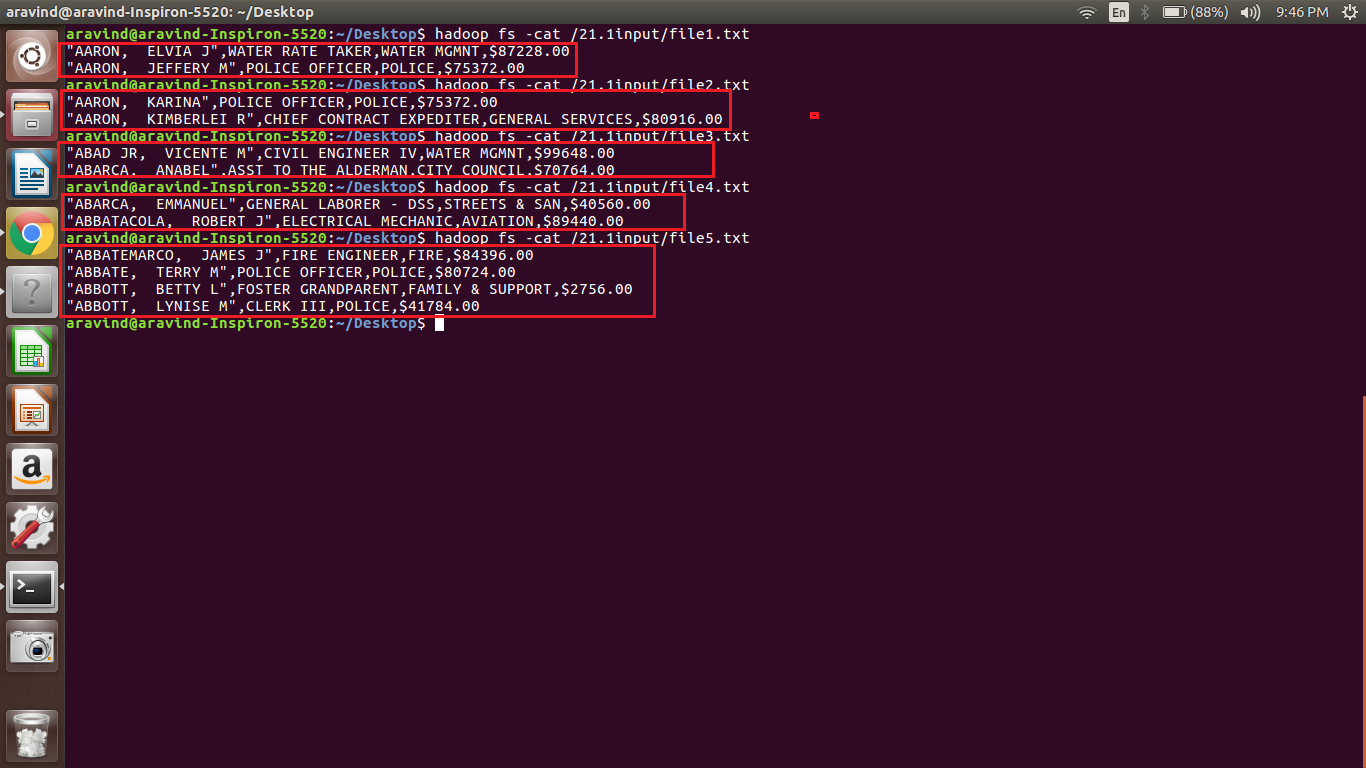
**21.1 Write a mapreduce program to use CombineFileInputFormat. Pass a directory containing multiple small files. The program should be able to pick multiple files in a single mapper.**

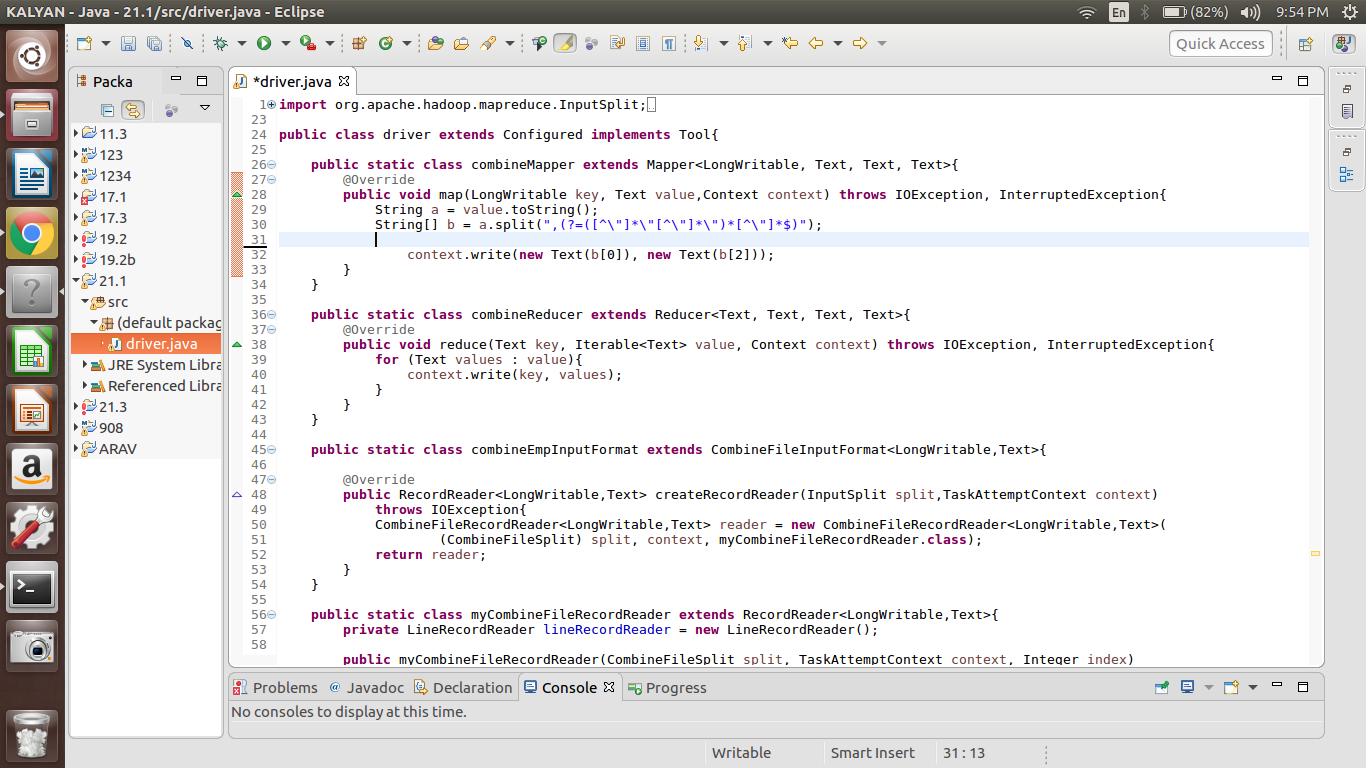
**LOGIC:**

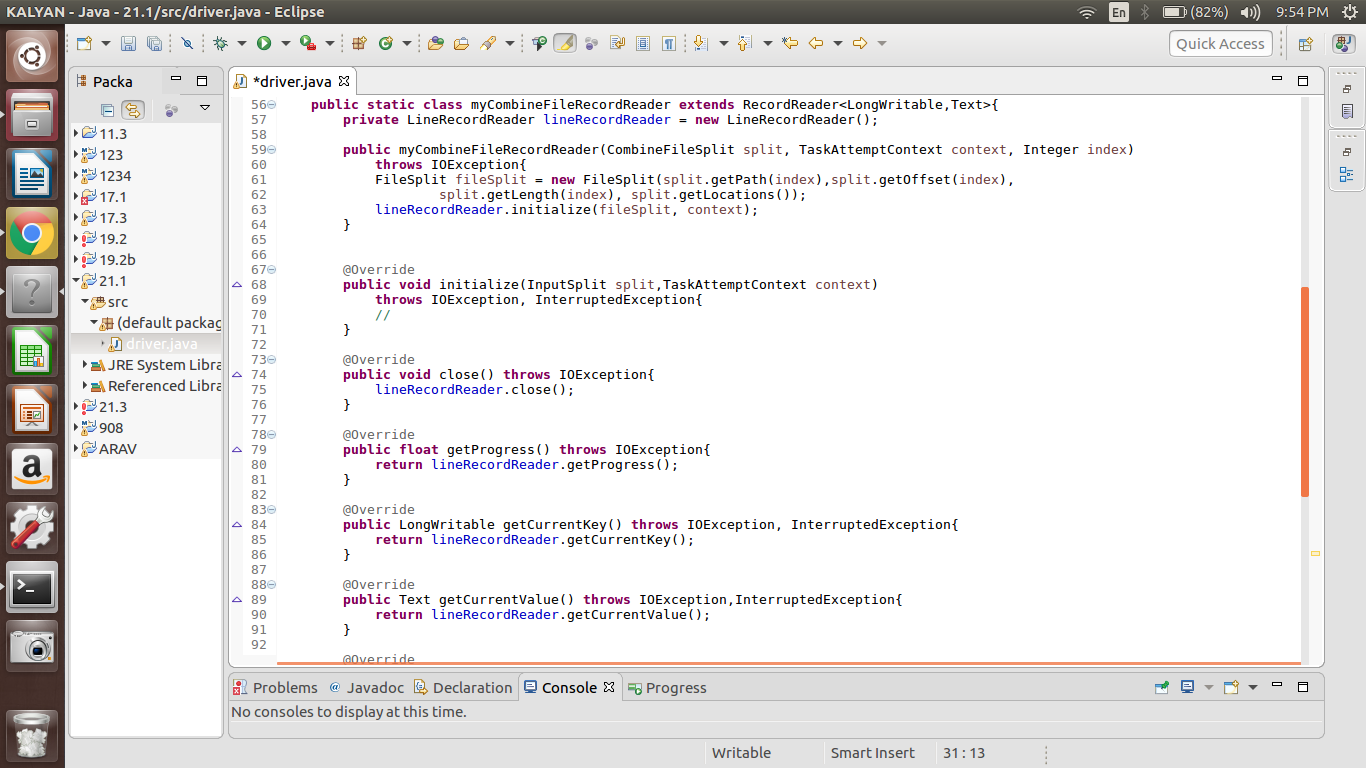
* **I’ve included all the classes in a single class itself.**
* **Driver class consists of a main class, mapper class, reducer class, combinefileinputformat class, record reader class.**
* **In main class, I’ve set all the classes and defined job and configurations.**
* **In mapper class, I’ve splitted the dataset using regex and set the name and department as key and value of mapper output.**
* **CombineFileInputFormat is an abstract class without any concrete classes, unlike FileInputFormat.**
* **To have CombineFileInputFormat equivalent of TextInputFormat, we need to create a concrete subclass of CombineFileInputFormat and implement the getRecordReader() method.**
* **CombineFileInputFormat combines many small files into a split, providing more data for each map task. It also considers data locality so performance is not compromised.**
* **In the mapreduce code I have set mapred.max.split.size. This determines the size of Input split that a mapper has to handle.**
* **While running the jar file, instead of passing the file as input pass the directory (containing small files) as argument.**

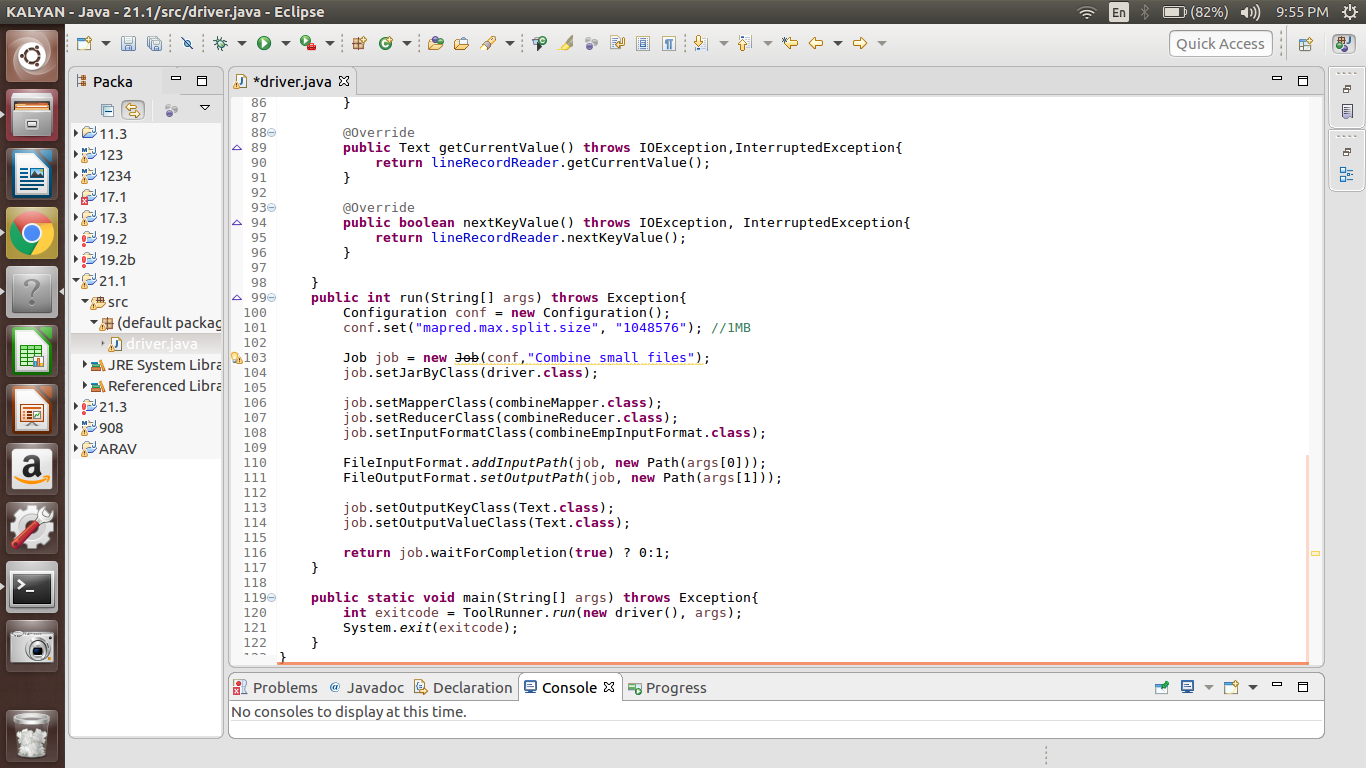
**INPUT FILE:**

****

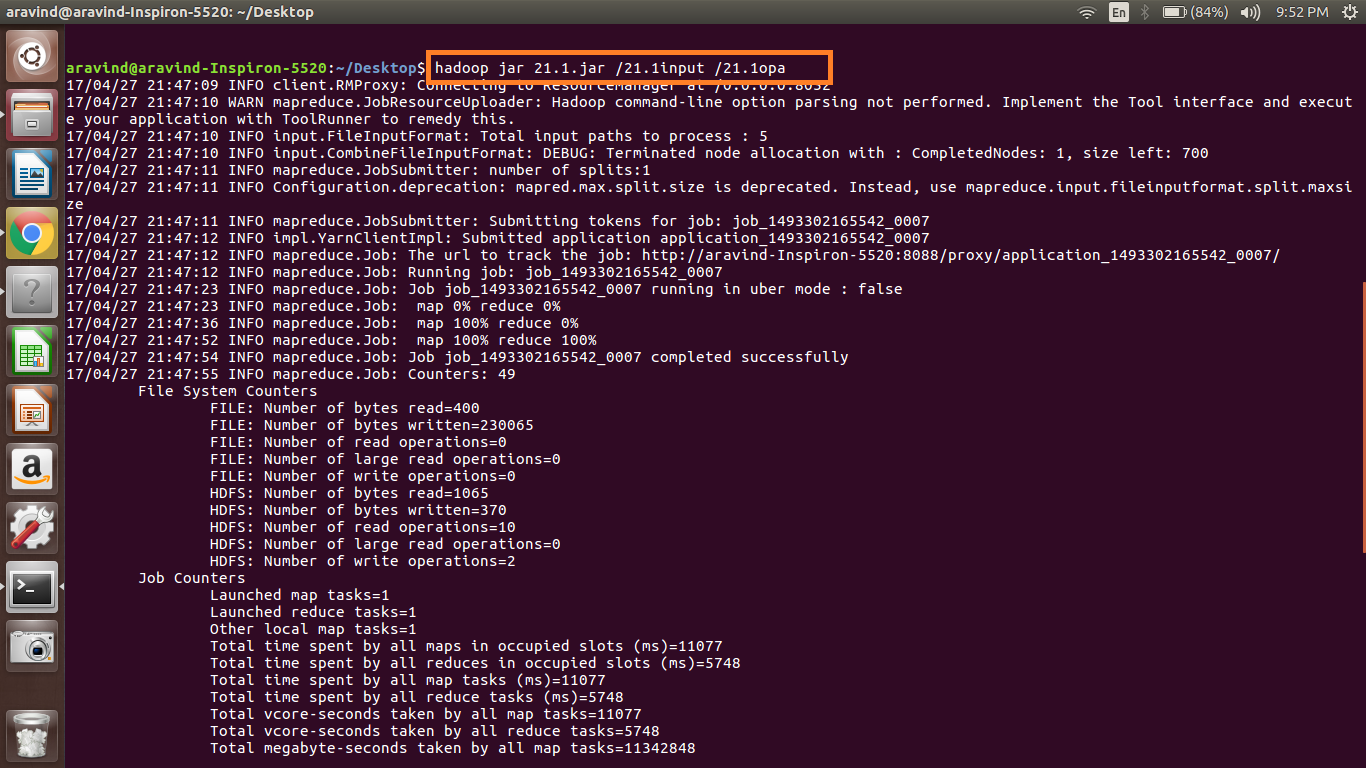
**DRIVER CLASS:**

****

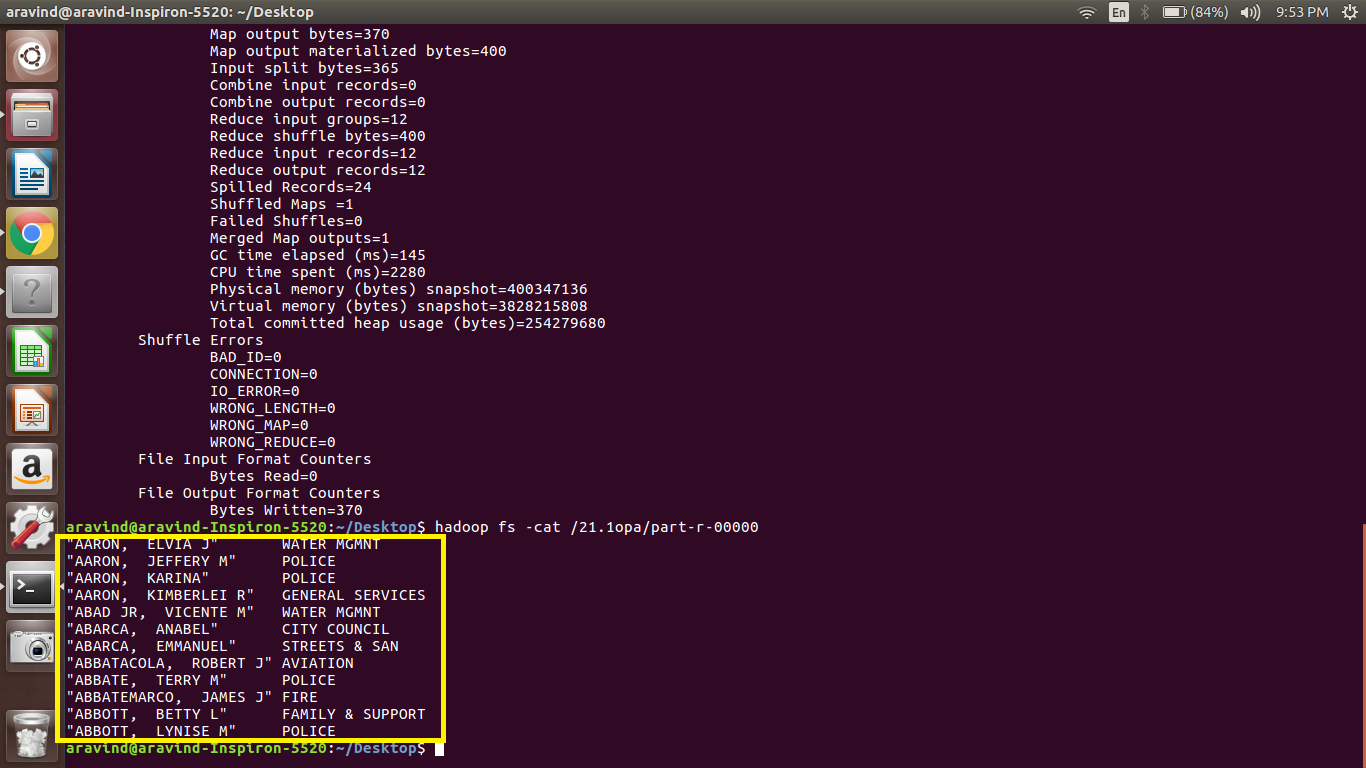
****

****

**RUNNING JAR:**

****

**OUTPUT:**

****