Instructions For Running Map Reduce Program

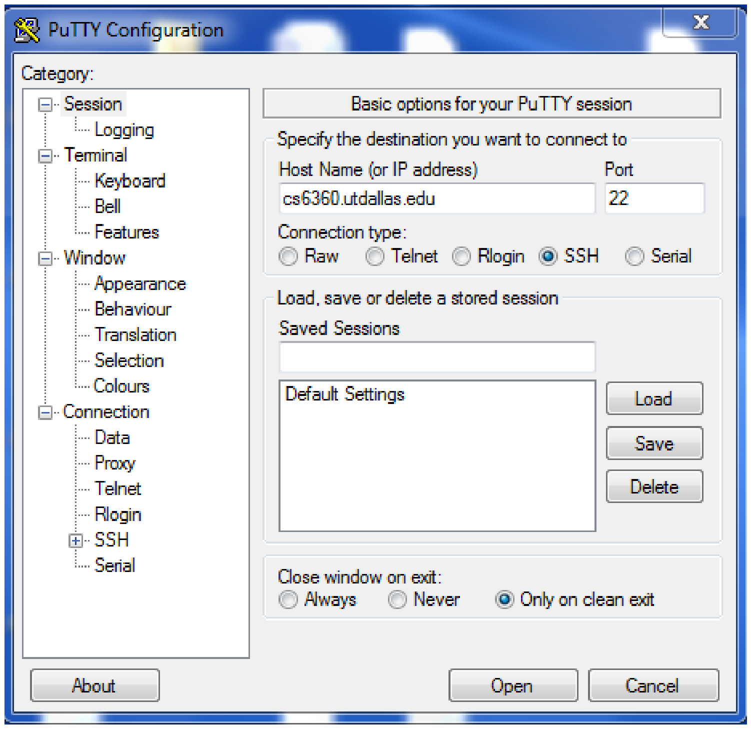
How to log in by SSH terminal?

Download and install **PuTTY**.

Run Putty

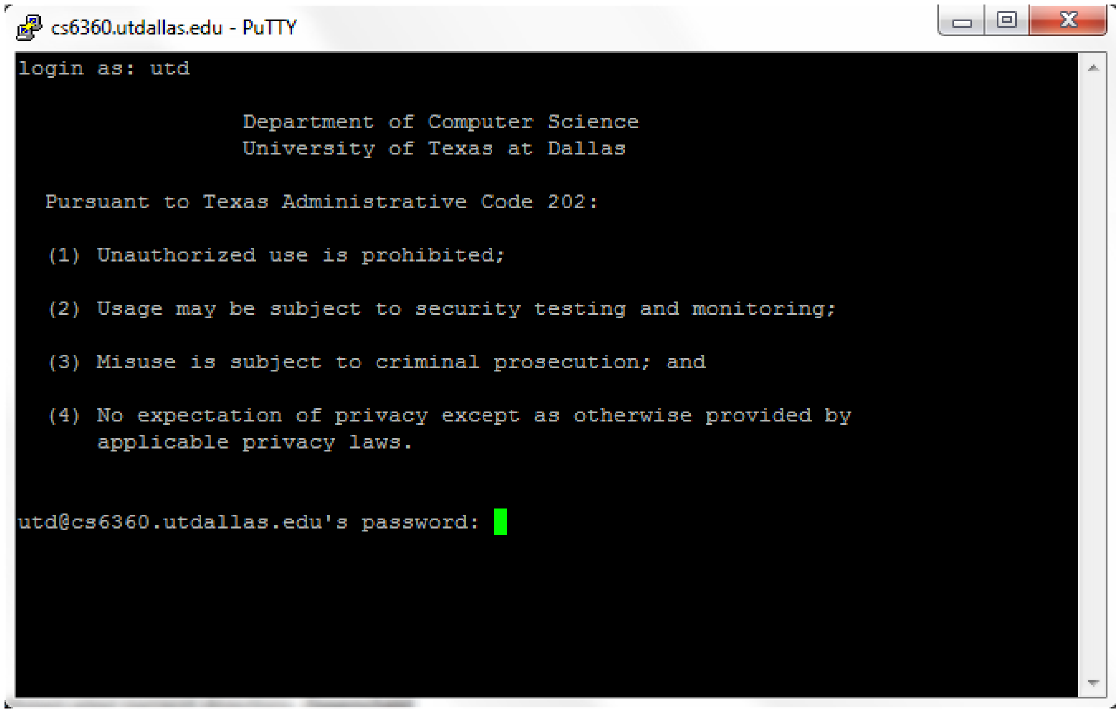
1. Log in **cs6360.utdallas.edu** with user: **utd** password: **hadoop.**

In PuTTY Configuration type cs6360.utdallas.edu and click Open.



Specify the destination you want to connect to
Host Name (or IP address): cs6360.utdallas.edu
Port: 22
Connection type: Raw Telnet Rlogin [✓ SSH] Serial
Close window on exit: Always Never [✓ Only on clean exit]

The following dialog will come. Put user name **utd** and password **hadoop**

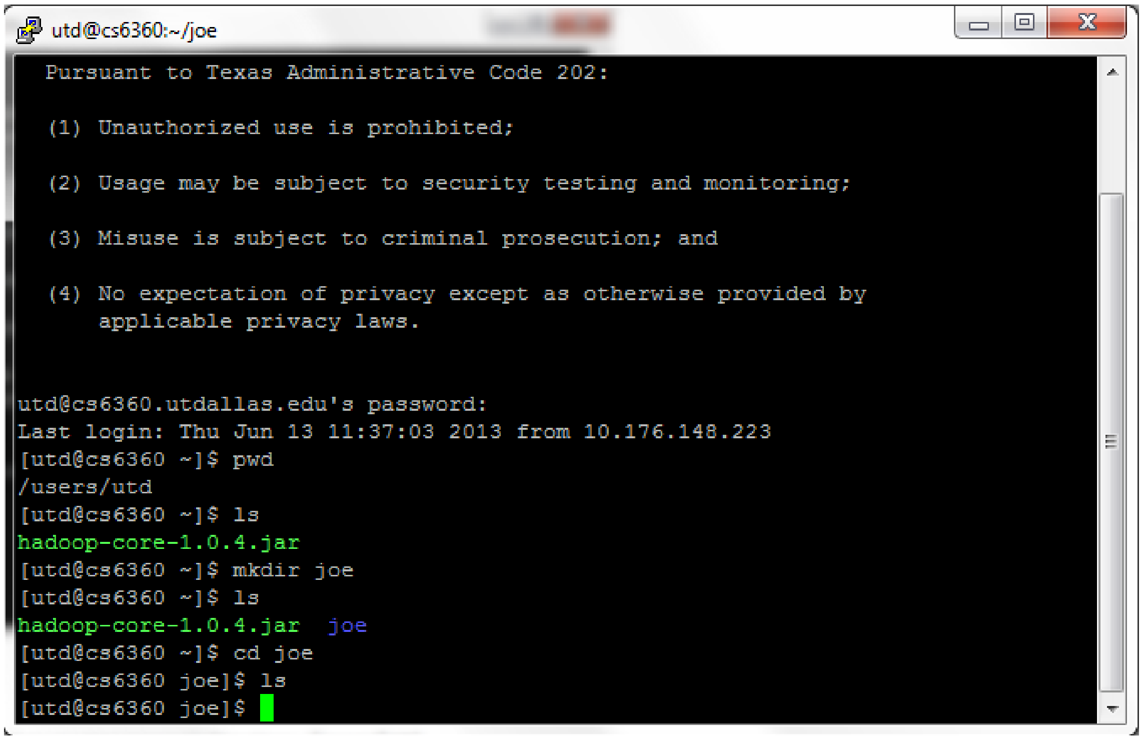


```
login as: utd

                 Department of Computer Science
                 University of Texas at Dallas

  Pursuant to Texas Administrative Code 202:

  (1) Unauthorized use is prohibited;

  (2) Usage may be subject to security testing and monitoring;

  (3) Misuse is subject to criminal prosecution; and

  (4) No expectation of privacy except as otherwise provided by
      applicable privacy laws.


utd@cs6360.utdallas.edu's password:
```

2. After log in verify your current directory by **pwd** command on ssh terminal. **pwd** shows your current directory **/users/utd**

3. Make sure that you have the **hadoop-core-1.0.4.jar** file in **/users/utd**. you can list all files and folder in **/users/utd** by **ls** command on terminal.

4. Create a Folder by **mkdir** command with your name(e.g. **Joe**) inside **/users/utd** . Invoke your directory by issuing **cd** command on terminal .



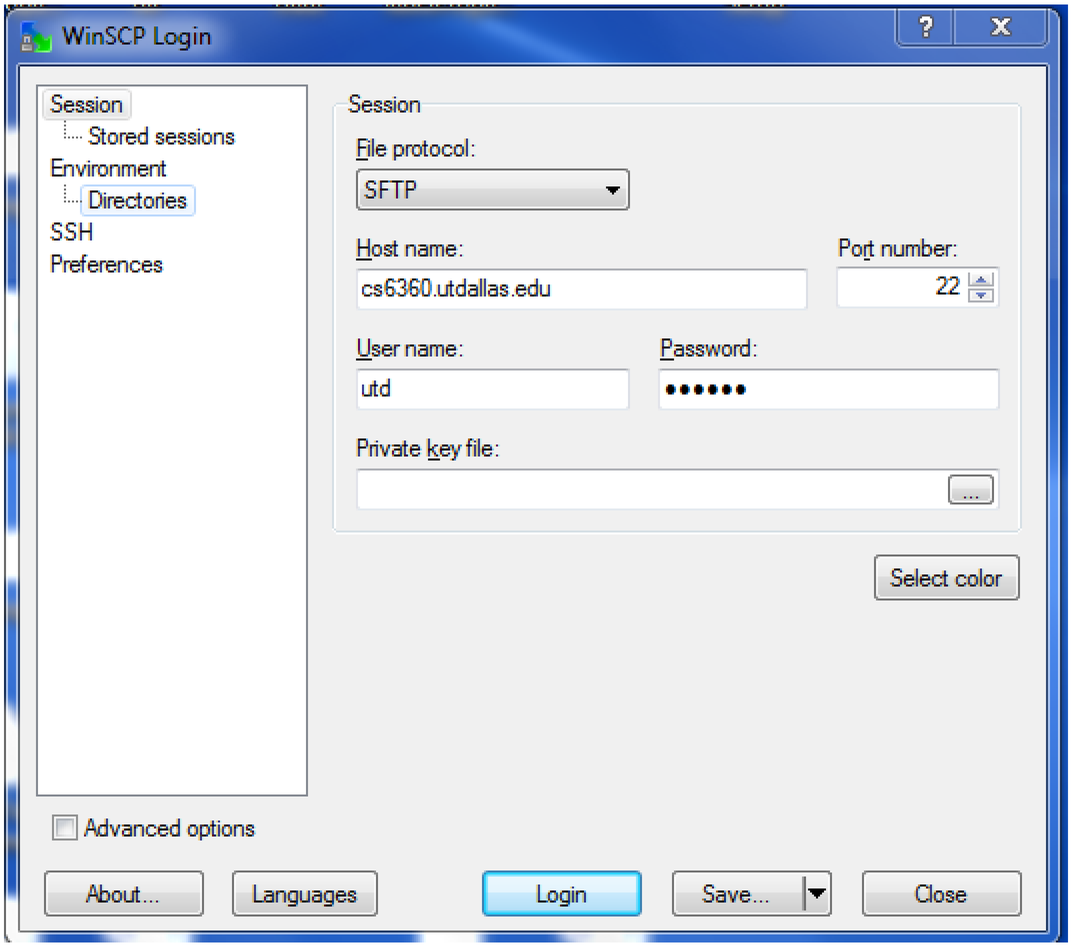
```
 Pursuant to Texas Administrative Code 202:

  (1) Unauthorized use is prohibited;

  (2) Usage may be subject to security testing and monitoring;

  (3) Misuse is subject to criminal prosecution; and

  (4) No expectation of privacy except as otherwise provided by
      applicable privacy laws.


utd@cs6360.utdallas.edu's password:
Last login: Thu Jun 13 11:37:03 2013 from 10.176.148.223
[utd@cs6360 ~]$ pwd
/users/utd
[utd@cs6360 ~]$ ls
hadoop-core-1.0.4.jar
[utd@cs6360 ~]$ mkdir joe
[utd@cs6360 ~]$ ls
hadoop-core-1.0.4.jar  joe
[utd@cs6360 ~]$ cd joe
[utd@cs6360 joe]$ ls
[utd@cs6360 joe]$
```

How to upload files in cs6360.utdallas.edu?

Download and install WinSCP.

Run WinSCP

1. Log in **cs6360.utdallas.edu** with user: **utd** password: **hadoop.**



Session

File protocol: SFTP

Host name: cs6360.utdallas.edu

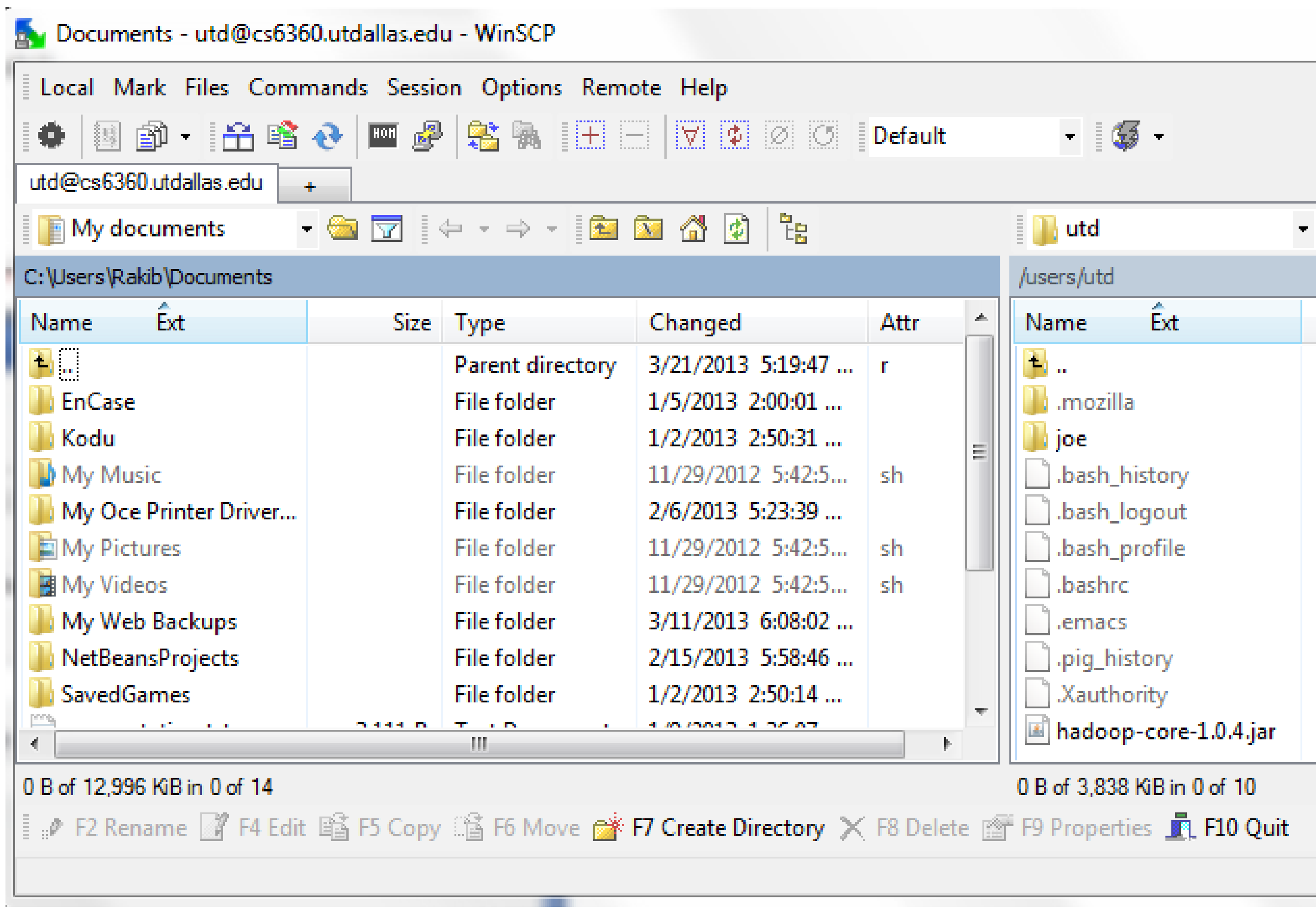
Port number: 22

User name: utd

Password: ●●●●●●

Private key file:

N.B. A warning message will be prompted. Click Yes and Continue (shown in next dialog) will give you the following dialog.


Documents - utd@cs6360.utdallas.edu - WinSCP
Local Mark Files Commands Session Options Remote Help
Default
utd@cs6360.utdallas.edu
My documents
utd
F2 Rename F4 Edit F5 Copy F6 Move F7 Create Directory F8 Delete F9 Properties F10 Quit

C:\Users\Rakib\Documents

Name	Ext	Size	Type	Changed	Attr
..			Parent directory	3/21/2013 5:19:47 ...	r
EnCase			File folder	1/5/2013 2:00:01 ...	
Kodu			File folder	1/2/2013 2:50:31 ...	
My Music			File folder	11/29/2012 5:42:5...	sh
My Oce Printer Driver...			File folder	2/6/2013 5:23:39 ...	
My Pictures			File folder	11/29/2012 5:42:5...	sh
My Videos			File folder	11/29/2012 5:42:5...	sh
My Web Backups			File folder	3/11/2013 6:08:02 ...	
NetBeansProjects			File folder	2/15/2013 5:58:46 ...	
SavedGames			File folder	1/2/2013 2:50:14 ...	

0 B of 12,996 KiB in 0 of 14

/users/utd

Name	Ext
..	
.mozilla	
joe	
.bash_history	
.bash_logout	
.bash_profile	
.bashrc	
.emacs	
.pig_history	
.Xauthority	
hadoop-core-1.0.4.jar	

0 B of 3,838 KiB in 0 of 10

The left side contains the local and the right side shows the remote pc. For uploading any file from local to remote just drag the file from local to remote directory.

How to Compile the .java file and build .jar file?

You can compile java file and also build jar file by using any standard IDE like Eclipse / NetBeans or any stand java IDE.

On the other hand you can compile java file and then create jar file from command line on Putty terminal

Step 1: **javac -classpath <hadoop-core-1.0.4.jar file path> <java file path>**

Step 2: **jar cvf <jar file> -C <manifest file directory> <class file directory>**

Home Work 1

In this homework you will learn how to solve problems using Map Reduce.

Please apply Hadoop mapreduce to derive some statistics from White House Visitor Log. There are currently 2.9 million records available at

http://www.whitehouse.gov/briefing-room/disclosures/visitor-records

Data is available as web only spreadsheet view and downloadable raw format in CSV (Comma Separated Value). In CSV format each column is separated by a comma "," in each line. The first line represents the heading for the corresponding columns in other lines. We are going to use this raw data for our mapreduce operation.

Download the 4 CSV files (zipped) from the site (at the bottom of the page), unzip, and copy to the UTD's HDFS directory - "/home/<your_utd_netid>/whitehouse/input". Use the *put* or *copyFromLocal* HDFS shell command to copy those files. Also produce the output to "/home/<your_name>/output" HDFS directory. For more information about HDFS shell, see here - http://hadoop.apache.org/docs/stable/file_system_shell.html

You are required to write efficient Hadoop ***MapReduce programs in Java*** to find the following information:

N.B. The customized dataset will be found at **/home/kma041000/HW1/input** and you should use this as input.

You are required to write efficient Hadoop ***MapReduce programs in Java*** to find the following information:

(i) The 10 most frequent visitors (NAMELAST, NAMEFIRST) to the White House.

(ii) The 10 most frequently visited people (visitee_namelast, visitee_namefirst) in the White House.

(iii) The 10 most frequent visitor-visitee combinations.

(iv) Average number of visitors by month of a year. Say what is the average number of visitors in January over last 4 years. Your program should produce average number of visitors for 12 months. Consider APPT_START_DATE as the visit date value.

Consider mapreduce chaining for efficiency. See the possible ways for chaining in Yahoo's tutorial and the stackoverflow discussion
http://developer.yahoo.com/hadoop/tutorial/module4.html#chaining
http://stackoverflow.com/questions/3059736/map-reduce-chainmapper-and-chainreducer

Home Work 1(i): The 10 most frequent visitors (NAMELAST, NAMEFIRST) to the White House.

1. Create a folder HW1 inside folder Joe(folder created earlier).

```
mkdir HW1
```

2. Inside HW1 folder copy/upload WhiteHouse.java by WinSCP

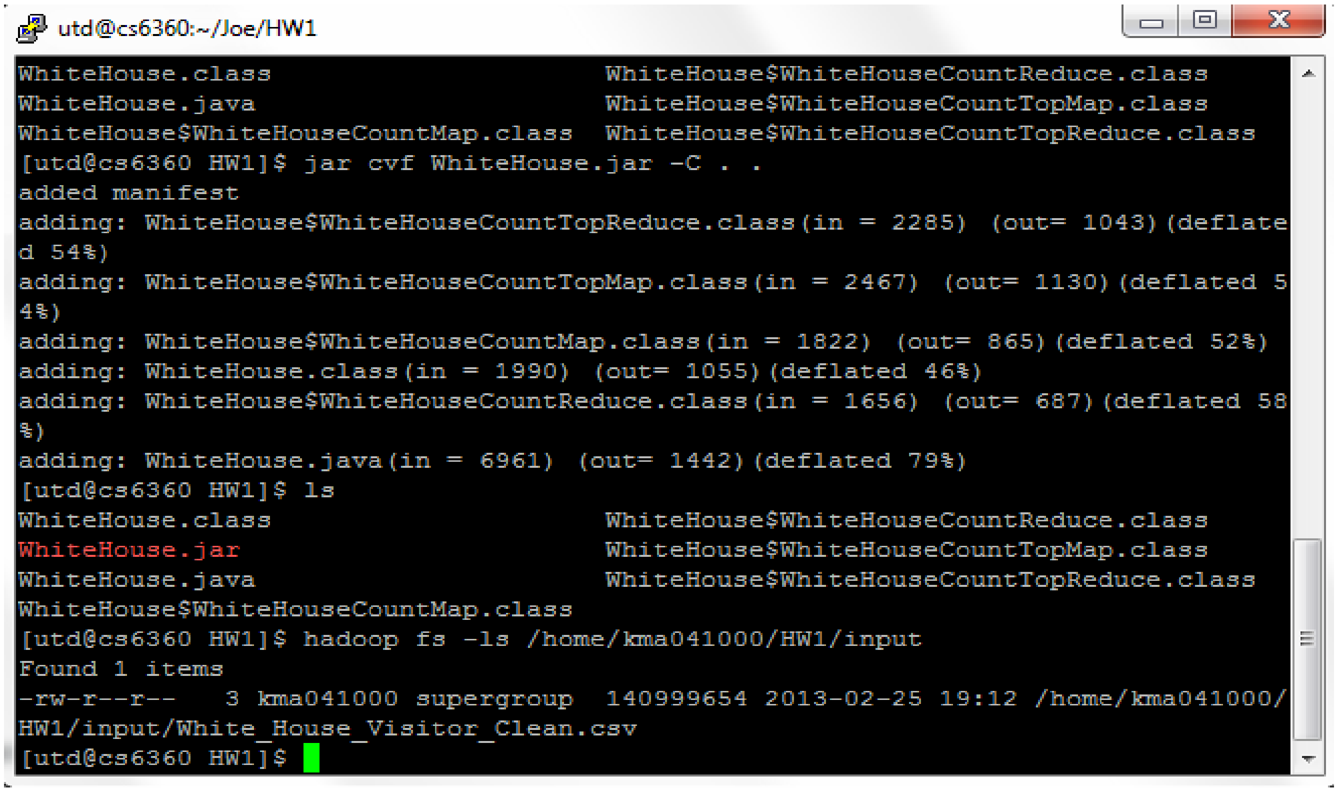
3. Run the following commands:

```
javac -classpath ../../hadoop-core-1.0.4.jar  WhiteHouse.java

jar cvf WhiteHouse.jar -C . .
```

4. You will see a jar file named WhiteHouse.jar is created in HW1 folder

5. Make sure that input files are inserted proper directory.

```
hadoop fs -ls /home/kma041000/HW1/input
```



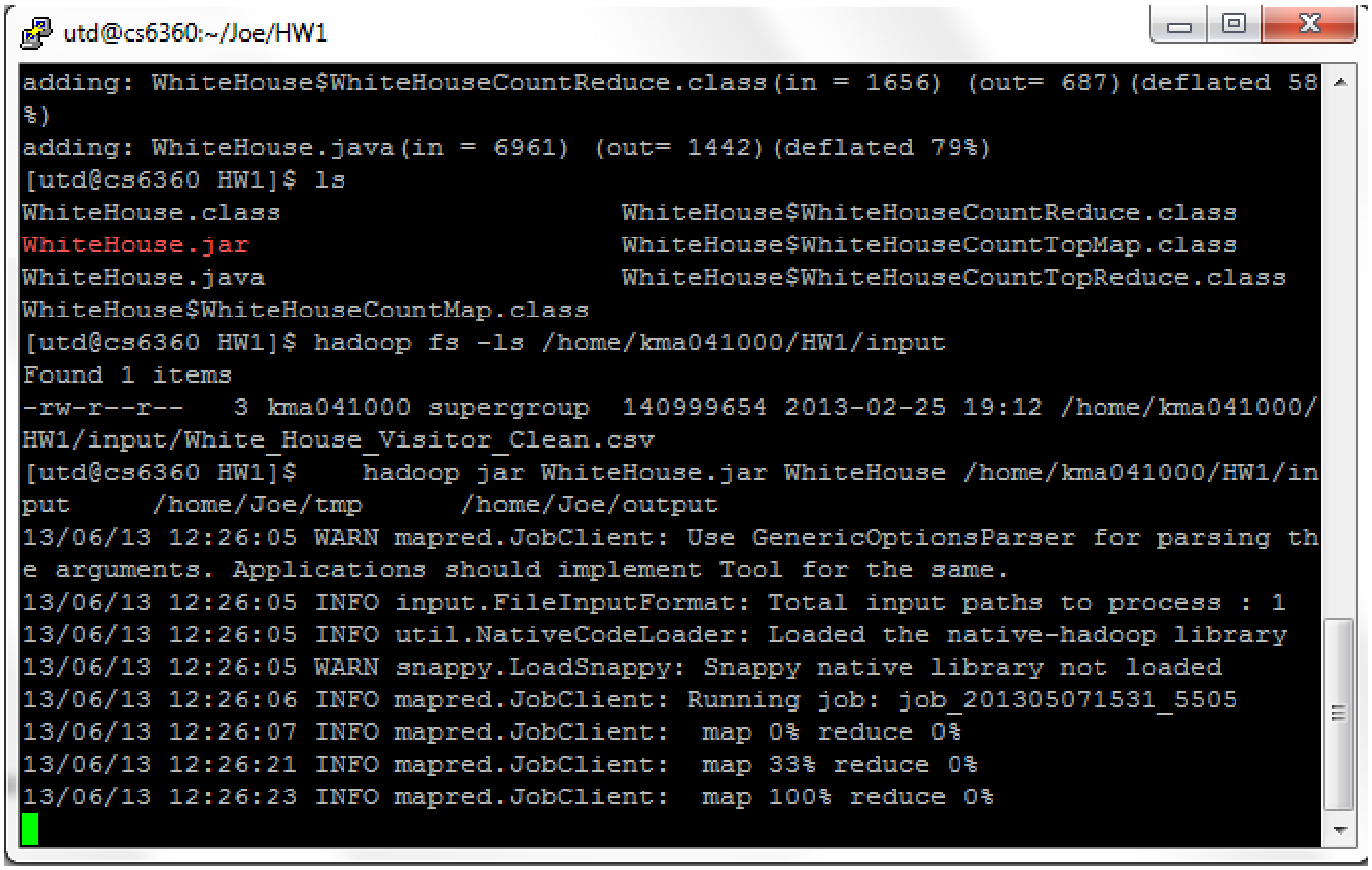
```
WhiteHouse.class                     WhiteHouse$WhiteHouseCountReduce.class
WhiteHouse.java                      WhiteHouse$WhiteHouseCountTopMap.class
WhiteHouse$WhiteHouseCountMap.class  WhiteHouse$WhiteHouseCountTopReduce.class
[utd@cs6360 HW1]$ jar cvf WhiteHouse.jar -C . .
added manifest
adding: WhiteHouse$WhiteHouseCountTopReduce.class(in = 2285) (out= 1043)(deflate
d 54%)
adding: WhiteHouse$WhiteHouseCountTopMap.class(in = 2467) (out= 1130)(deflated 5
4%)
adding: WhiteHouse$WhiteHouseCountMap.class(in = 1822) (out= 865)(deflated 52%)
adding: WhiteHouse.class(in = 1990) (out= 1055)(deflated 46%)
adding: WhiteHouse$WhiteHouseCountReduce.class(in = 1656) (out= 687)(deflated 58
%)
adding: WhiteHouse.java(in = 6961) (out= 1442)(deflated 79%)
[utd@cs6360 HW1]$ ls
WhiteHouse.class                     WhiteHouse$WhiteHouseCountReduce.class
WhiteHouse.jar                       WhiteHouse$WhiteHouseCountTopMap.class
WhiteHouse.java                      WhiteHouse$WhiteHouseCountTopReduce.class
WhiteHouse$WhiteHouseCountMap.class
[utd@cs6360 HW1]$ hadoop fs -ls /home/kma041000/HW1/input
Found 1 items
-rw-r--r--   3 kma041000 supergroup  140999654 2013-02-25 19:12 /home/kma041000/
HW1/input/White_House_Visitor_Clean.csv
[utd@cs6360 HW1]$ 
```

6. Run map reduce program

hadoop jar WhiteHouse.jar WhiteHouse /home/kma041000/HW1/input /home/Joe/tmp /home/Joe/output

(For chaining purpose the java program requires three file locations. First is input file location, the second is the intermediate file location and the third one is the final output file location.)

This will gives you following which indicates that map and reduce job status

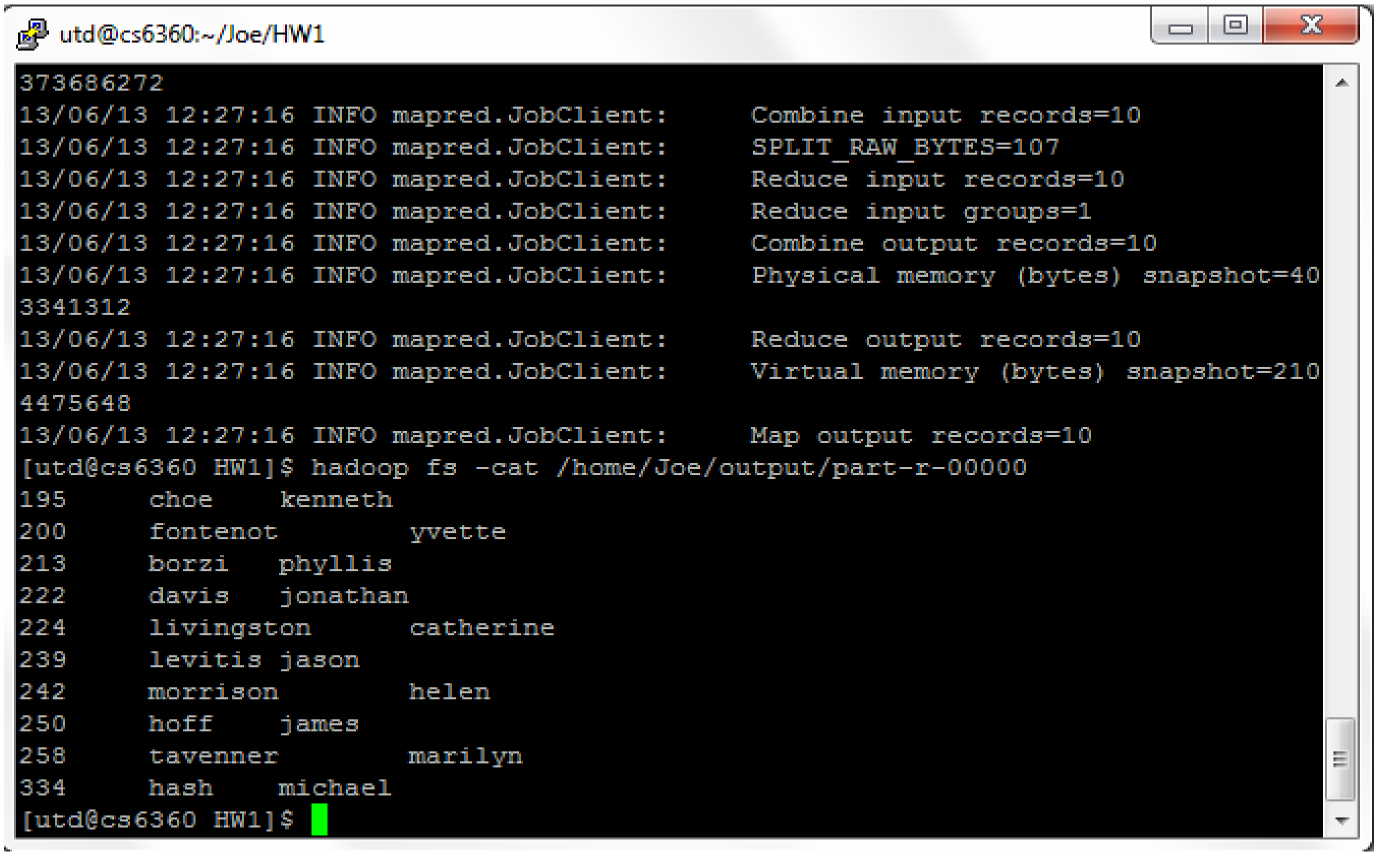

utd@cs6360:~/Joe/HW1

```
adding: WhiteHouse$WhiteHouseCountReduce.class(in = 1656) (out= 687)(deflated 58
%)
adding: WhiteHouse.java(in = 6961) (out= 1442)(deflated 79%)
[utd@cs6360 HW1]$ ls
WhiteHouse.class                    WhiteHouse$WhiteHouseCountReduce.class
WhiteHouse.jar                      WhiteHouse$WhiteHouseCountTopMap.class
WhiteHouse.java                     WhiteHouse$WhiteHouseCountTopReduce.class
WhiteHouse$WhiteHouseCountMap.class
[utd@cs6360 HW1]$ hadoop fs -ls /home/kma041000/HW1/input
Found 1 items
-rw-r--r--   3 kma041000 supergroup  140999654 2013-02-25 19:12 /home/kma041000/
HW1/input/White_House_Visitor_Clean.csv
[utd@cs6360 HW1]$   hadoop jar WhiteHouse.jar WhiteHouse /home/kma041000/HW1/in
put     /home/Joe/tmp      /home/Joe/output
13/06/13 12:26:05 WARN mapred.JobClient: Use GenericOptionsParser for parsing th
e arguments. Applications should implement Tool for the same.
13/06/13 12:26:05 INFO input.FileInputFormat: Total input paths to process : 1
13/06/13 12:26:05 INFO util.NativeCodeLoader: Loaded the native-hadoop library
13/06/13 12:26:05 WARN snappy.LoadSnappy: Snappy native library not loaded
13/06/13 12:26:06 INFO mapred.JobClient: Running job: job_201305071531_5505
13/06/13 12:26:07 INFO mapred.JobClient:  map 0% reduce 0%
13/06/13 12:26:21 INFO mapred.JobClient:  map 33% reduce 0%
13/06/13 12:26:23 INFO mapred.JobClient:  map 100% reduce 0%
```

7. You will see the **part-r-00000** file in **/home/Joe/output** directory. The following command will give you the output.

hadoop fs -cat /home/Joe/output/part-r-00000

This will give the following output.


utd@cs6360:~/Joe/HW1

```
373686272
13/06/13 12:27:16 INFO mapred.JobClient:     Combine input records=10
13/06/13 12:27:16 INFO mapred.JobClient:     SPLIT_RAW_BYTES=107
13/06/13 12:27:16 INFO mapred.JobClient:     Reduce input records=10
13/06/13 12:27:16 INFO mapred.JobClient:     Reduce input groups=1
13/06/13 12:27:16 INFO mapred.JobClient:     Combine output records=10
13/06/13 12:27:16 INFO mapred.JobClient:     Physical memory (bytes) snapshot=40
3341312
13/06/13 12:27:16 INFO mapred.JobClient:     Reduce output records=10
13/06/13 12:27:16 INFO mapred.JobClient:     Virtual memory (bytes) snapshot=210
4475648
13/06/13 12:27:16 INFO mapred.JobClient:     Map output records=10
[utd@cs6360 HW1]$ hadoop fs -cat /home/Joe/output/part-r-00000
195     choe    kenneth
200     fontenot        yvette
213     borzi   phyllis
222     davis   jonathan
224     livingston      catherine
239     levitis jason
242     morrison        helen
250     hoff    james
258     tavenner        marilyn
334     hash    michael
[utd@cs6360 HW1]$
```

WhiteHouse.java

```
import java.io.IOException;

import java.util.TreeMap;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.fs.FileSystem;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.NullWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Job;

import org.apache.hadoop.mapreduce.Mapper;

import org.apache.hadoop.mapreduce.Reducer;

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

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



/**

 *

 * @author Rakib

 */

public class WhiteHouse {


    public static class WhiteHouseCountMap extends Mapper<Object, Text, Text, IntWritable>
    {
        private final static IntWritable one = new IntWritable(1);
        private Text word = new Text();

     @Override
     public void map(Object key, Text value, org.apache.hadoop.mapreduce.Mapper.Context context) throws
IOException, InterruptedException {

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

      //StringTokenizer tokenizer = new StringTokenizer(line, ",");
          if (container.length > 0) {
            String firstName = container[1].trim();
       String lastName = container[0].trim();

       if (!"NAMELAST".equals(lastName) && !"NAMEFIRST".equals(firstName)) {
           String name = lastName + "\t" + firstName;
           word.set(name);
           context.write(word, one);
         }
           }
         }
        }

    public static class WhiteHouseCountReduce extends Reducer<Text, IntWritable, Text, IntWritable> {
     @Override
```

```
            public void reduce(Text key, Iterable<IntWritable> values, Context context
             ) throws IOException, InterruptedException {
              int sum = 0;
              for (IntWritable val : values) {
           sum += val.get();
          }

              context.write(key, new IntWritable(sum));
            }
          }



   public static class WhiteHouseCountTopMap extends Mapper<Object, Text, NullWritable, Text> {



    private TreeMap<Integer, Text> visitorToRecordMap = new TreeMap<Integer, Text>();


    @Override
            public void map(Object key, Text value, org.apache.hadoop.mapreduce.Mapper.Context context)
throws IOException, InterruptedException {
            String[] container = value.toString().split("\t");


     if (container.length > 0)
     {

        String containerValue = container[2].trim() + "\t" + container[0].trim() + "\t" + container[1].trim();
```

```
      visitorToRecordMap.put(Integer.parseInt(container[2].trim()), new Text(containerValue));


     if (visitorToRecordMap.size() > 10)

     {

        visitorToRecordMap.remove(visitorToRecordMap.firstKey());

     }

   }

       }


  @Override

  protected void cleanup(org.apache.hadoop.mapreduce.Mapper.Context context) throws IOException,

    InterruptedException {

    // Output our ten records to the reducers with a null key

    for (Text t : visitorToRecordMap.values()) {


       context.write(NullWritable.get(), t);


    }

    }

}


public static class WhiteHouseCountTopReduce extends Reducer<NullWritable, Text, NullWritable, Text> {


private TreeMap<Integer, Text> visitorToRecordMap = new TreeMap<Integer, Text>();


public void reduce(NullWritable key, Iterable<Text> values, Reducer.Context context

          ) throws IOException, InterruptedException {
```

```
for (Text val : values)
{

String value = val.toString();
String[] container = value.split("\t");

 //StringTokenizer tokenizer = new StringTokenizer(line, ",");
        if (container.length > 0)
 {
           String count = container[0].trim();
   visitorToRecordMap.put(Integer.parseInt(count), new Text(value));

   if (visitorToRecordMap.size() > 10)
   {
       visitorToRecordMap.remove(visitorToRecordMap.firstKey());
   }
 }
}


 for (Text t : visitorToRecordMap.values()) {
    //System.out.println(t.toString());
    context.write(NullWritable.get(), t);
 }
      }
     }
```

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



Configuration conf = new Configuration();



if (args.length != 3) {
  System.err.println("Usage: whitehouse <in> <temp> <out>");
  System.exit(2);
}

FileSystem fs = FileSystem.get(conf);
        fs.delete(new Path(args[1]), true);
        fs.delete(new Path(args[2]), true);

Job countingJob = new Job(conf, "whitehouse_count");
countingJob.setJarByClass(WhiteHouse.class);
countingJob.setMapperClass(WhiteHouseCountMap.class);
countingJob.setCombinerClass(WhiteHouseCountReduce.class);
countingJob.setReducerClass(WhiteHouseCountReduce.class);

countingJob.setOutputKeyClass(Text.class);
countingJob.setOutputValueClass(IntWritable.class);



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

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

        if (countingJob.waitForCompletion(true))
        {
          Job countingTopJob = new Job(conf, "whitehouse_top");
          countingTopJob.setJarByClass(WhiteHouse.class);
          countingTopJob.setMapperClass(WhiteHouseCountTopMap.class);
          countingTopJob.setCombinerClass(WhiteHouseCountTopReduce.class);
          countingTopJob.setReducerClass(WhiteHouseCountTopReduce.class);

          countingTopJob.setOutputKeyClass(NullWritable.class);
          countingTopJob.setOutputValueClass(Text.class);



          FileInputFormat.addInputPath(countingTopJob, new Path(args[1]));
          FileOutputFormat.setOutputPath(countingTopJob, new Path(args[2]));

          System.exit(countingTopJob.waitForCompletion(true) ? 0 : 1);
        }

              }
  }
```