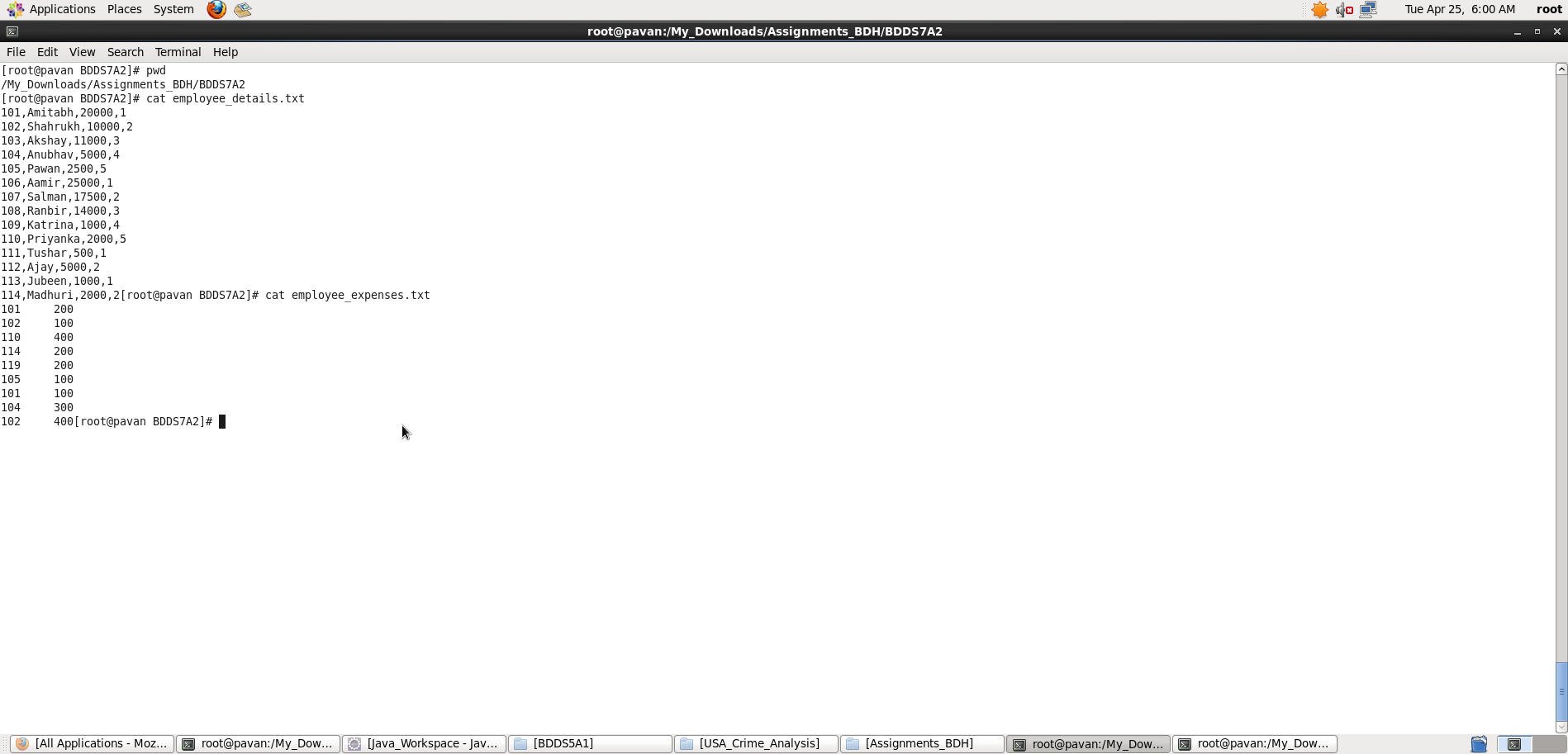
# Big Data Hadoop Training

Session 7 Assignment 2 Solution

**We have employee\_details and employee\_expenses files. Use local mode while running Pig and write Pig Latin script to get below results:**

**Input files stored in HDFS:**

****

**(a) Top 5 employees (employee id and employee name) with highest rating.**

**(In case two employees have same rating, employee with name coming first in dictionary should get preference)**

A) **Pig Latin Script**:

emp\_details = load './employee\_details.txt' USING PigStorage(',') AS (empId:int, empName:chararray, salary:int, rating:int);

grpall = GROUP emp\_details ALL;

out = FOREACH grpall {

sort = ORDER emp\_details BY rating DESC, empName ASC;

top = LIMIT sort 5;

GENERATE FLATTEN(top);

}

top5empId\_empName = FOREACH out GENERATE empId,empName;

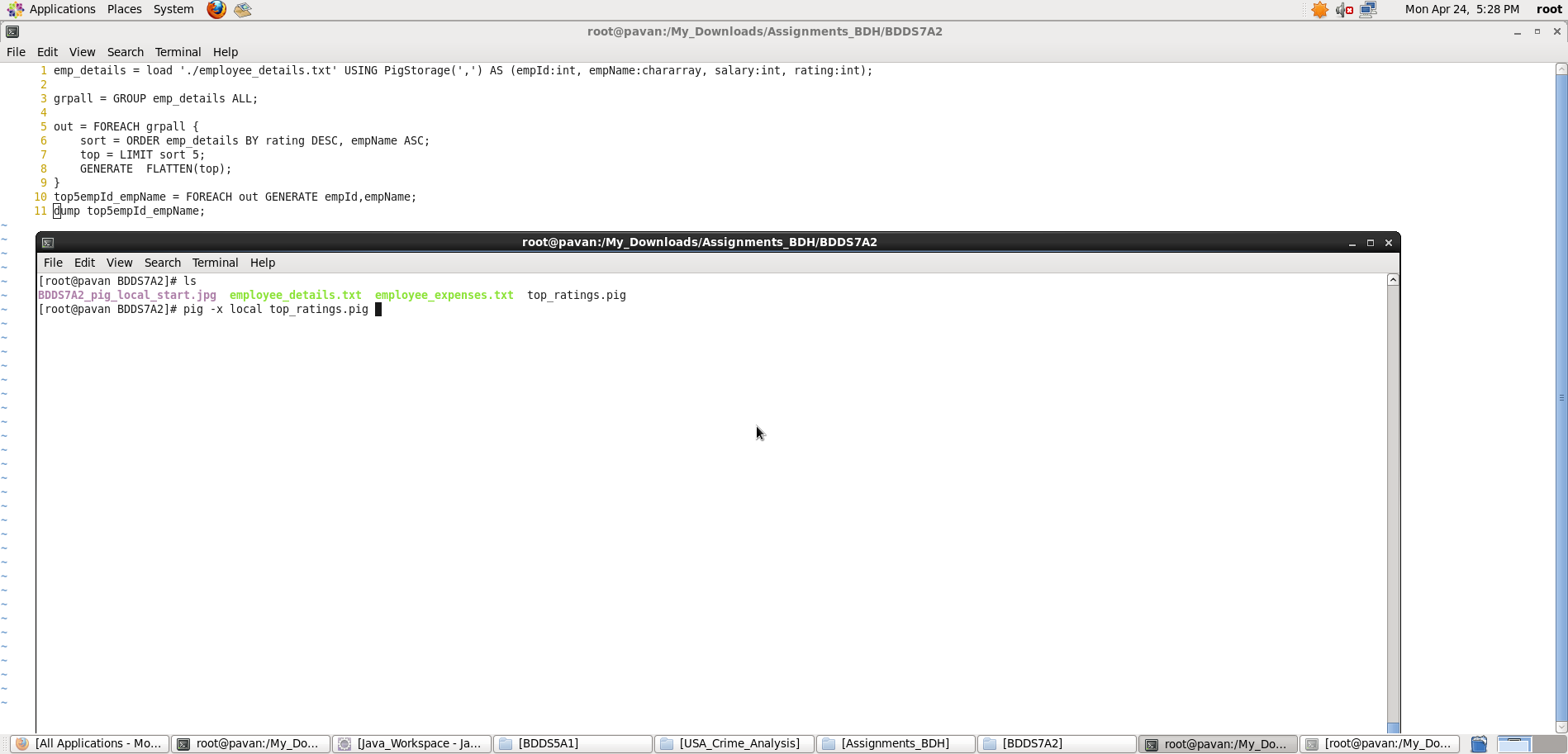
dump top5empId\_empName;

**Explanation :**

* Loaded data from employee\_details.txt into “emp\_details” relation. Here ‘,’ is the delimiter and schema is provided as mentioned in the question
* Then, all records are grouped together
* Now, iterating over tuples inside the group I have sorted the emp\_details by rating in descending order and then those with same rating are sorted in ascending order according to the employee with name coming first in dictionary.
* Limiting my output to top 5 records
* Then flattening the tuples inside group
* **Outputting only “the employee id and employee name” with top rating in descending order as per Question using dump command.**

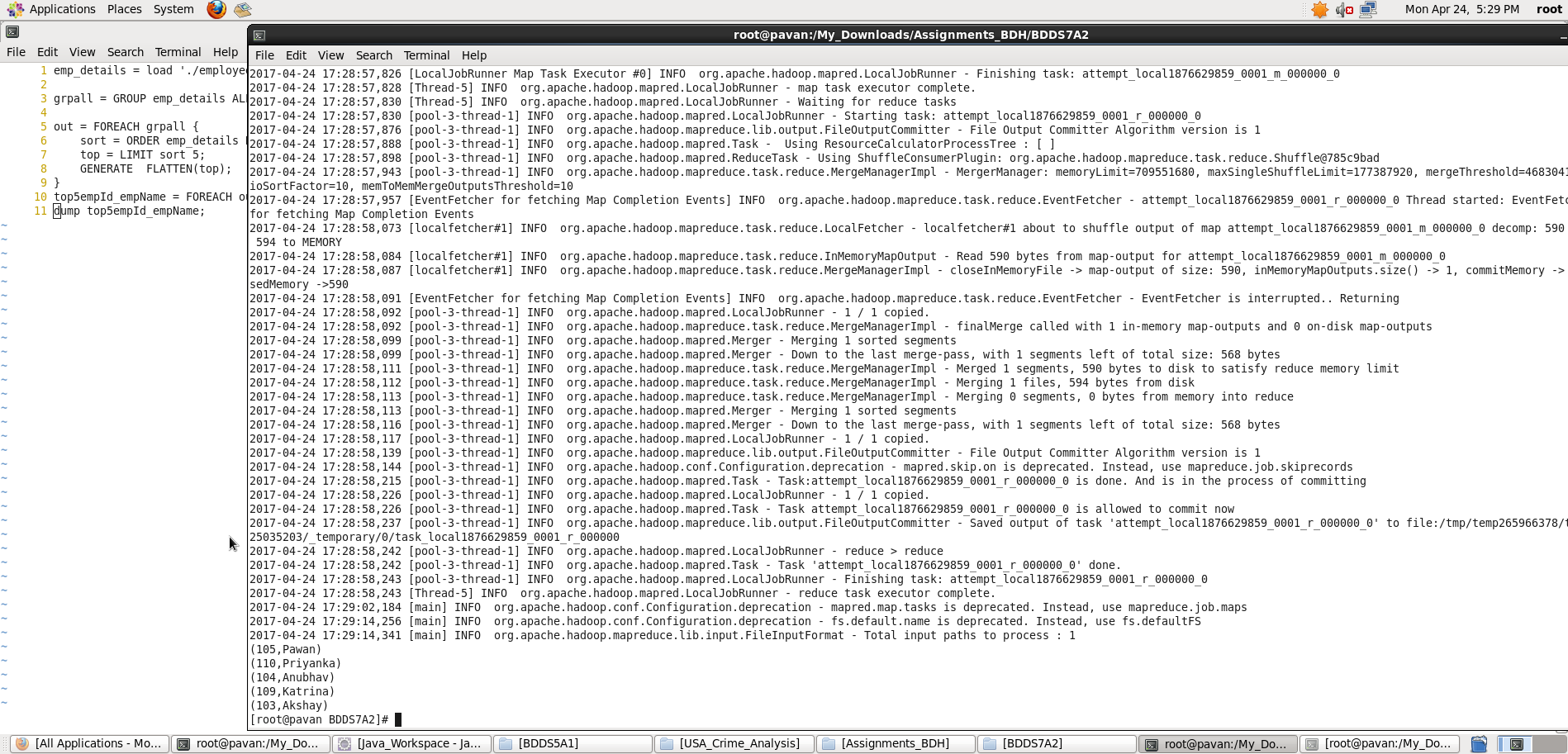
Screenshots

Running **Pig Latin Script in Local Mode** inside /My\_Downloads/Assignments\_BDH/BDDS7A2/ directory



Output displayed: **<employee\_id> <employee\_name>**

**Note : We can store the output if required in a file using “store” command.**



**b) Top 3 employees (employee id and employee name) with highest salary, whose employee id is an odd number.**

**(In case two employees have same salary, employee with name coming first in dictionary should get preference)**

1. **Pig Latin Script:**

emp\_details = load './employee\_details.txt' USING PigStorage(',') AS (empId:int, empName:chararray, salary:int, rating:int);

grpall = GROUP emp\_details ALL;

out = FOREACH grpall {

sort = ORDER emp\_details BY salary DESC, empName ASC;

Id\_odd = FILTER sort BY (empId%2)==1;

top3 = LIMIT Id\_odd 3;

GENERATE FLATTEN(top3);

}

top3empId\_empName = FOREACH out GENERATE empId,empName;

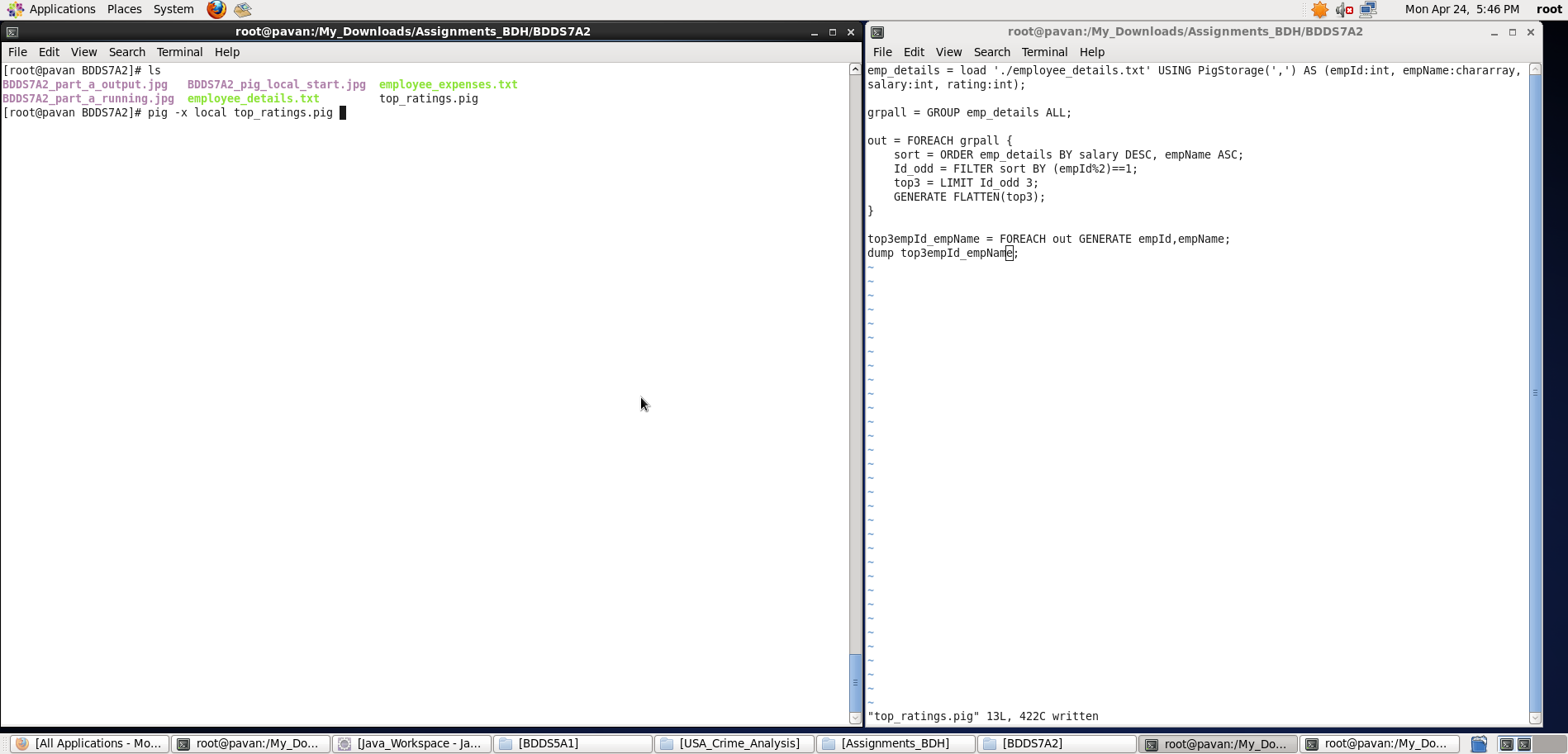
dump top3empId\_empName;

**Explanation :**

* Loaded data from employee\_details.txt into “emp\_details” relation. Here ‘,’ is the delimiter and schema is provided as mentioned in the question
* Then, all records are grouped together
* Now, iterating over tuples inside the group I have sorted the emp\_details by salary in descending order and then those with same salary are sorted in ascending order according to the employee with name coming first in dictionary.
* Then, filtered the list such that empId is odd.
* Limiting my output to top 3 records
* Then flattening the tuples inside group
* **Outputting only “the employee id and employee name” with highest salary in descending order as per Question using dump command.**

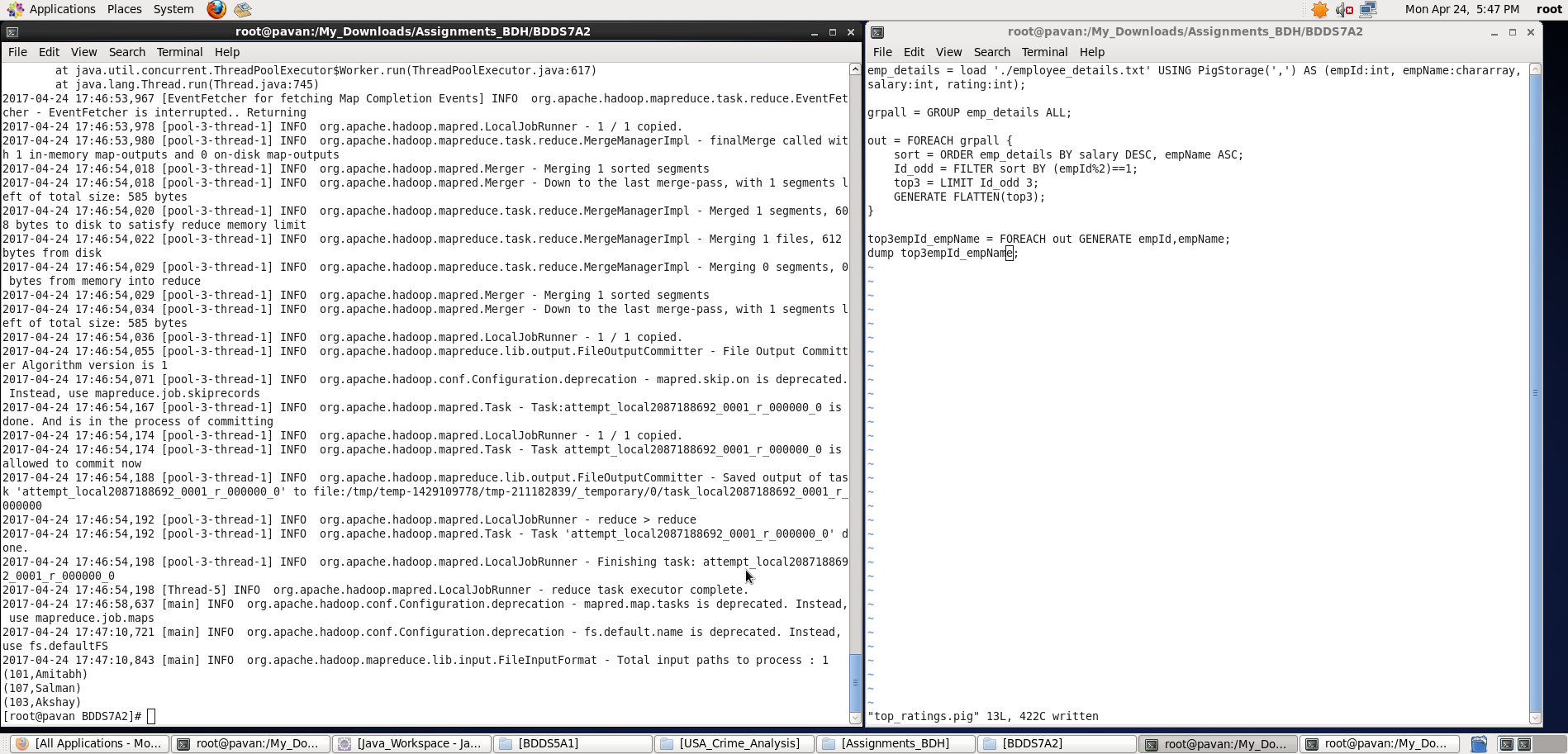
Screenshots

Running **Pig Latin Script in Local Mode** inside /My\_Downloads/Assignments\_BDH/BDDS7A2/ directory



Output displayed: **<employee\_id> <employee\_name>**

**Note : We can store the output if required in a file using “store” command.**



c) **Employee (employee id and employee name) with maximum expense**

**(In case two employees have same expense, employee with name coming first in dictionary should get preference)**

**Pig Latin Script:**

emp\_details = load './employee\_details.txt' USING PigStorage(',') AS (empId:int, empName:chararray, salary:int, rating:int);

emp\_expenses = load './employee\_expenses.txt' USING PigStorage() AS (empId:int, expenses:int);

emp\_joined\_data = JOIN emp\_details BY empId, emp\_expenses BY empId;

grpall = GROUP emp\_joined\_data ALL;

out = FOREACH grpall {

sort = ORDER emp\_joined\_data BY emp\_expenses::expenses DESC, emp\_details::empName ASC;

topexp = LIMIT sort 1;

GENERATE FLATTEN(topexp);

}

topexpempId\_empName = FOREACH out GENERATE emp\_details::empId,emp\_details::empName;

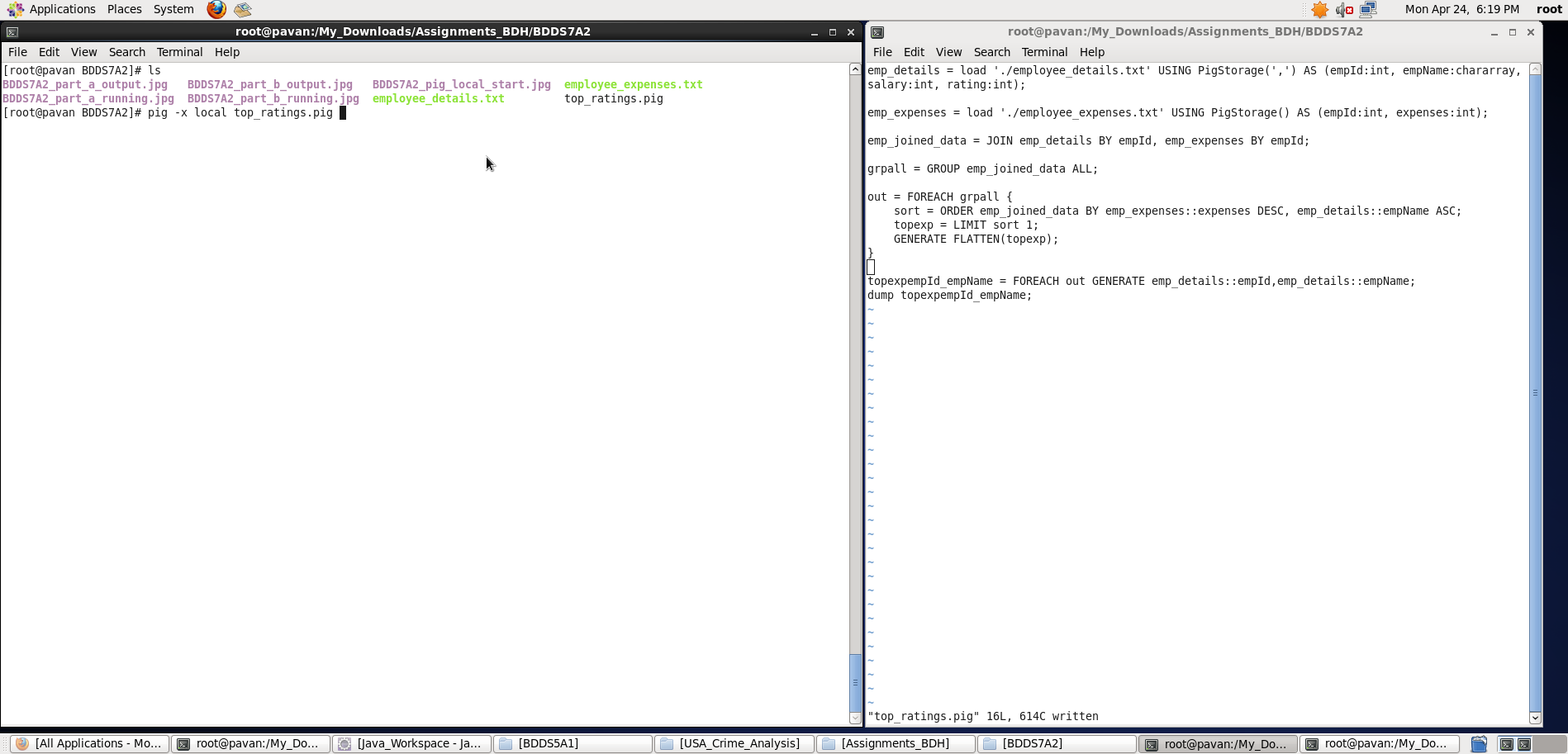
dump topexpempId\_empName;

**Explanation :**

* Loaded data from employee\_details.txt into “emp\_details” relation. Here ‘,’ is the delimiter and schema is provided as mentioned in the question
* Loaded data from employee\_expenses.txt into “emp\_expenses” relation. Here, schema is provided as mentioned in the question
* Both the relations are joined with the same key empID.
* Then, all records are grouped together
* Now, iterating over tuples inside the group I have sorted the emp\_joined\_data by expenses (from emp\_expenses) in descending order and then those with same expenses are sorted in ascending order according to the employee with name coming first in dictionary.
* Limiting my output to top 1 record
* Then flattening the tuples inside group
* **Outputting only “the employee id and employee name” with maximum expense as per Question using dump command.**

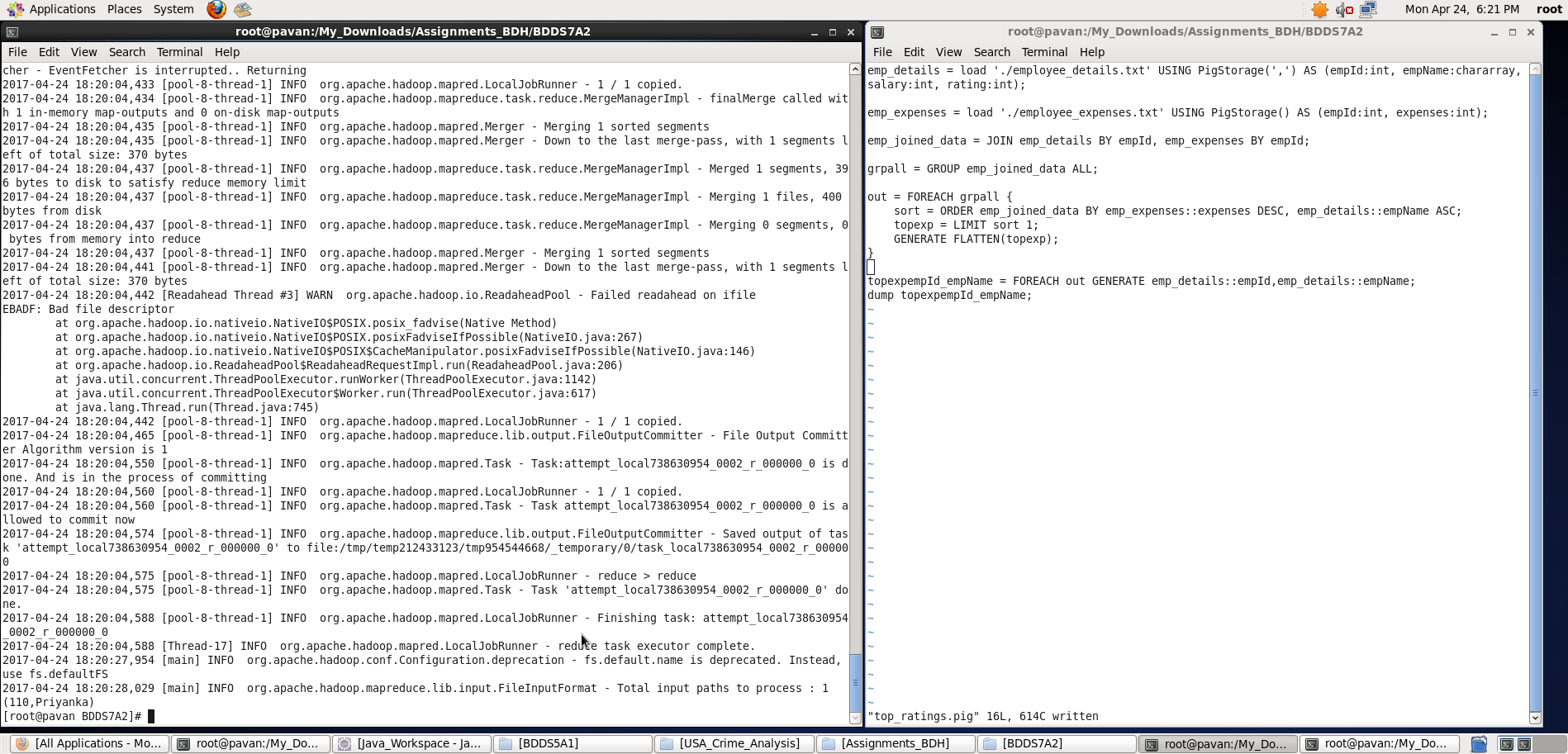
Screenshots

Running **Pig Latin Script in Local Mode** inside /My\_Downloads/Assignments\_BDH/BDDS7A2/ directory



Output displayed: **<employee\_id> <employee\_name>**

**Note : We can store the output if required in a file using “store” command.**



d) **List of employees (employee id and employee name) having entries in employee\_expenses file**

**Pig Latin Script:**

emp\_details = load './employee\_details.txt' USING PigStorage(',') AS (empId:int, empName:chararray, salary:int, rating:int);

emp\_expenses = load './employee\_expenses.txt' USING PigStorage() AS (empId:int, expenses:int);

emp\_joined\_data = JOIN emp\_details BY empId, emp\_expenses BY empId;

out = FOREACH emp\_joined\_data GENERATE emp\_details::empId,emp\_details::empName;

list\_emp\_with\_expenses = DISTINCT out;

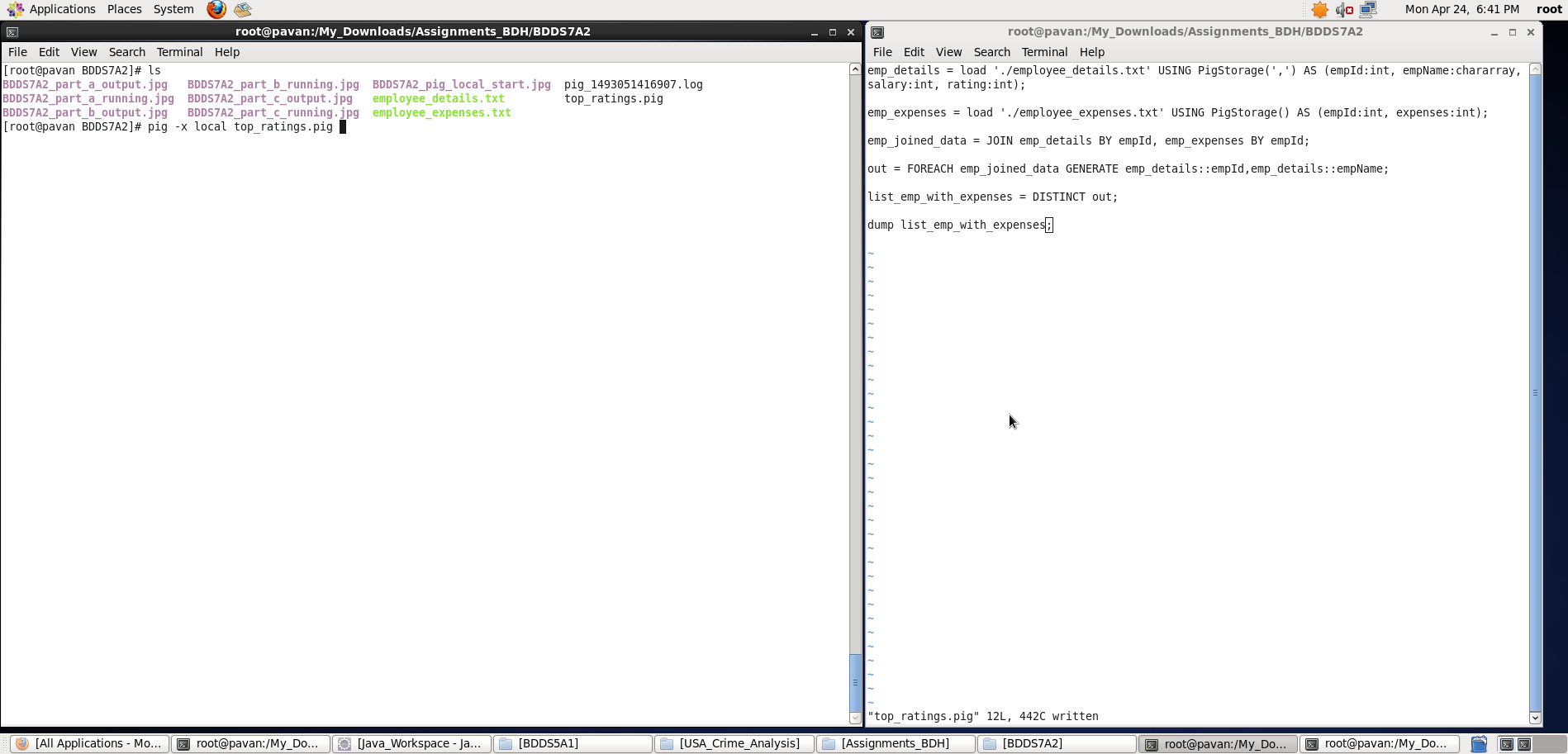
dump list\_emp\_with\_expenses;

**Explanation :**

* Loaded data from employee\_details.txt into “emp\_details” relation. Here ‘,’ is the delimiter and schema is provided as mentioned in the question
* Loaded data from employee\_expenses.txt into “emp\_expenses” relation. Here, schema is provided as mentioned in the question
* Both the relations are joined with the same key empID
* **Outputting only the list of employees: “the employee id and employee name” who have entries in expenses as per Question using dump command.**
* **Note: since expenses are provided twice for two employees, I have used DISTINCT to get the exact list without redundancy.**

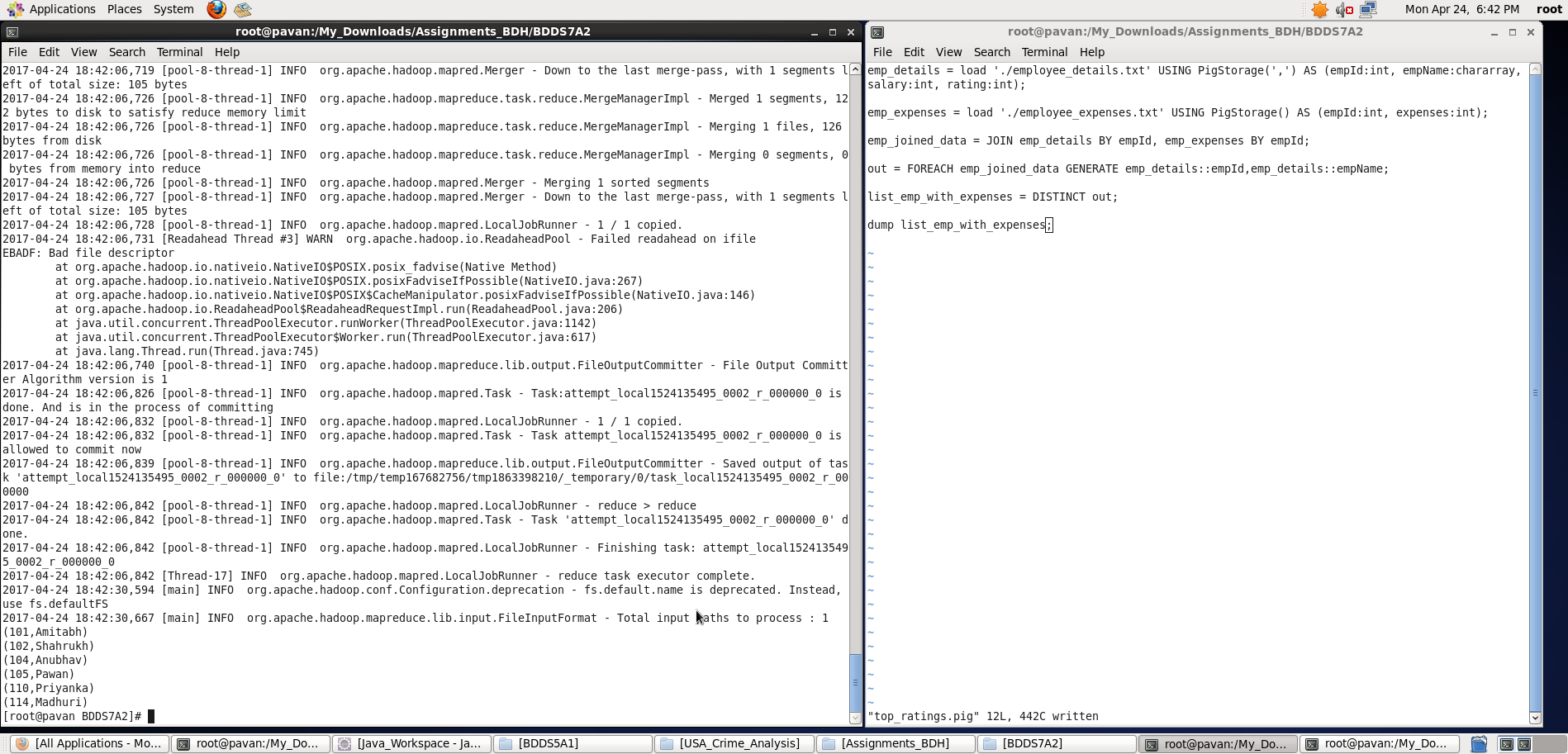
Screenshots

Running **Pig Latin Script in Local Mode** inside /My\_Downloads/Assignments\_BDH/BDDS7A2/ directory



Output displayed: **<employee\_id> <employee\_name>**

**Note : We can store the output if required in a file using “store” command.**



e) **List of employees (employee id and employee name) having no entry in employee\_expenses file.**

**Pig Latin Script:**

emp\_details = load './employee\_details.txt' USING PigStorage(',') AS (empId:int, empName:chararray, salary:int, rating:int);

emp\_expenses = load './employee\_expenses.txt' USING PigStorage() AS (empId:int, expenses:int);

emp\_joined\_data = JOIN emp\_details BY empId LEFT OUTER, emp\_expenses BY empId;

dump emp\_joined\_data;

leftover = FILTER emp\_joined\_data BY emp\_expenses::empId is null;

list\_emp\_no\_expenses = FOREACH leftover GENERATE emp\_details::empId,emp\_details::empName;

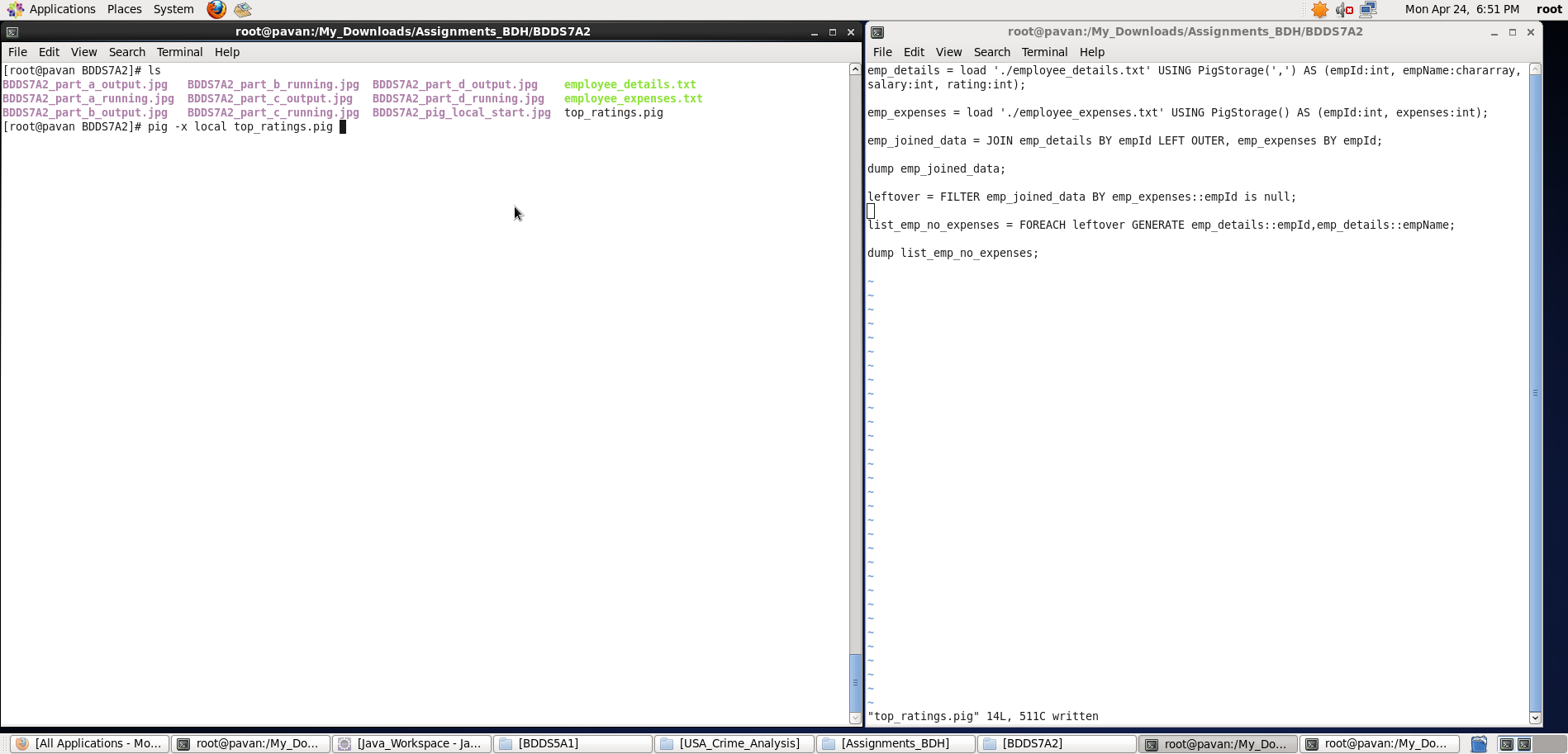
dump list\_emp\_no\_expenses;

**Explanation :**

* Loaded data from employee\_details.txt into “emp\_details” relation. Here ‘,’ is the delimiter and schema is provided as mentioned in the question
* Loaded data from employee\_expenses.txt into “emp\_expenses” relation. Here, schema is provided as mentioned in the question
* Both the relations are joined with the same key empID
* Note: Here emp\_details relation is **LEFT OUTER Joined**
* **Now, filtering out those employees whose expenses field is NULL , we get the desired output**
* **Outputting only the list of employees: “the employee id and employee name” who don’t have entry in expenses as per Question using dump command.**

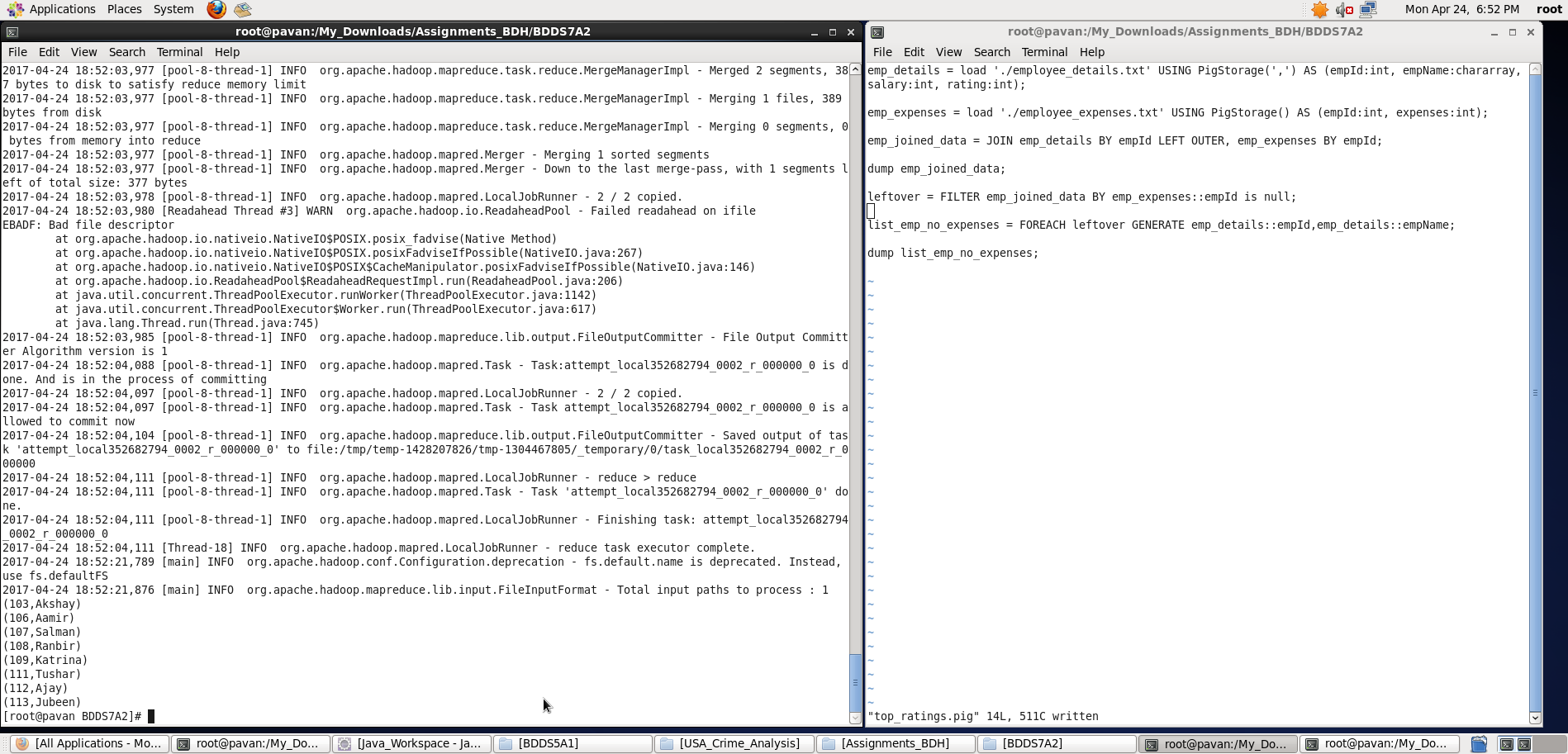
Screenshots

Running **Pig Latin Script in Local Mode** inside /My\_Downloads/Assignments\_BDH/BDDS7A2/ directory



Output displayed: **<employee\_id> <employee\_name>**

**Note : We can store the output if required in a file using “store” command.**



Thus, with the help of Pig Latin script , I have displayed the results for all the 5 tasks.