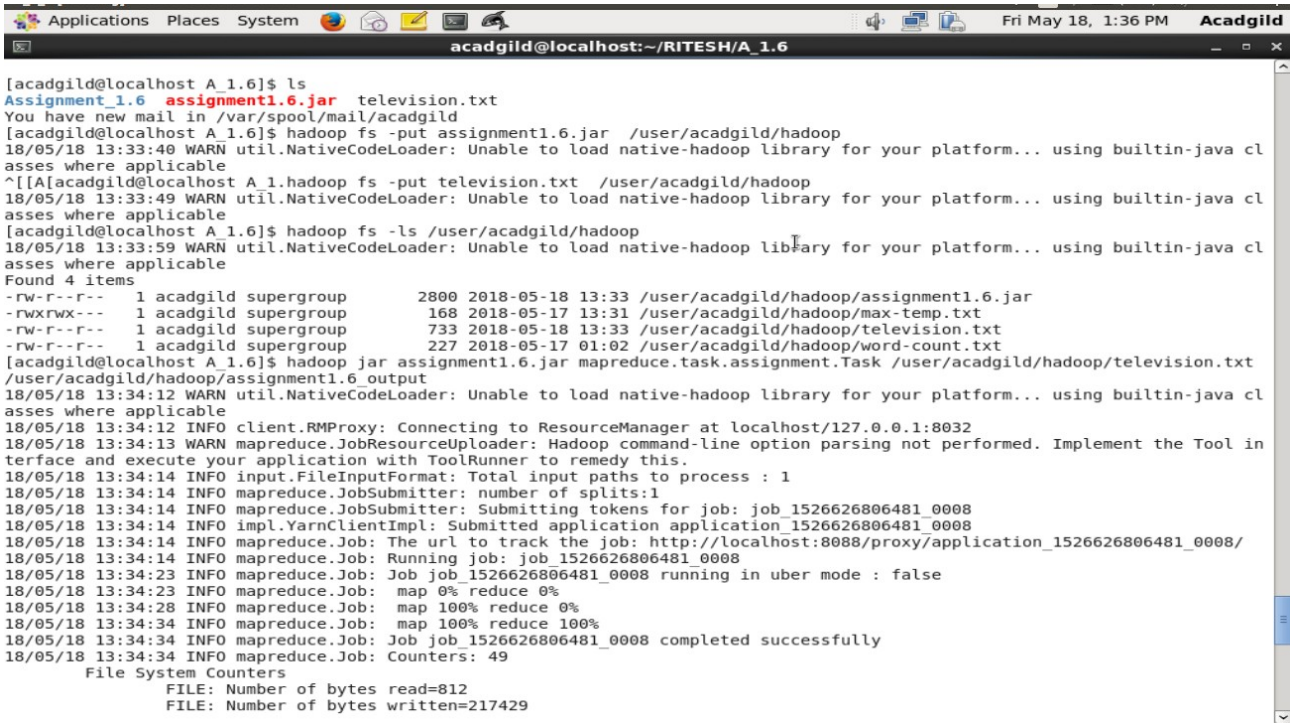
```
package mapreduce.task.assignment;

import org.apache.hadoop.fs.Path;
import org.apache.hadoop.conf.*;
import org.apache.hadoop.mapreduce.Job;
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;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;

public class Task {
        public static void main(String[] args) throws Exception {
                Configuration conf = new Configuration();
                Job job = new Job(conf, "Assignmnet1.6");
                job.setJarByClass(Task.class);
            //Key is LongWritable as it is the index
                job.setMapOutputKeyClass(LongWritable.class);
                //Value is Text as the whole string is required
                job.setMapOutputValueClass(Text.class);

            //Key is LongWritable as it is the index
                job.setOutputKeyClass(LongWritable.class);
                //Value is Text as the whole string is required
```

```
		job.setOutputValueClass(Text.class);
		job.setMapperClass(TaskMapper.class);

		job.setInputFormatClass(TextInputFormat.class);
		job.setOutputFormatClass(TextOutputFormat.class);

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

		/*
		Path out=new Path(args[1]);
		out.getFileSystem(conf).delete(out);
		*/

		job.waitForCompletion(true);
	}
}
```

Mapper Code -

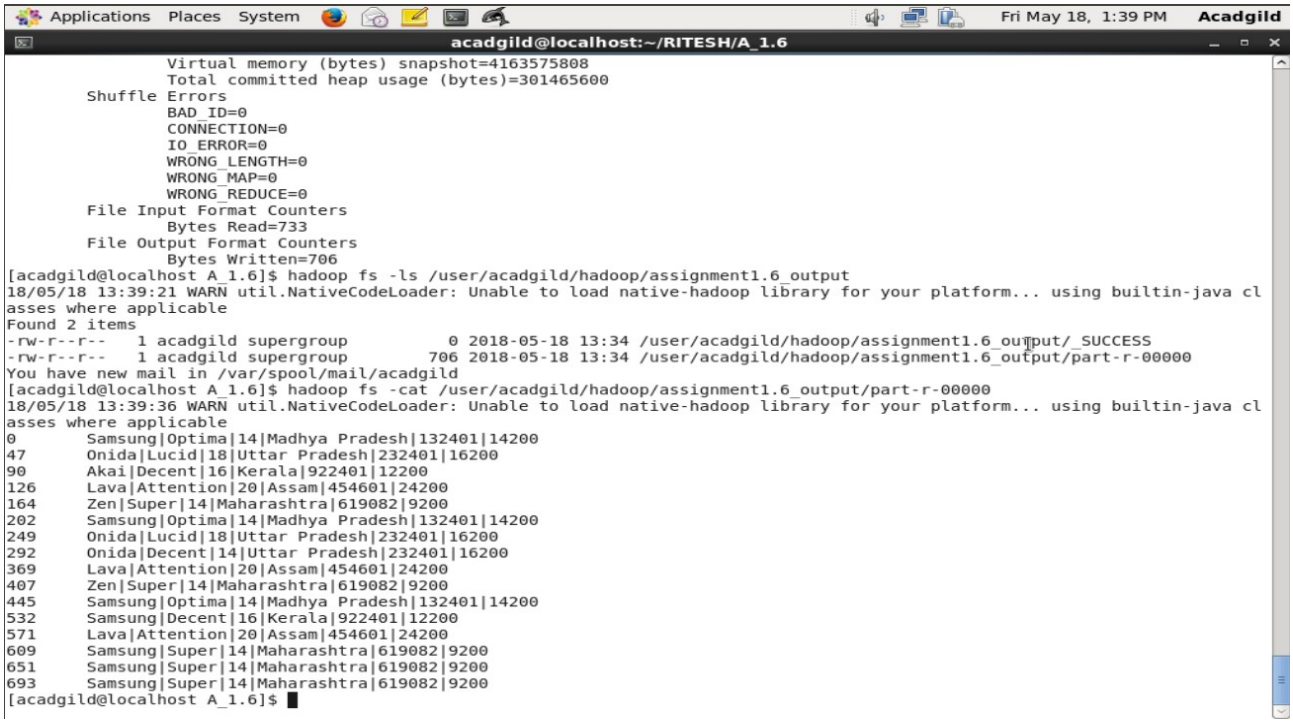
```
package mapreduce.task.assignment;

import java.io.IOException;

import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.*;

public class TaskMapper extends Mapper<LongWritable, Text,LongWritable,Text> {

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

		String company = lineArray[0];
		String product = lineArray[1];
		//Checking if company name or product name must not equal to NA
		if(!(company.equals("NA")||product.equals("NA")))
		{
			context.write(key,value);
		}
```

```
    }
}
```



Execution Steps:-

- television.txt was put to hdfs.
- After the driver code and mapping code , a jar file of was generated.
- assignmen1.6.jar file(mapping code) was put to hdfs.
- Mapping task was performed by the following command – **hadoop jar assignmen1.6.jar mapreduce.task.assignment.task /user/acadgild/hadoop/television.txt /user/acadgild/hadoop/assignment1.6_output**
- To see the content of the output following command was used - **hadoop cat /user/acadgild//hadoop/assignment1.6_output/part-r-00000**



Output

As in the above screen shot ,it is evident that the content of the output file does not contain any records whose company name or product name equals “NA”.