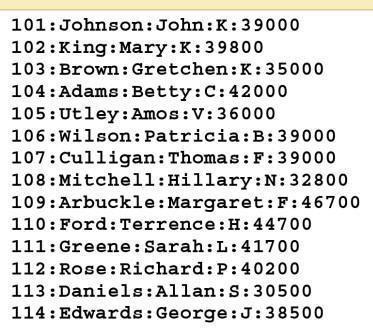
CPAN133 – Operating Systems Winter 2023

**Exercise-6**

Due: Wednesday, 11:59 PM

Part1: Exercise on advance file processing commands: cut, sort, uniq, comm.[3 marks]

Write the command for each of the following questions.

1. Consider the following two files: create these two files by using the cat command (text file is provided you can copy from there and paste it to the shell ).
   1. programmer:

**A: cat > programmer.txt**

**101:Johnson:John:K:39000**

**102:King:Mary:K:39800**

**103:Brown:Gretchen:K:35000**

**104:Adams:Betty:C:42000**

**105:Utley:Amos:V:36000**

**106:Wilson:Patricia:B:39000**

**107:Culligan:Thomas:F:39000**

**108:Mitchell:Hillary:N:32800**

**109:Arbuckle:Margaret:F:46700**

**110:Ford:Terrence:H:44700**

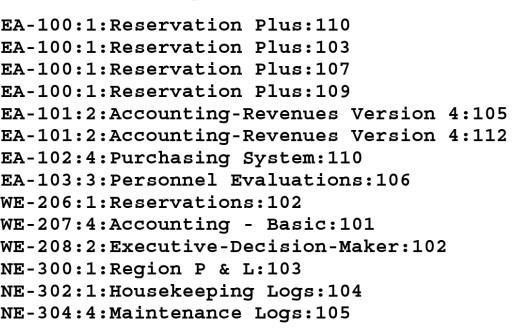
**111:Greene:Sarah:L:41700**

**112:Rose:Richard:P:40200**

**113:Daniele:Akkab:S:30500**

**114:Edwards:George:J:38500**

* 1. project



**A: cat > project.txt**

**EA-100:1:Reservation Plus:110**

**EA-100:1:Reservation Plus:103**

**EA-100:1:Reservation Plus:107**

**EA-100:1:Reservation Plus:109**

**EA-101:2:Accounting-Renevues Version 4:105**

**EA-101:2:Accounting-Renevues Version 4:112**

**EA-102:4:Purchasing System:110**

**EA-103:3:Personnel Evaluations:106**

**WE-206:1:Reservations:102**

**WE-207:4:Accounting - Basic:101**

**WE-208:2:Executive-Decision-Maker:102**

**NE-300:1:Region P & L:103**

**NE-302:1:Housekeeping Logs:104**

**NE-304:4:Maintance Logs:105**

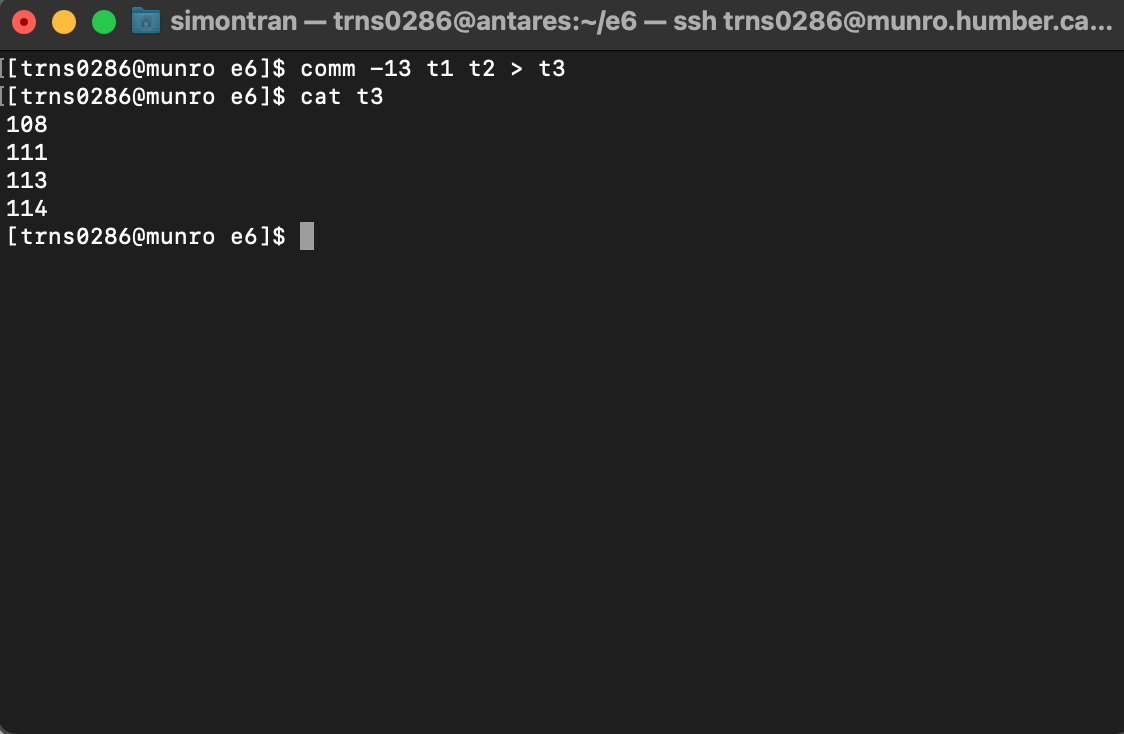
* 1. Cut the programmer\_number fields from the project file (field 4), and piping the output to the sort command to place any duplicate numbers together. Pipe the sorted output to the uniq command to remove any duplicate programmer\_numbers. Finally, redirect the output to a file t1. (The t1 file is a list of programmer numbers that identifies programmers who are assigned to projects.)

**A: cut -d ':' -f4 project.txt | sort | uniq > t1**

* 1. The next step is to cut the programmer\_number field (field 1) from the programmer file, and pipe the output as you did in Step **c.** Finally, redirect the output to a file t2, which is a list of programmer numbers that identifies all of the programmers.

**A: cut -d ":" -f1 programmer.txt | sort | uniq > t2**

* 1. Now that t1 and t2 are sorted in the same order, you can match them. Use the comm command to select the lines from t1 that do not match lines in t2, and redirect the output to another file t3, which lists programmer numbers of all programmers who are not assigned to projects. Display the content of file f3, take the screenshot and attached here.



2) We use diff command to find out the differences between two files. The output that is returned is the following, [1 marks]

Command: $ diff file1 file2

Output:

1,2c1,2

< alice

< Alex

---

> Allen

> Alix

4c4

< kat

---

> Kit

Write in your own words how you will interpret this output.

**A: Lines 1 and 2 in file1 must be changed to match lines 1 and 2 in file2. The words alice and Alex are from lines 1 and 2 in file1 and the words Allen and Alix are from lines 1 and 2 in file2.**

**Line 4 from file1 must be changed to match line 4 in file2. The word kat is from line 4 of file1 and the word Kit is from line 4 of file2.**

3) Write the commands for each of the following questions: [2 mark]

1. Use the grep command to search for all of the lines that contain the pattern 'Lines' from a file called input.txt and save/send those lines in the file output.txt

**A: grep ‘Lines’ input.txt > output.txt**

1. Use the tr command to convert all the characters from lowercase to uppercase of the file output.txt

**A: cat input.txt | tr '[a-z]' '[A-Z]'**

**Part3:** Exercise with egrep commands [3 marks]

1. Try the following commands with the file words which is in: (/usr/share/dict)

In each case explain the result

1. egrep –ic “video” words
2. egrep "^video$" words
3. egrep "(video|stereo)tape" words
4. egrep "^[^a-z]" words
5. egrep "^[^a-z]?" words
6. egrep "^[^a-z]$” words

**A:**

**a) Will return the number of lines ‘video’ appears in words, ignoring case sensitivity.**

**b) Will return lines that only have ‘video’.**

**c) Will return lines that have either ‘videotape’ or ‘stereotape’.**

**d) Will return lines that do not start with lower case letter.**

**e) Will return everything.**

**f) Will return lines that do not start or end with a lower case letter.**

**Part4: [1 marks]**

1. When we create a link of a file by using the following command, which is not true about it-

ln –s myfile.txt myfile.sym.lnk

a) If the file is deleted/renamed the symbolic link would not be valid

b) Applicable to files only

c) Valid across different file systems

d) if the file is changed the symbolic link will reflect the new content

**A: b) is not true, soft links can be created for directories.**

1. When we create a link of a file by using the following command, which is true about it

ln myfile.txt myfile.hard.lnk

a) If the file is deleted the hard link would point to a invalid file

b) Applicable to files and directories

c) Valid across different file systems

d) Points to the i‐node of the file

**A: d) is true, it points to the inode of the file.**

1,2 This exercise is from: **Palmer, Michael J. *Guide to UNIX using Linux*. 4th ed. Boston, MA: Course Technology, 2008. Web**