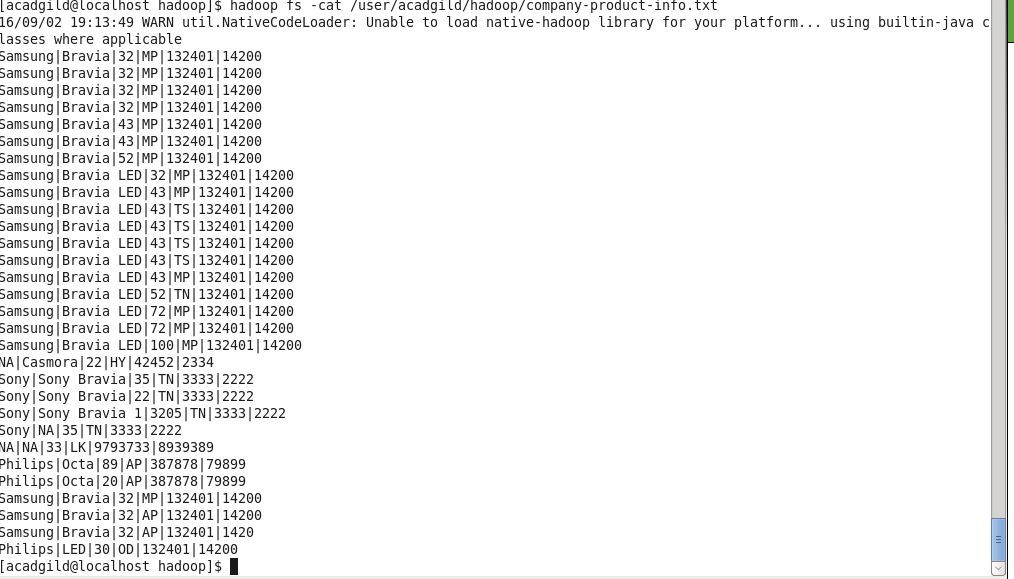
**Input:**

Company Name|Product Name|Size in inches|State|Pin Code|Price



**Task8:** Write a Map Reduce program that takes the output of Task 5 (refer session 5, assignment 1) as input, and produce output which is sorted on the total units sold.

You may use a single reducer for the sorting.

Use Sequence File formats as an output for Task 5 and as input for this task..

I have created 2 MR programs

  a) One to count number of units sold per company/producy/size and output file format would be (company/producy/size, number\_of\_units\_sold)

  b) another MR to take first MR output and parse in MAP program by seperater ',' and generate ouput to REDUCE program as (number\_of\_units\_sold, company/producy/size)

  As sort(by default in ascending order) works on key, map output will be sorted and passed to reducer.

Package: acadgild.session6.task8

**First MR :**

Driver Class: DriverProg.java

Mapper Class: MapperProg.java

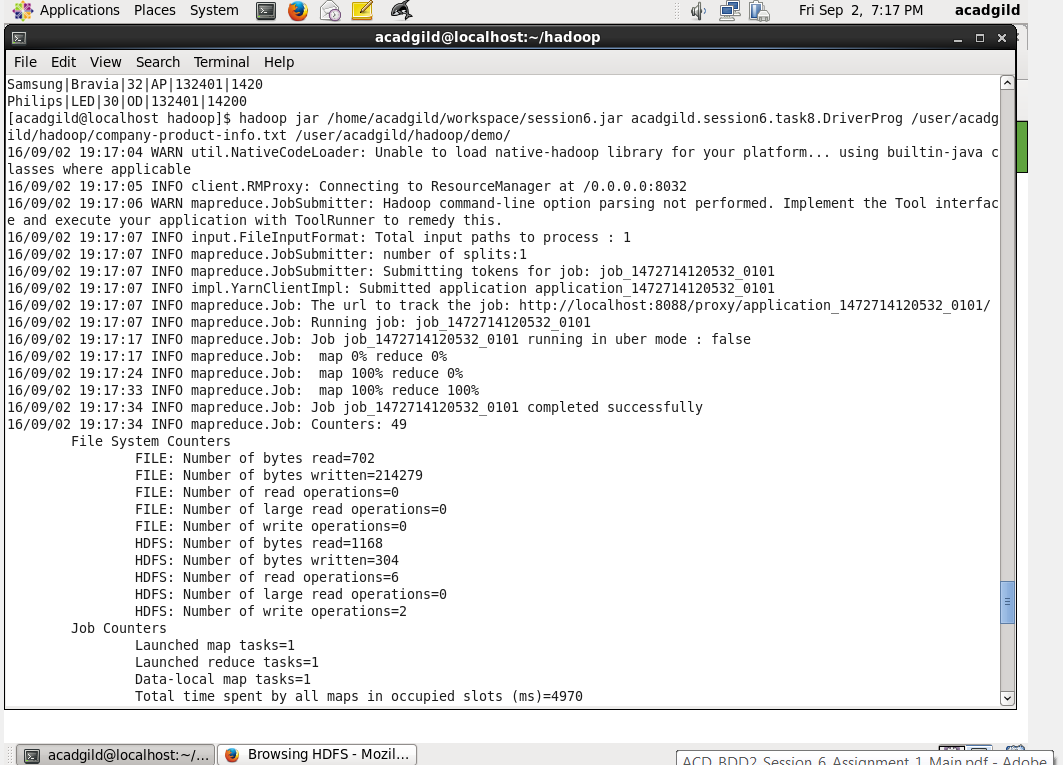
Reducer Claass: ReducerProg.java

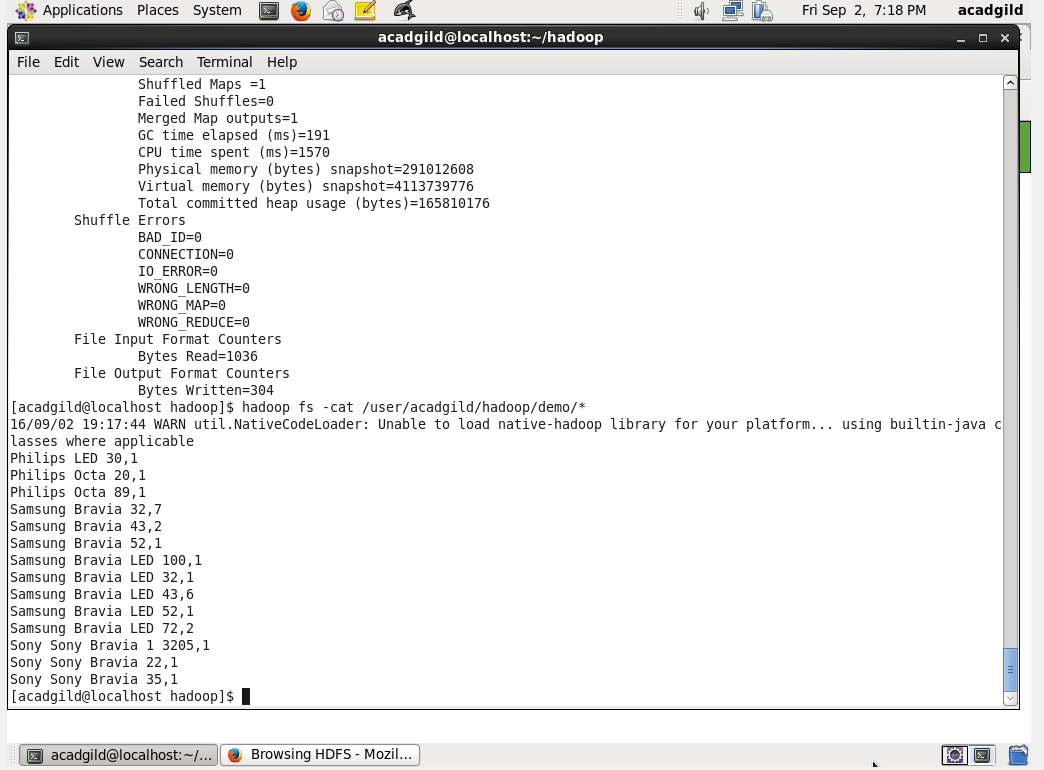
**Second MR:**

SortedProg.java --- which will have all in line classes for Driver, Mapper and Reducer

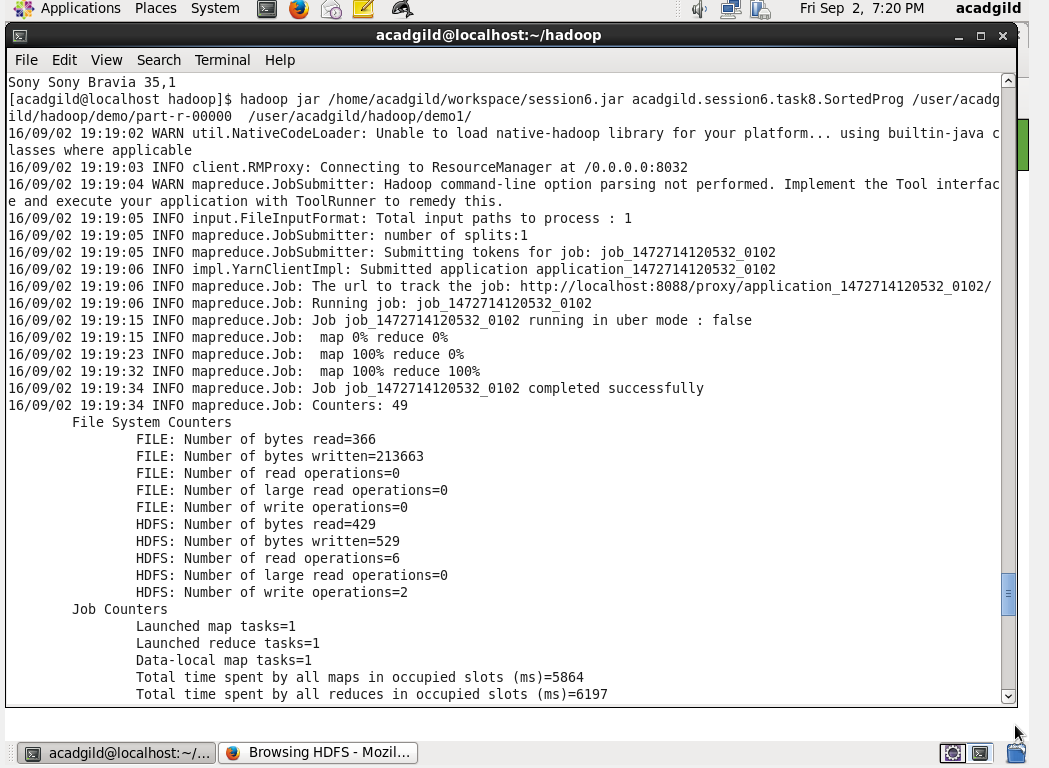
**Output:**

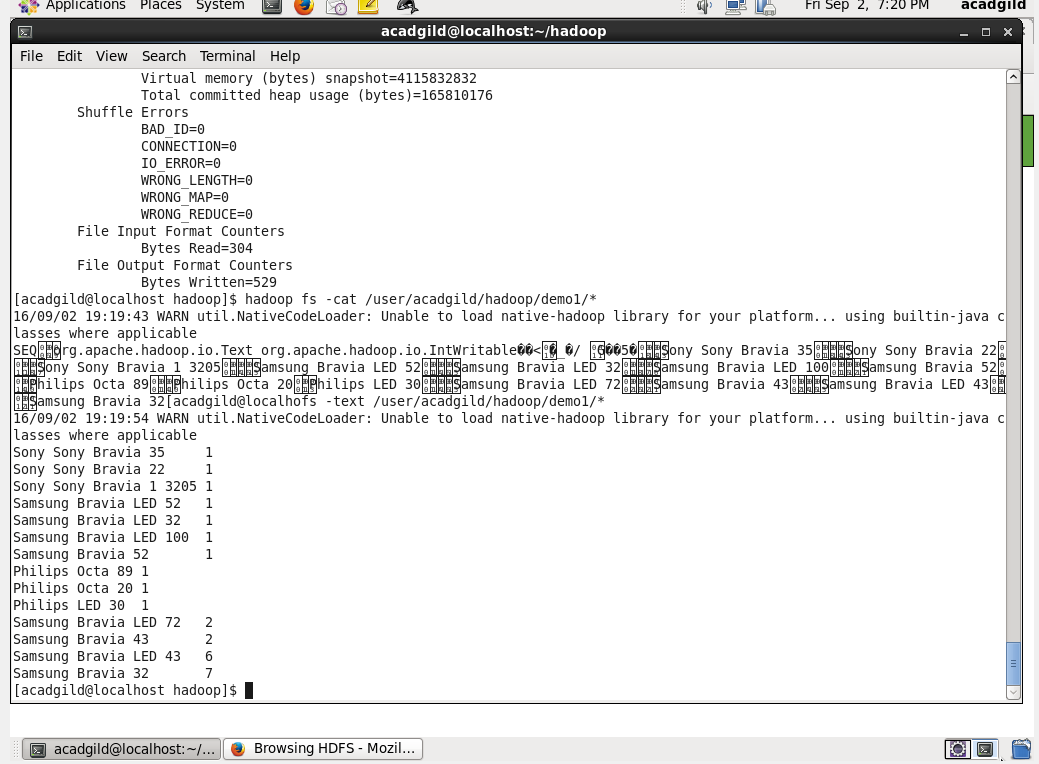
First MR output:





Second MR Output:





**Task9:** Enhance the Map Reduce program of Task 8 (refer session 6, assignment 1) to use multiple reducers for sorting.

The driver should accept three additional values: the minimum units sold, the maximum units sold and number of reducers to use.

Use units sold as key and company as value.

Write a custom partitioner to divide the keys on the basis of range.

Take minimum to be 0 and maximum to be 10. Divide them across 2 reducers..

I have created 2 MR programs

  a) One to count number of units sold per company/producy/size and output file format would be (company/producy/size, number\_of\_units\_sold)

  b) another MR to take first MR output and parse in MAP program by seperater ',' and partition the half of mapper output to one reducer and rest to another.

Package: acadgild.session6.task8

**First MR :**

Driver Class: DriverProg.java

Mapper Class: MapperProg.java

Reducer Claass: ReducerProg.java

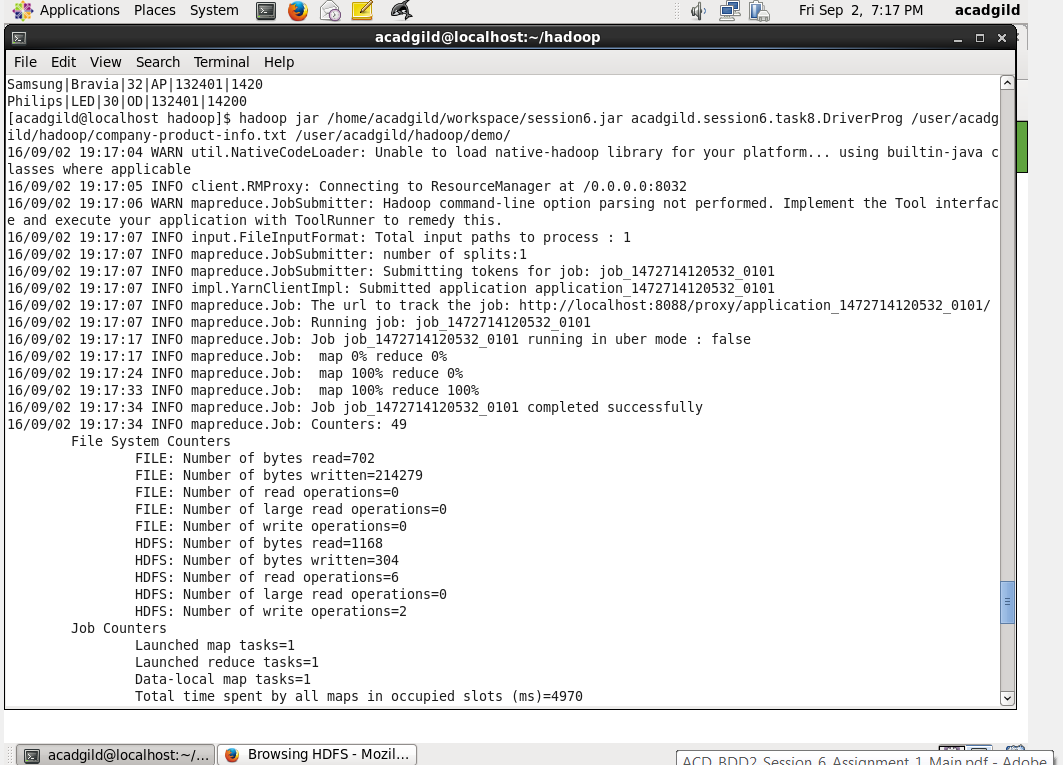
**Second MR:**

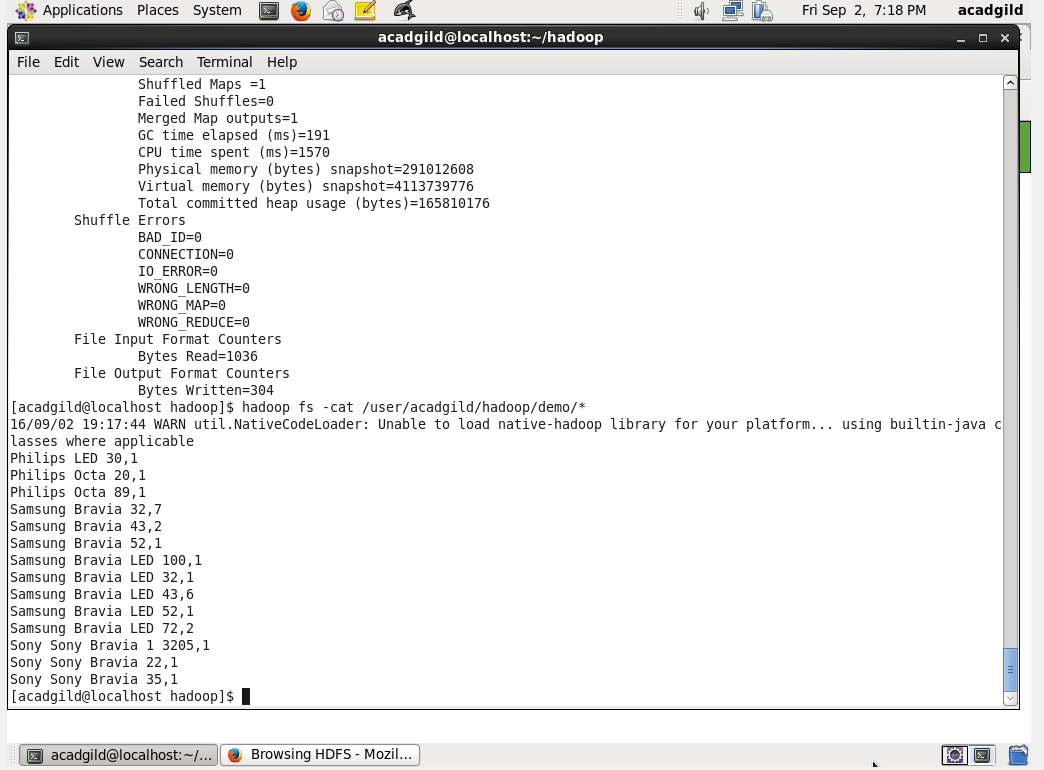
SortedProg.java --- which will have all in line classes for Driver, Mapper and Reducer

Partioner Class : PartitionerProg1.java

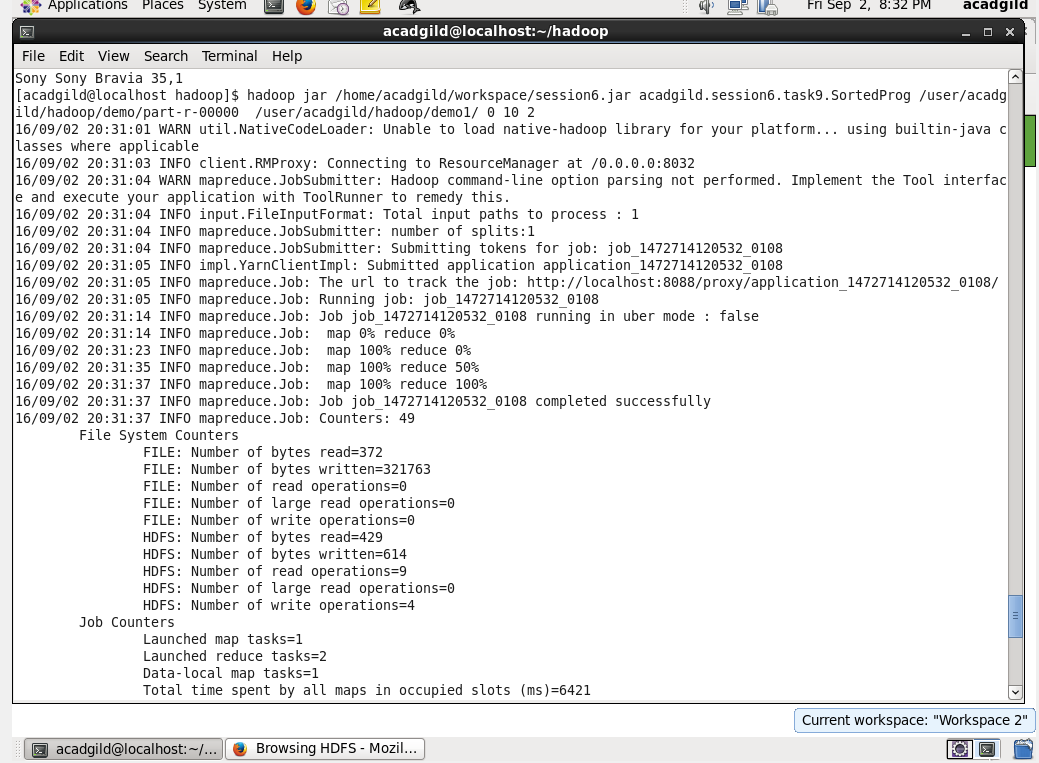
**Output:**

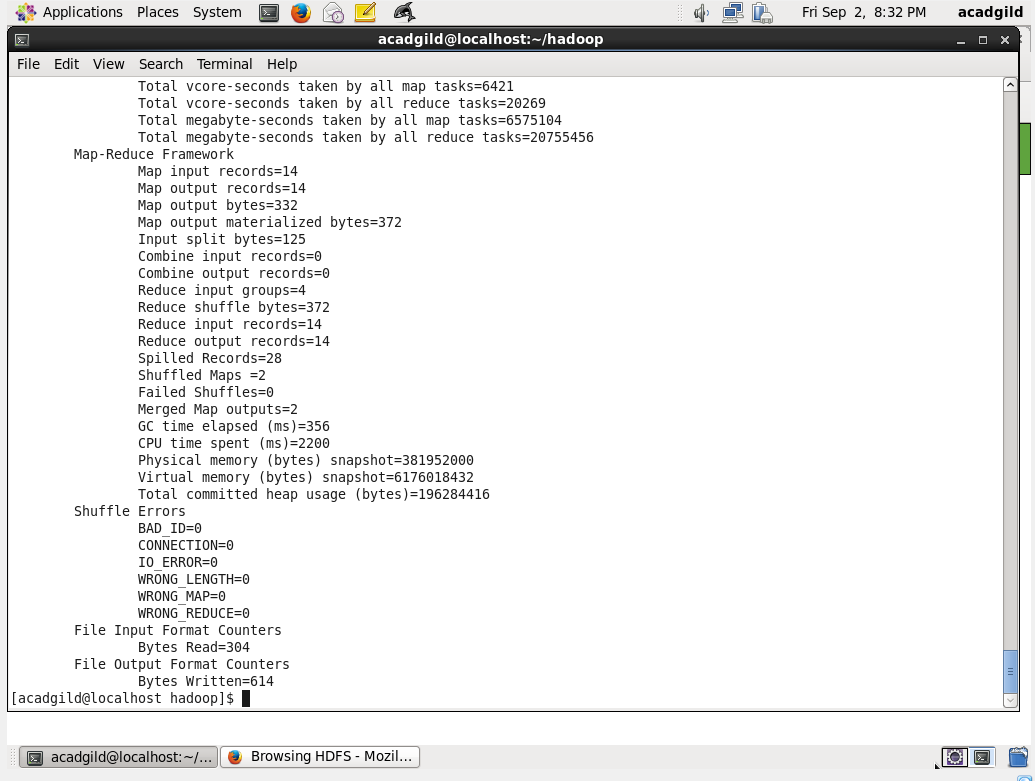
First MR output:

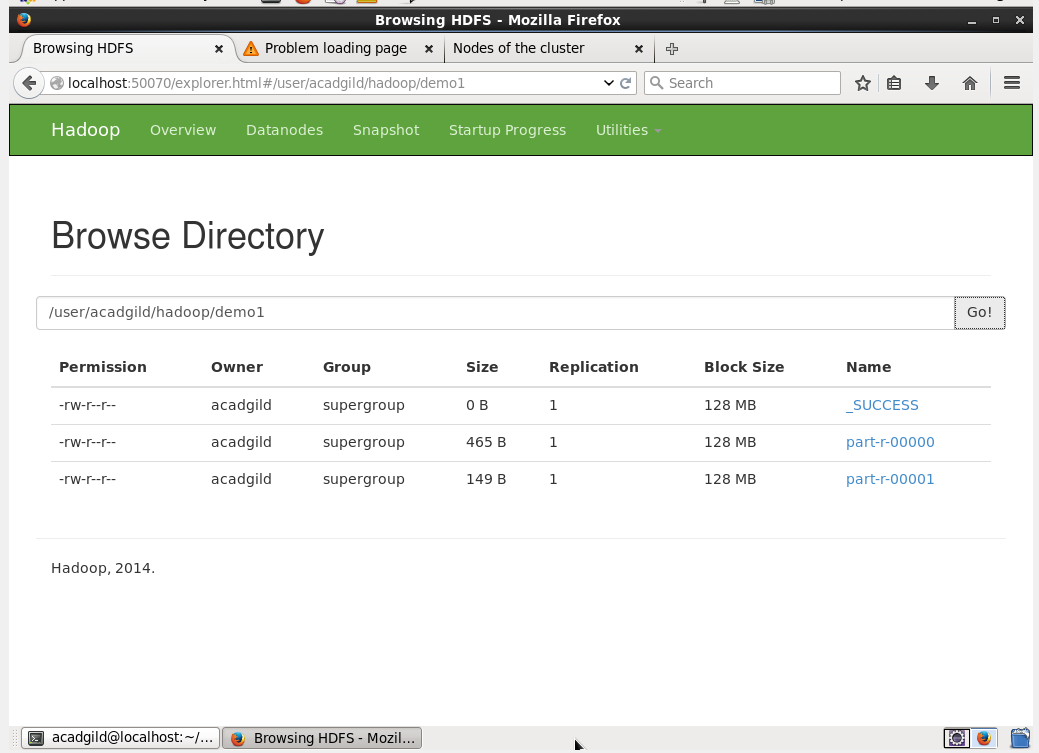


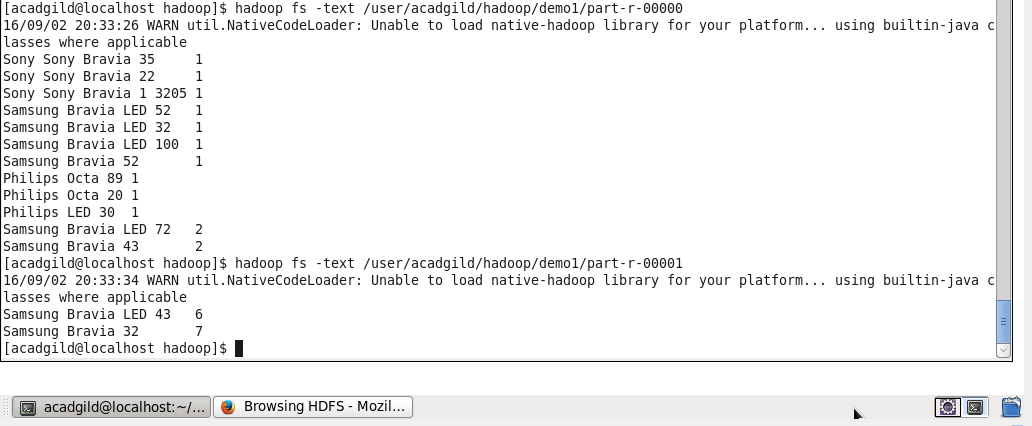


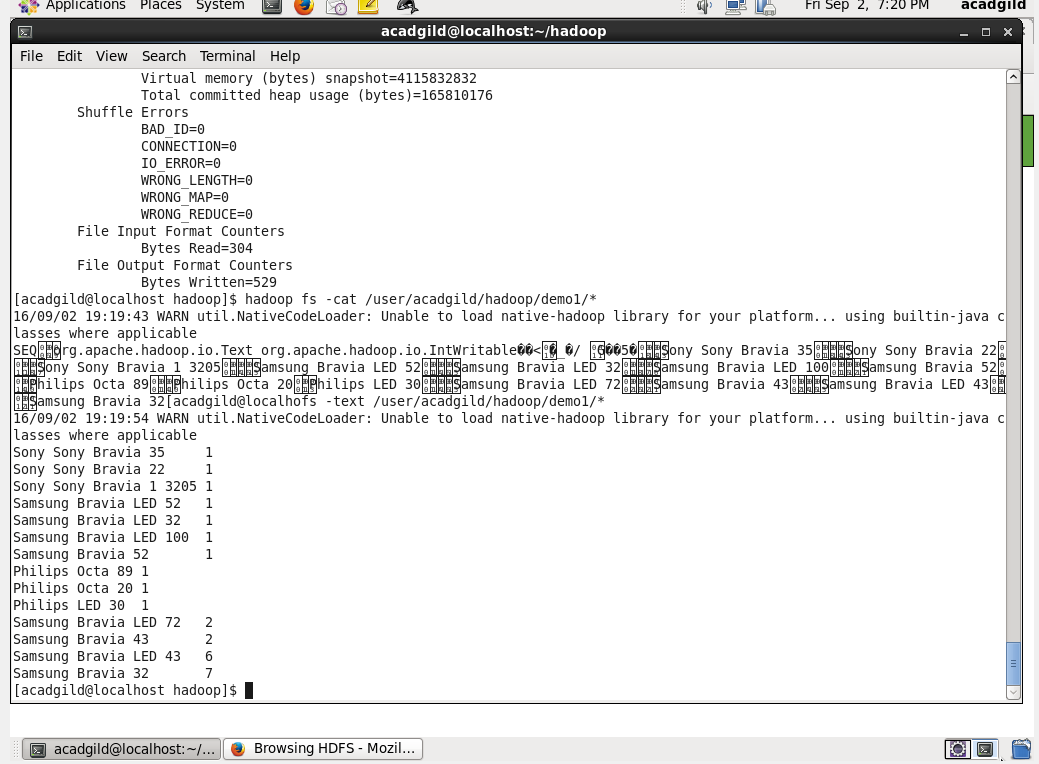
Second MR Output:











**Task10:** Enhance the task 8 (refer session 6, assignment 1) to calculate the top 3 state-wise sales for each company.

You may use multiple reducers for this activity.

I have created 2 MR programs

  a) One to count number of units sold per company/producy/size and output file format would be (company/state number\_of\_units\_sold)

  b) another MR to take first MR output and parse in MAP program by seperater '\t' and generate ouput to REDUCE program as (companyname, numberofunitssold)

Package: acadgild.session6.task10

**First MR :**

Driver Class: DriverProg.java

Mapper Class: MapperProg.java

Reducer Claass: ReducerProg.java

**Second MR:**

Driver Class: SortedProg.java

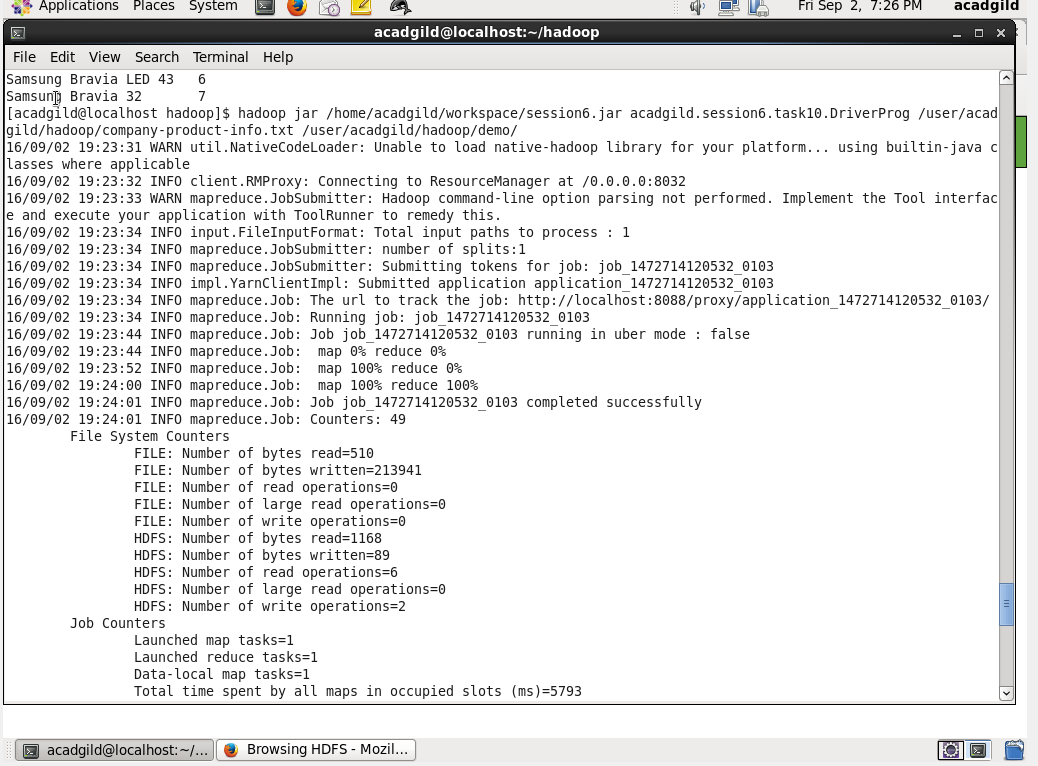
Mapper Class: MapperProg1.java

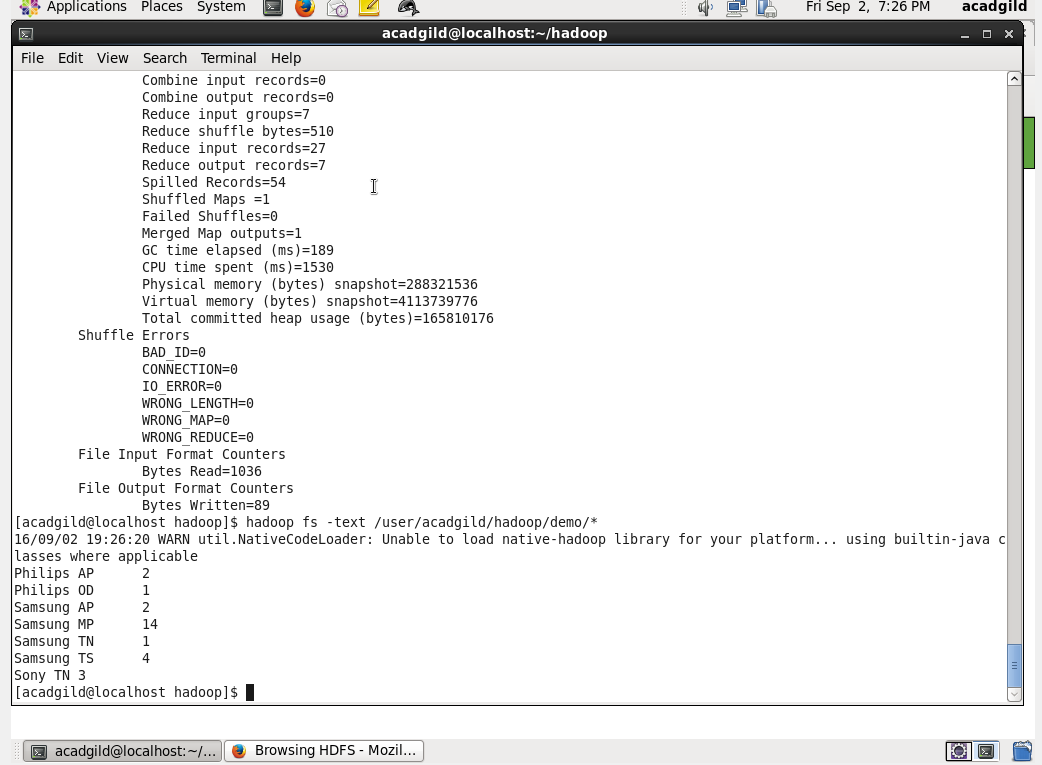
Reducer Class : ReducerProg1.java

Partitioner Class : PartitionerProg1.java

**Output:**

First MR output:





Second MR Output:

