2. Problem Statement

Read two streams
1. List of strings input by user
2. Real-time set of offensive words
Find the word count of the offensive words inputted by the user as per the real-time set of offensive words.

Solution-

Below is the code which has been used to find the Offensive Words-

```
package org.scala

import org.apache.spark.SparkConf
import org.apache.spark.storage.StorageLevel
import org.apache.spark.streaming.{Seconds, StreamingContext}
import scala.collection.mutable.ArrayBuffer
import org.apache.spark.streaming.dstream.DStream
import org.apache.spark.SparkContext

object OffensiveWordCount {
  //ArrayBuffer to store list of offensive words in memory
  val wordList: ArrayBuffer[String] = ArrayBuffer.empty[String];

  def main(args: Array[String]) {
    if (args.length < 2) {
      System.err.println("Usage: OffensiveWordCount <hostname> <port>")
      System.exit(1)
    }

    StreamingExamples.setStreamingLogLevels()

    // Create the context with a 60 second batch size
    val sparkConf = new SparkConf().setAppName("OffensiveWordCount")
    val ssc = new StreamingContext(sparkConf, Seconds(60))

    //Creating text file stream to store offensive words.

    val offensiveLines = ssc.textFileStream("hdfs://localhost:9000/offensiveWords/");
    val lines = ssc.socketTextStream(args(0), args(1).toInt, StorageLevel.MEMORY_AND_DISK_SER);

    //getting offensive words from file
    val offensiveWordCount = offensiveLines.flatMap(line => line.split(" ")).map(word => (word, 1)).reduceByKey(_ + _);
    //storing offensive words in ArrayBuffer
    offensiveWordCount.foreachRDD(a => { a.foreach(f => {wordList += f._1})});

    //Getting all word count of all words entered by user
    val wordCount = lines.flatMap(line => line.split(" ")).map(word => (word, 1)).reduceByKey(_ + _);

    //Getting word count of offensive words only
    val offensiveWordsRDD = wordCount.filter {x => matchWord(x._1)%2==1 };
    offensiveWordsRDD.print();

    ssc.start()
    ssc.awaitTermination()
  }

  /**
    * Filter Method for offensive words
    */
  def matchWord(ln : String): Double={
    val lineWords = ln.trim.toLowerCase();
    var num: Double = 0;

    for(y<-wordList)
    {
      if(y.toLowerCase() == lineWords)
      {
        num = 1;
        return num;
      }

    }

    return num;
  }

}
```

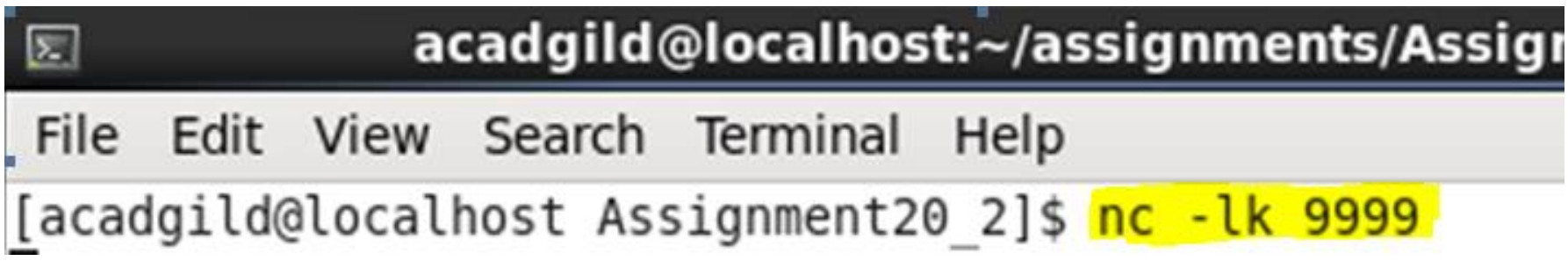
Below is the code for Streaming-

```
package org.scala
import org.apache.log4j.{Level, Logger}
import org.apache.spark.internal.Logging

/** Utility functions for Spark Streaming examples. */
object StreamingExamples extends Logging {

  /** Set reasonable logging levels for streaming if the user has not configured log4j. */
  def setStreamingLogLevels(): Unit = {
    val log4jInitialized = Logger.getRootLogger.getAllAppenders.hasMoreElements
    if (!log4jInitialized) {
      // We first log something to initialize Spark's default logging, then we override the
      // logging level.
      logInfo("Setting log level to [WARN] for streaming example." +
```

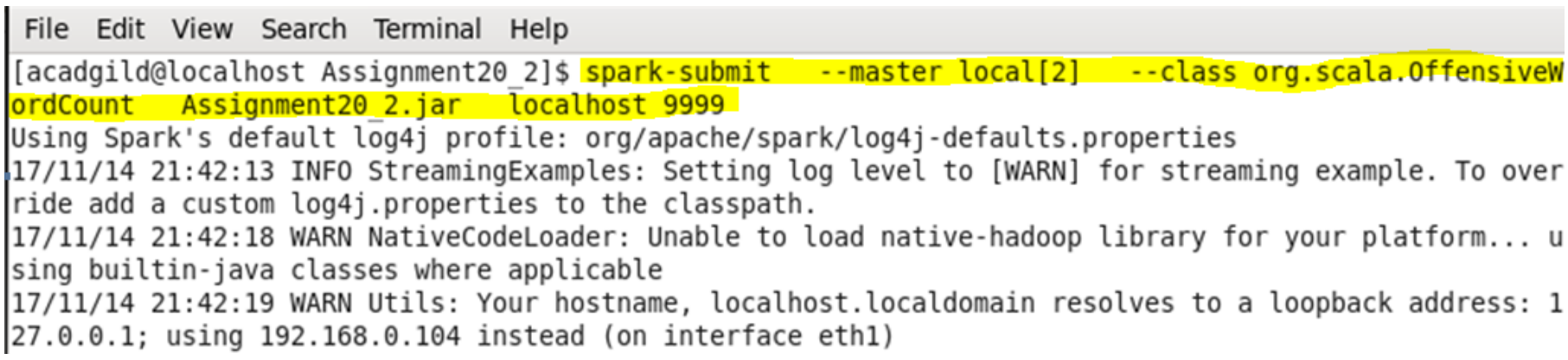
```
        " To override add a custom log4j.properties to the classpath.")
      Logger.getRootLogger.setLevel(Level.WARN)
    }
  }
}
```

Now in order to run the spark application we will start netcat server with below command-

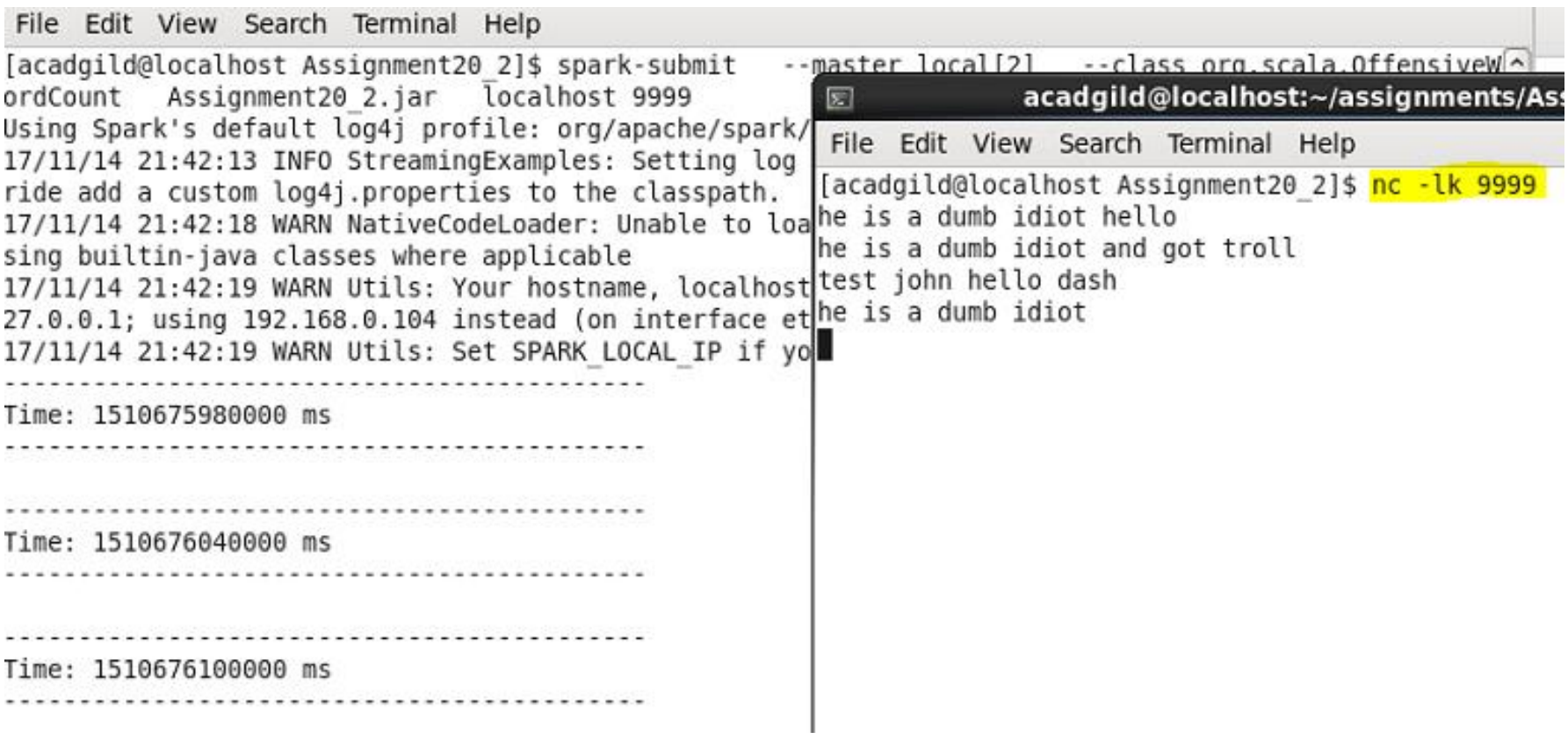
Nc –lk 9999



Now we will transfer the jar to local file system and we will start the streaming application as shown below-



Now we are putting the text input to HDFS and will provide the input from netcat as well as shown below-



Nothing is printed on spark app as the list of offensive words in empty

Now loading first offensive word file

```
File  Edit  View  Search  Terminal  Help
[acadgild@localhost offensiveWords]$ ls
offensiveWordsFile.txt  offWords2.txt
[acadgild@localhost offensiveWords]$ cat offensiveWordsFile.txt
dumb
not
idiot
test
[acadgild@localhost offensiveWords]$ hadoop fs -put offensiveWordsFile.txt /offe
nsiveWords
17/11/14 21:47:43 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
[acadgild@localhost offensiveWords]$ hadoop fs -ls /offensiveWords/
17/11/14 21:48:27 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
Found 1 items
-rw-r--r--   1 acadgild supergroup         20 2017-11-14 21:47 /offensiveWords/o
ffensiveWordsFile.txt
[acadgild@localhost offensiveWords]$
```

Now upon entering same input text, the app is returning word count of offensive words in file OffensiveWordsFile.txt

```
File  Edit  View  Search  Terminal  Help
-------------------------------------------
Time: 1510676100000 ms
-------------------------------------------

-------------------------------------------
Time: 1510676160000 ms
-------------------------------------------

-------------------------------------------
Time: 1510676220000 ms
-------------------------------------------

-------------------------------------------
Time: 1510676280000 ms
-------------------------------------------

-------------------------------------------
Time: 1510676340000 ms
-------------------------------------------

-------------------------------------------
Time: 1510676400000 ms
-------------------------------------------
(dumb,3)
(test,1)
(idiot,3)
```

```
acadgild@localhost:~/assignments/Assignme
File  Edit  View  Search  Terminal  Help
[acadgild@localhost Assignment20_2]$ nc -lk 9999
he is a dumb idiot hello
he is a dumb idiot and got troll
test john hello dash
he is a dumb idiot
he is a dumb idiot hello
he is a dumb idiot and got troll
test john hello dash
he is a dumb idiot
```

Adding another file with offensive words

```
File  Edit  View  Search  Terminal  Help
[acadgild@localhost offensiveWords]$ ls
offensiveWordsFile.txt  offWords2.txt
[acadgild@localhost offensiveWords]$ cat offWords2.txt
troll
john
dash
[acadgild@localhost offensiveWords]$ hadoop fs -put offWords2.txt /offensiveWord
s
17/11/14 21:52:42 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
[acadgild@localhost offensiveWords]$ hadoop fs -ls /offensiveWords/
17/11/14 21:53:33 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
Found 2 items
-rw-r--r--   1 acadgild supergroup         16 2017-11-14 21:52 /offensiveWords/o
ffWords2.txt
-rw-r--r--   1 acadgild supergroup         20 2017-11-14 21:47 /offensiveWords/o
ffensiveWordsFile.txt
[acadgild@localhost offensiveWords]$
```

From below screenshot we could see that the app is now returning the word count of offensive words that were loaded previously as well as loaded later from file offWords2.txt.