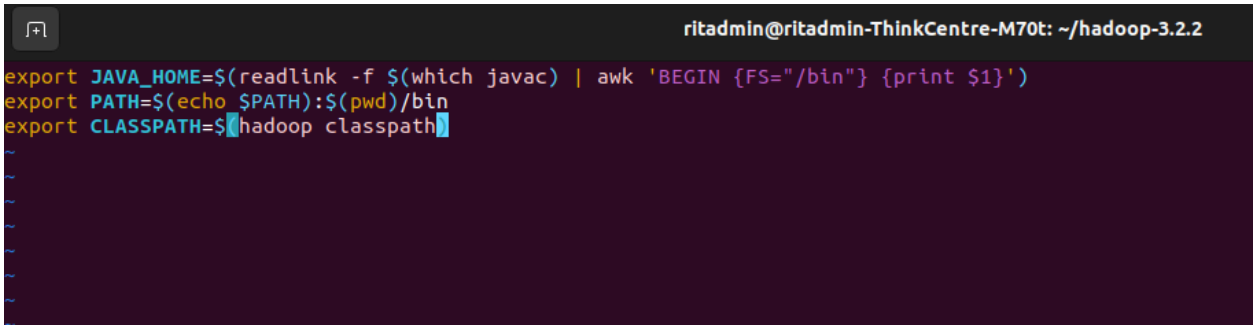
BIG DATA LAB 3
NAME: JESWIN MS
USN: 1MS21CI024

Objective: How to write a mapreduce odd even program and execute on hadoop

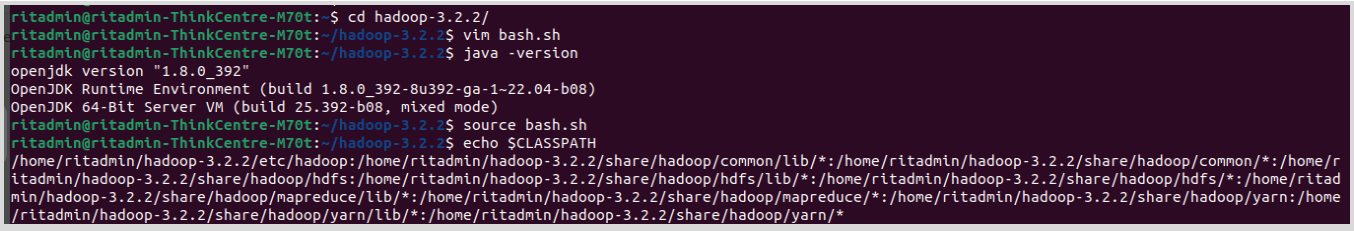
Setup:

- We download the `hadoop-3.2.2.tar.gz` file and extract the file contents in a folder in home directory
- We create a bash.sh file inside the folder using `vim bash.sh.` If vim is not installed, use `sudo apt install vim`
- Next we paste this content below inside the bash.sh file.

```
export JAVA_HOME=$(readlink -f $(which javac) | awk 'BEGIN {FS="/bin"} {print $1}')
export PATH=$(echo $PATH):$(pwd)/bin
export CLASSPATH=$(hadoop classpath)
```



- Run the command `source bash.sh`



- Now the environment is set up in order to run the map reduce program.

Instructions for running odd even program:

driver.java

```
package oddeven;

import java.io.*;

import java.util.*;

import org.apache.hadoop.mapred.*;

import org.apache.hadoop.io.*;

import org.apache.hadoop.fs.Path;



public class driver

{

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

{

      JobConf conf=new JobConf(driver.class);

      conf.setMapperClass(mapper.class);

      conf.setReducerClass(reducer.class);

      conf.setOutputKeyClass(Text.class);

      conf.setOutputValueClass(IntWritable.class);

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

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

      JobClient.runJob(conf);

}

}
```

mapper.java

```
package oddeven;

import java.io.*;

import java.util.*;

import org.apache.hadoop.mapred.*;

import org.apache.hadoop.io.*;



public class mapper extends MapReduceBase implements
Mapper<LongWritable , Text , Text , IntWritable>

{

public void map(LongWritable key,Text
value,OutputCollector<Text,IntWritable> output,Reporter r)
throws IOException

{

      String[] line=value.toString().split(" ");

      for(String num:line){

            int number=Integer.parseInt(num);

            if(number%2==0) {

                  output.collect(new Text("even"),new
IntWritable(number));

            }

            else{

                  output.collect(new Text("odd"),new
IntWritable(number));

            }

      }

}
```

```
}
```

reducer.java

```
package oddeven;

import java.io.*;

import java.util.*;

import org.apache.hadoop.mapred.*;

import org.apache.hadoop.io.*;

public class reducer extends MapReduceBase implements
Reducer<Text,IntWritable,Text,IntWritable>

{

public void reduce(Text key,Iterator<IntWritable>
value,OutputCollector<Text,IntWritable> output ,Reporter r)
throws IOException

{

      int sum=0,count=0;

      while(value.hasNext()){

            sum+=value.next().get();

            count++;

      }

      output.collect(new Text("Sum of "+key+" Numbers"),new
IntWritable(sum));

      output.collect(new Text(key+" Number count"),new
IntWritable(count));

}

}
```

- Create a Directory named `oddeven` inside the hadoop directory which was initially setup and move to it.(mkdir and cd)

- Create 3 java files(driver,mapper and reducer) as shown above and save them (vim <file-name>.java and :wq to save the file
- Run `javac -d . *.java` inorder to compile all the programs present inside the wordcount directory and the respective classes will be saved inside a folder named wordcount in the same directory.
- Run `echo Main-Class: oddeven.driver > Manifest.txt`
- Run `jar cfm oddeven.jar Manifest.txt oddeven/*.class`
- Create an input text file (vi oeinput.txt) specifying the text on which the wordcount operation will be performed.
- `hadoop jar oddeven.jar oeinput.txt output` Inorder to run the program on the input.txt file and store the results on output.
- `cat output/*` to display the output of the mapreduce program.