Assignment 20.2:

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.

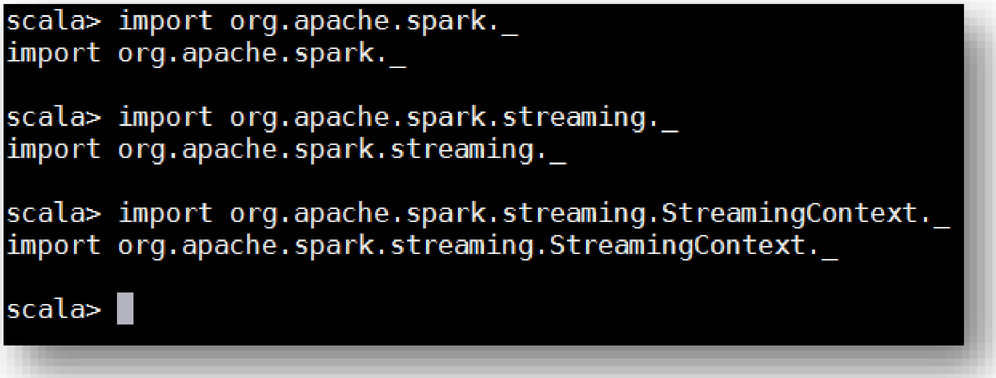
Steps:

1. **We will first import the streaming packages**

import org.apache.spark._

import org.apache.spark.streaming._

import org.apache.spark.streaming.StreamingContext._

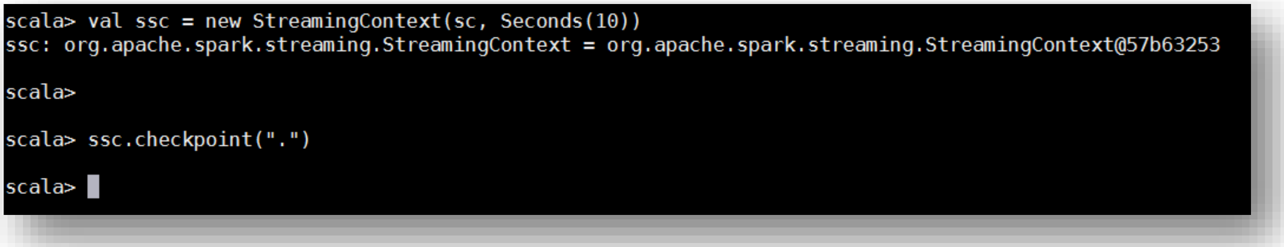


```
scala> import org.apache.spark._
import org.apache.spark._

scala> import org.apache.spark.streaming._
import org.apache.spark.streaming._

scala> import org.apache.spark.streaming.StreamingContext._
import org.apache.spark.streaming.StreamingContext._

scala> 
```

2. **Now, we will create a real-time streaming context with a window of 10 seconds**

val ssc = new StreamingContext(sc, Seconds(10))

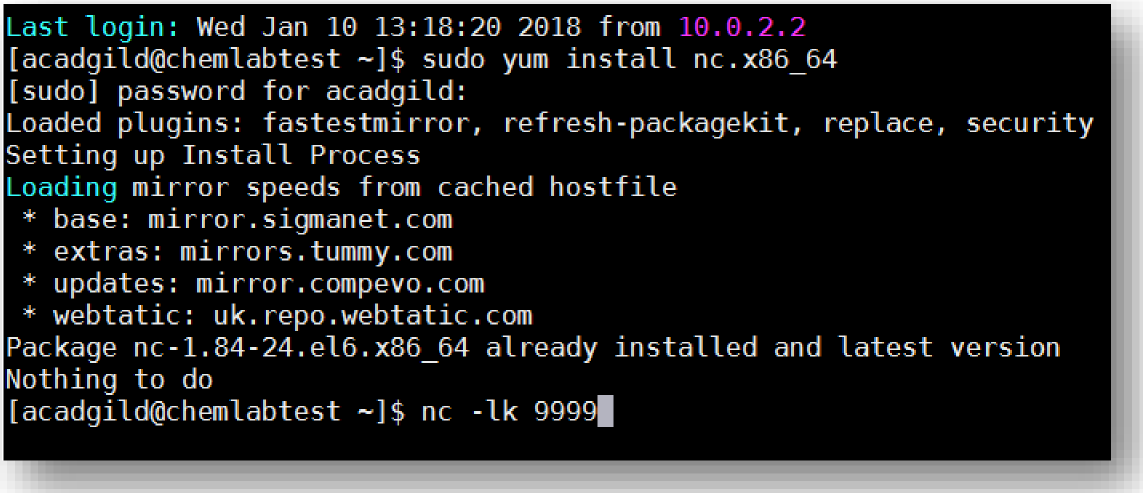
ssc.checkpoint(".")



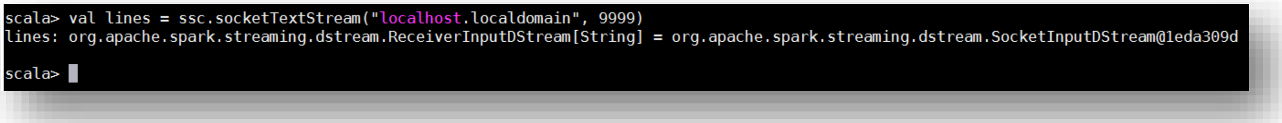
```
scala> val ssc = new StreamingContext(sc, Seconds(10))
ssc: org.apache.spark.streaming.StreamingContext = org.apache.spark.streaming.StreamingContext@57b63253

scala>

scala> ssc.checkpoint(".")

scala> 
```

Now, we will insert lines of offensive word in **nc –lk 9999** after installing **sudo yum install nc.x86_64**
The following command will be used to read those inserted lines

val lines = ssc.socketTextStream("localhost.localdomain", 9999)



```
Last login: Wed Jan 10 13:18:20 2018 from 10.0.2.2
[acadgild@chemlabtest ~]$ sudo yum install nc.x86_64
[sudo] password for acadgild:
Loaded plugins: fastestmirror, refresh-packagekit, replace, security
Setting up Install Process
Loading mirror speeds from cached hostfile
 * base: mirror.sigmanet.com
 * extras: mirrors.tummy.com
 * updates: mirror.compevo.com
 * webtatic: uk.repo.webtatic.com
Package nc-1.84-24.el6.x86_64 already installed and latest version
Nothing to do
[acadgild@chemlabtest ~]$ nc -lk 9999
```



```
scala> val lines = ssc.socketTextStream("localhost.localdomain", 9999)
lines: org.apache.spark.streaming.dstream.ReceiverInputDStream[String] = org.apache.spark.streaming.dstream.SocketInputDStream@1eda309d

scala> 
```

3. Logic to count the words inputted real time:

Then, we will split the words with a space and count the number of times the word has been inserted

val words = lines.flatMap(_.split(" "))

val wordDstream = words.map(word => (word, 1))

The following commands will create an RDD for the words and no. of times the word has been inserted and give us an output.

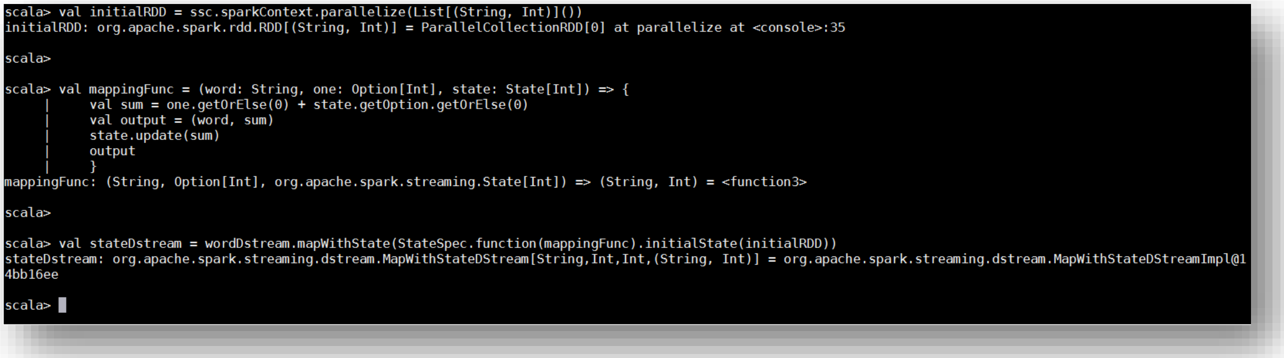
val initialRDD = ssc.sparkContext.parallelize(List[(String, Int)]())

val mappingFunc = (word: String, one: Option[Int], state: State[Int]) => {

   val sum = one.getOrElse(0) + state.getOption.getOrElse(0)

```
   val output = (word, sum)

   state.update(sum)

   output

   }
```

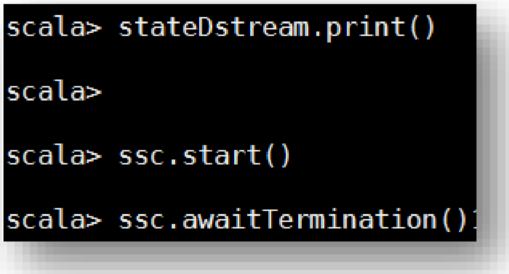
```
val stateDstream =
wordDstream.mapWithState(StateSpec.function(mappingFunc).initialState(initialRDD))
```



```
scala> val initialRDD = ssc.sparkContext.parallelize(List[(String, Int)]())
initialRDD: org.apache.spark.rdd.RDD[(String, Int)] = ParallelCollectionRDD[0] at parallelize at <console>:35

scala>

scala> val mappingFunc = (word: String, one: Option[Int], state: State[Int]) => {
     |     val sum = one.getOrElse(0) + state.getOption.getOrElse(0)
     |     val output = (word, sum)
     |     state.update(sum)
     |     output
     |     }
mappingFunc: (String, Option[Int], org.apache.spark.streaming.State[Int]) => (String, Int) = <function3>

scala>

scala> val stateDstream = wordDstream.mapWithState(StateSpec.function(mappingFunc).initialState(initialRDD))
stateDstream: org.apache.spark.streaming.dstream.MapWithStateDStream[String,Int,Int,(String, Int)] = org.apache.spark.streaming.dstream.MapWithStateDStreamImpl@1
4bb16ee

scala>
```

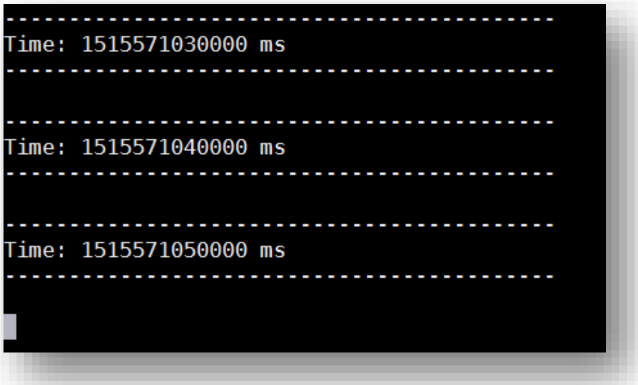
The following commands will start the real-time streaming

```
stateDstream.print()

ssc.start()
ssc.awaitTermination()
```



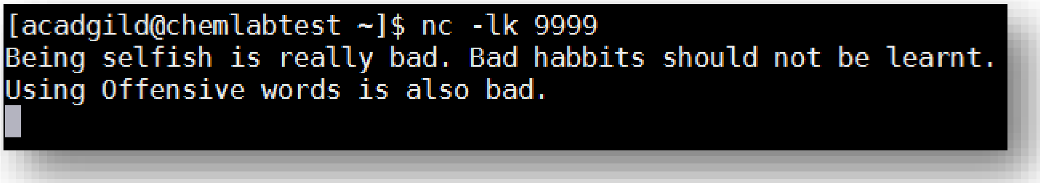
```
scala> stateDstream.print()

scala>

scala> ssc.start()

scala> ssc.awaitTermination()
```

The streaming will look as following:



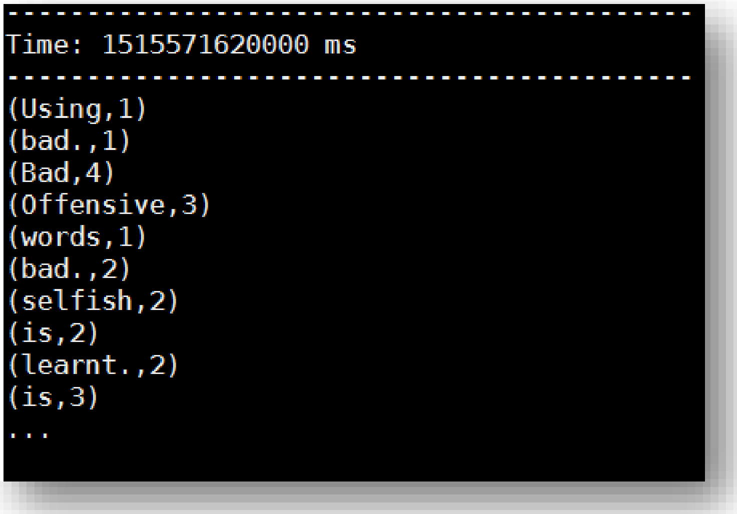
```
-------------------------------------------
Time: 1515571030000 ms
-------------------------------------------

-------------------------------------------
Time: 1515571040000 ms
-------------------------------------------

-------------------------------------------
Time: 1515571050000 ms
-------------------------------------------
```

Following is the sentence inserted taking bad as an offensive word.



```
[acadgild@chemlabtest ~]$ nc -lk 9999
Being selfish is really bad. Bad habbits should not be learnt.
Using Offensive words is also bad.
```

Output:



```
-------------------------------------------
Time: 1515571620000 ms
-------------------------------------------
(Using,1)
(bad.,1)
(Bad,4)
(Offensive,3)
(words,1)
(bad.,2)
(selfish,2)
(is,2)
(learnt.,2)
(is,3)
...
```

As you can see in the output, each word has a count of no. of times the word has been inserted. However the word “Bad” has occurred 4 times.