**Session 20 Assignment 1**

**// Importing APIs and Declaring accumulator**

scala> import org.apache.spark.\_

import org.apache.spark.\_

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

import org.apache.spark.streaming.\_

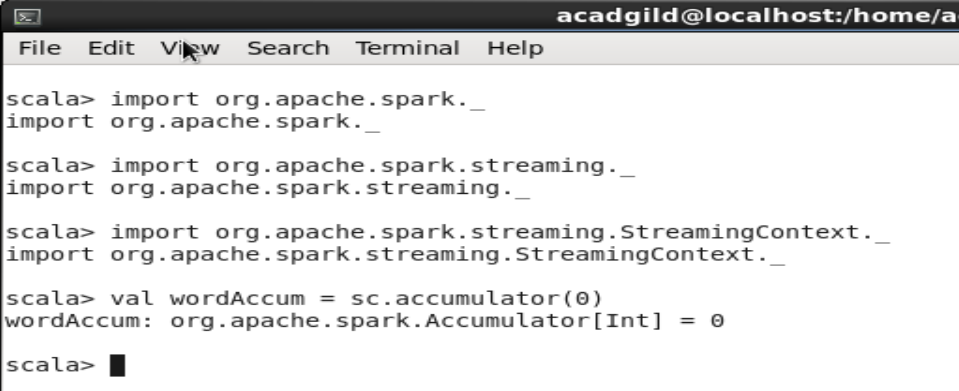
scala> import org.apache.spark.streaming.StreamingContext.\_

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

scala> val wordAccum = sc.accumulator(0)

wordAccum: org.apache.spark.Accumulator[Int] = 0

scala>



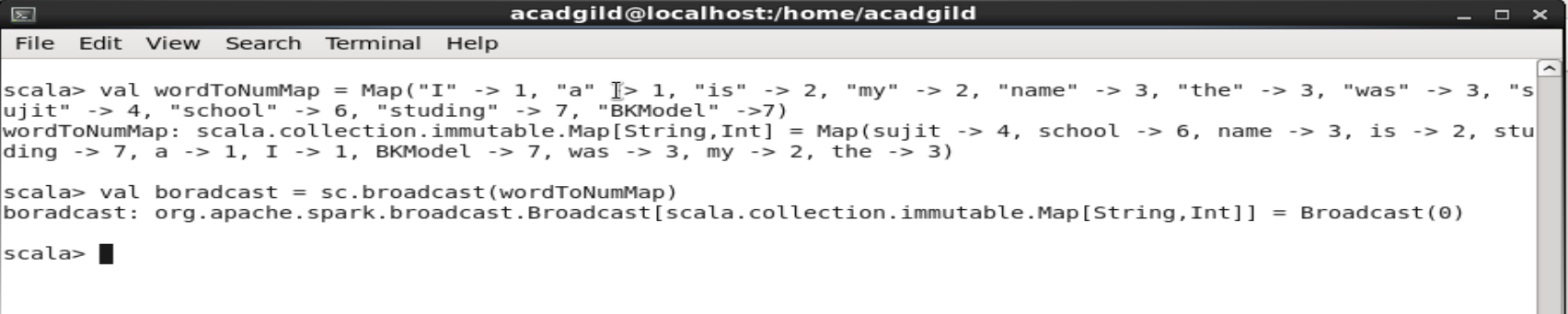
**// Mapping words to numbers and Broadcasting the mapping**

scala> val wordToNumMap = Map("I" -> 1, "a" -> 1, "is" -> 2, "my" -> 2, "name" -> 3, "the" -> 3, "was" -> 3, "sujit" -> 4, "school" -> 6, "studing" -> 7, "BKModel" ->7)

wordToNumMap: scala.collection.immutable.Map[String,Int] = Map(sujit -> 4, school -> 6, name -> 3, is -> 2, studing -> 7, a -> 1, I -> 1, BKModel -> 7, was -> 3, my -> 2, the -> 3)

scala> val boradcast = sc.broadcast(wordToNumMap)

boradcast: org.apache.spark.broadcast.Broadcast[scala.collection.immutable.Map[String,Int]] = Broadcast(0)



**// Defining function lineWordNumSum**

scala> def lineWordNumSum(line:String):Int = {

| var sum:Int= 0

| var words = line.split(" ")

| for (word <- words)

| {

| sum += boradcast.value.get(word).getOrElse(0)

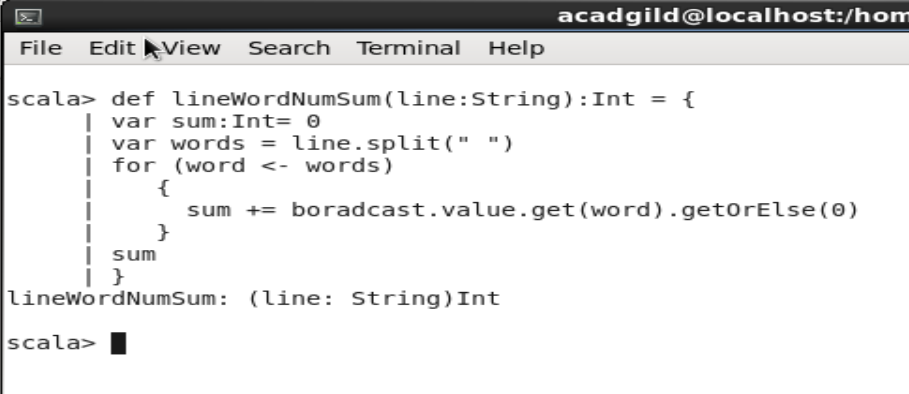
| }

| sum

| }

lineWordNumSum: (line: String)Int

scala>



**// Settingup Streaming Context and assigning local port**

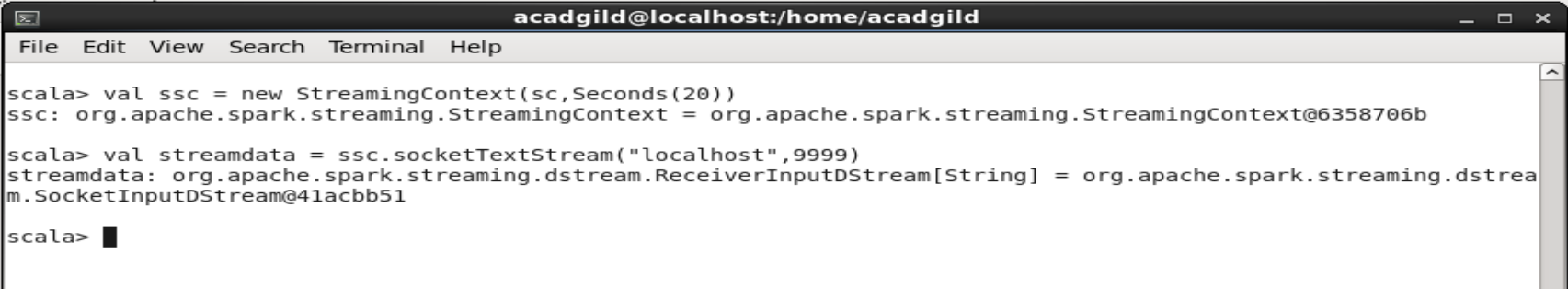
scala> val ssc = new StreamingContext(sc,Seconds(20))

ssc: org.apache.spark.streaming.StreamingContext = org.apache.spark.streaming.StreamingContext@6358706b

scala> val streamdata = ssc.socketTextStream("localhost",9999)

streamdata: org.apache.spark.streaming.dstream.ReceiverInputDStream[String] = org.apache.spark.streaming.dstream.SocketInputDStream@41acbb51

scala>



**// Processing each RDD and calculating sum for each line and printing both Even sum for a line and excluded odd sum.**

scala> streamdata.foreachRDD(line => {

| val lineStr = line.collect().toList.mkString("")

| if (lineStr != "")

| {

| var lineTotal = lineWordNumSum(lineStr)

| if (lineTotal % 2 == 0)

| {

| wordAccum += lineTotal

| println("Sum of all the words in lines with Even line total is : " + wordAccum.value.toInt)

| }

| else

| {

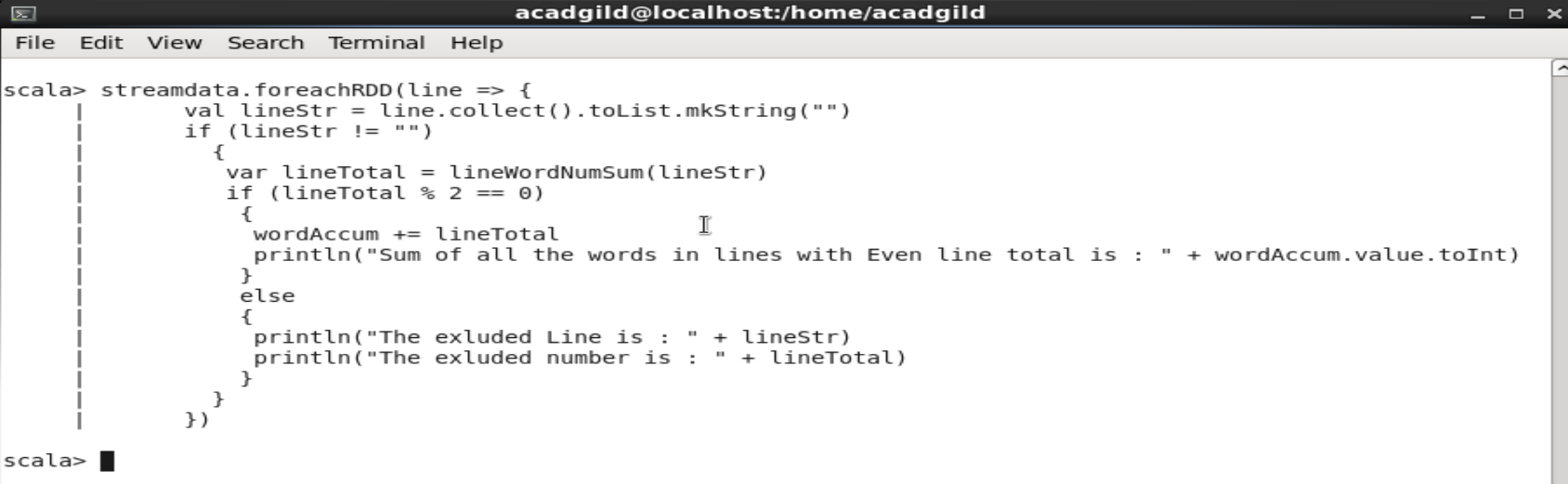
| println("The exluded Line is : " + lineStr)

| println("The exluded number is : " + lineTotal)

| }

| }

| })



**// Starting spark streaming**

scala> ssc.start()

**// In new terminal opening server for 9999 to key in the words**

scala> nc -lk 9999

