CECS 536, Fall 2020 Take Home 1

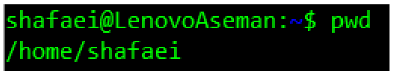
Sima Shafaei

INSTRUCTIONS:

This is an open notes exam. You can consult your handouts and class notes, rewatch the class videos, even Google for answers. However, you should not ask another person for help. Each answer that you submit should be your own work. If you cannot complete a question, you can show your work for partial credit (show the commands that you tried, and explain why you think they didn't work). Please answer using only the command line tools that we have discussed in class; do not use tools we have not covered (awk, sed, shell functions). Give only one answer per question. The exam has 3 parts, with multiple questions on each. Below it is shown how much each part is worth; you can assume that all questions are worth the same. On each question, give only the command that you use to solve the question (C) or the command and the answer that you obtain (C+A), as requested.

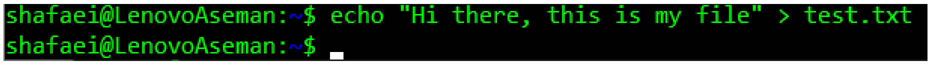
1. (16 points) Log into the Unix/Ubuntu terminal. Do the following in the order given.

(a) (C+A) show your home directory.



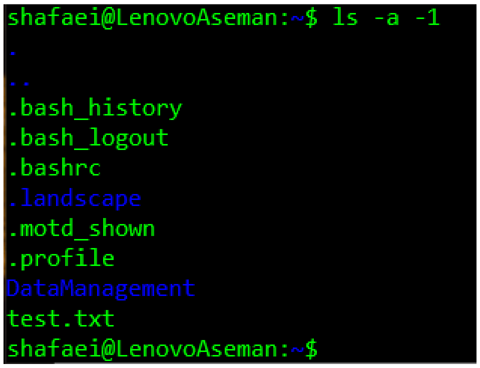
```
shafaei@LenovoAseman:~$ pwd
/home/shafaei
```

(b) (C) Create a file called test.txt by using the command echo ‘‘Hi there, this is my file’’ >test.txt



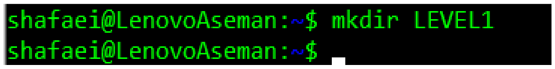
```
shafaei@LenovoAseman:~$ echo "Hi there, this is my file" > test.txt
shafaei@LenovoAseman:~$
```

(c) (C+A) List all files in your home directory.



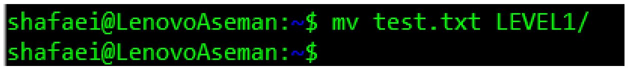
```
shafaei@LenovoAseman:~$ ls -a -1
.
..
.bash_history
.bash_logout
.bashrc
.landscape
.motd_shown
.profile
DataManagement
test.txt
shafaei@LenovoAseman:~$
```

(d) (C) Create a directory called LEVEL1.



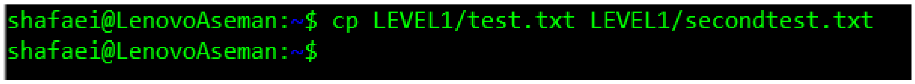
```
shafaei@LenovoAseman:~$ mkdir LEVEL1
shafaei@LenovoAseman:~$
```

(e) (C) move the file test.txt to inside directory LEVEL1.



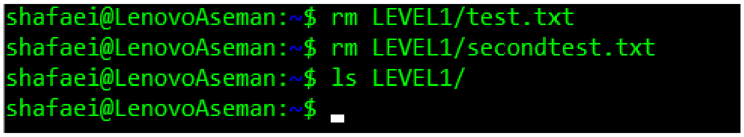
```
shafaei@LenovoAseman:~$ mv test.txt LEVEL1/
shafaei@LenovoAseman:~$
```

(f) (C) copy the file test.txt into another file called secondtest.txt; this new file should be also inside LEVEL1.



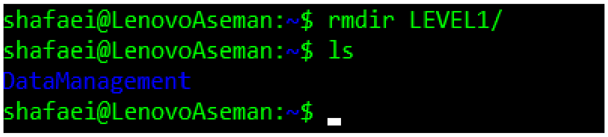
```
shafaei@LenovoAseman:~$ cp LEVEL1/test.txt LEVEL1/secondtest.txt
shafaei@LenovoAseman:~$
```

(g) (C) delete the files test.txt and secondtest.txt.



```
shafaei@LenovoAseman:~$ rm LEVEL1/test.txt
shafaei@LenovoAseman:~$ rm LEVEL1/secondtest.txt
shafaei@LenovoAseman:~$ ls LEVEL1/
shafaei@LenovoAseman:~$
```

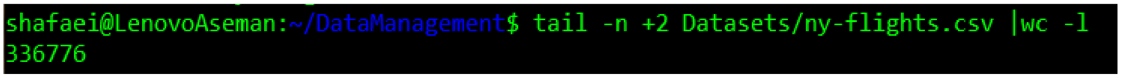
(h) (C) delete the directory LEVEL1.



```
shafaei@LenovoAseman:~$ rmdir LEVEL1/
shafaei@LenovoAseman:~$ ls
DataManagement
shafaei@LenovoAseman:~$
```

2.(66 points) The file ny-flights.csv is available on Blackboard, in the folder DATASETS. Download it and move it to your Unix/Ubuntu home directory. The file contains a list of flights originating on any New York airport (JFK, LGA and EWR are included) to any other destination, one flight per line. You can see the header of the file using head -1, as usual. Using this file, answer the following questions.

(a) (C+A) how many flights are there in this dataset?

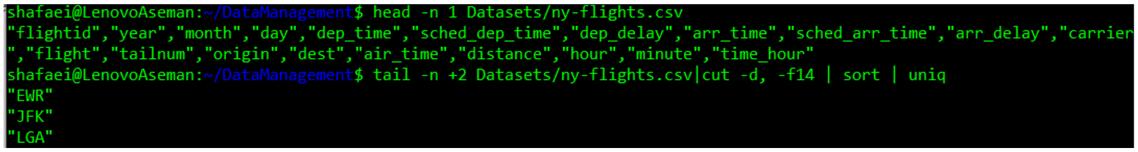


```
shafaei@LenovoAseman:~/DataManagement$ tail -n +2 Datasets/ny-flights.csv |wc -l
336776
```

Explanation of the command:

The first part (tail -n +2 Datasets/ny-flights.csv) will remove the header of the dataset, the second part (wc -l) will count the number of lines in the dataset

(b) (C) How could you check what I just said, about this being a list of flights originating only at JFK, LGA, or EWR?



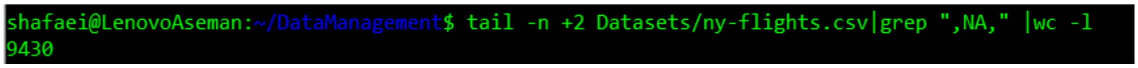
```
shafaei@LenovoAseman:~/DataManagement$ head -n 1 Datasets/ny-flights.csv
"flightid","year","month","day","dep_time","sched_dep_time","dep_delay","arr_time","sched_arr_time","arr_delay","carrier
","flight","tailnum","origin","dest","air_time","distance","hour","minute","time_hour"
shafaei@LenovoAseman:~/DataManagement$ tail -n +2 Datasets/ny-flights.csv|cut -d, -f14 | sort | uniq
"EWR"
"JFK"
"LGA"
```

Explanation of the command:

The first command will show the header of dataset. So I can see which field shows the origin of the flights (field number 14)

In second command: The first part (tail -n +2 Datasets/ny-flights.csv) will remove the header of the dataset, the second part (cut –d, -f14) will select only the value of origin in each line, the third part sort the value of origin and the last part (uniq) shows all existing value in the field of origin

(c) (C+A) This file has missing values. These values are marked as “NA” (not available). Can you find out how many records/lines have at least one of those?

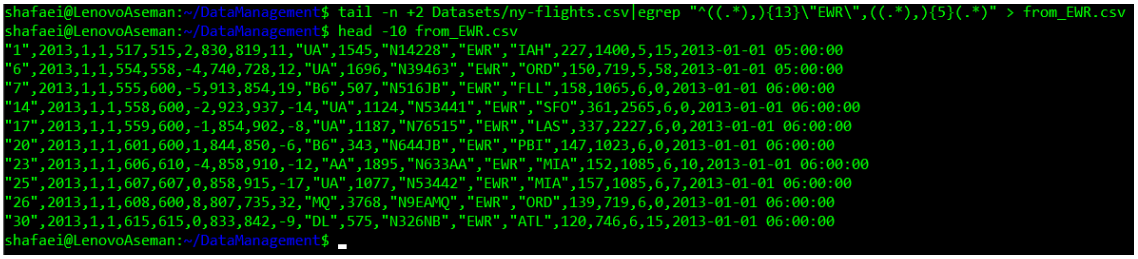


```
shafaei@LenovoAseman:~/DataManagement$ tail -n +2 Datasets/ny-flights.csv|grep ",NA," |wc -l
9430
```

Explanation of the command:

The first part (tail -n +2 Datasets/ny-flights.csv) will remove the header of the dataset, the second part (grep ",NA,") will search for the fields with that value of "NA" and the last part (wc -l) will count the number of lines that have "NA" value in them

(d) (C) create a file with all the flights from EWR (not to EWR!).



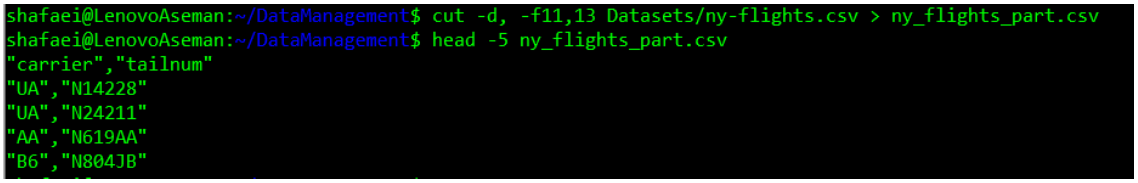
```
shafaei@LenovoAseman:~/DataManagement$ tail -n +2 Datasets/ny-flights.csv|egrep "^((.*),){13}\"EWR\",((.*),){5}(.*)" > from_EWR.csv
shafaei@LenovoAseman:~/DataManagement$ head -10 from_EWR.csv
"1",2013,1,1,517,515,2,830,819,11,"UA",1545,"N14228","EWR","IAH",227,1400,5,15,2013-01-01 05:00:00
"6",2013,1,1,554,558,-4,740,728,12,"UA",1696,"N39463","EWR","ORD",150,719,5,58,2013-01-01 05:00:00
"7",2013,1,1,555,600,-5,913,854,19,"B6",507,"N516JB","EWR","FLL",158,1065,6,0,2013-01-01 06:00:00
"14",2013,1,1,558,600,-2,923,937,-14,"UA",1124,"N53441","EWR","SFO",361,2565,6,0,2013-01-01 06:00:00
"17",2013,1,1,559,600,-1,854,902,-8,"UA",1187,"N76515","EWR","LAS",337,2227,6,0,2013-01-01 06:00:00
"20",2013,1,1,601,600,1,844,850,-6,"B6",343,"N644JB","EWR","PBI",147,1023,6,0,2013-01-01 06:00:00
"23",2013,1,1,606,610,-4,858,910,-12,"AA",1895,"N633AA","EWR","MIA",152,1085,6,10,2013-01-01 06:00:00
"25",2013,1,1,607,607,0,858,915,-17,"UA",1077,"N53442","EWR","MIA",157,1085,6,7,2013-01-01 06:00:00
"26",2013,1,1,608,600,8,807,735,32,"MQ",3768,"N9EAMQ","EWR","ORD",139,719,6,0,2013-01-01 06:00:00
"30",2013,1,1,615,615,0,833,842,-9,"DL",575,"N326NB","EWR","ATL",120,746,6,15,2013-01-01 06:00:00
shafaei@LenovoAseman:~/DataManagement$
```

Explanation of the command:

In the first command, the first part (tail -n +2 Datasets/ny-flights.csv) will remove the header of the dataset, the second part will select all flights from EWR and save the result in a new file (from_EWR.csv).

- ((.*),){13} : The value first 13 fields can be anything that are separated with a comma
- \"EWR\", : The value of 14th field should be "EWR" and end with a comma
- ((.*),){5}(.*): after origin field we will have 6 fields which only the last one does not have a

(e) (C) create a file with two attributes/columns, carrier and tailnumber.



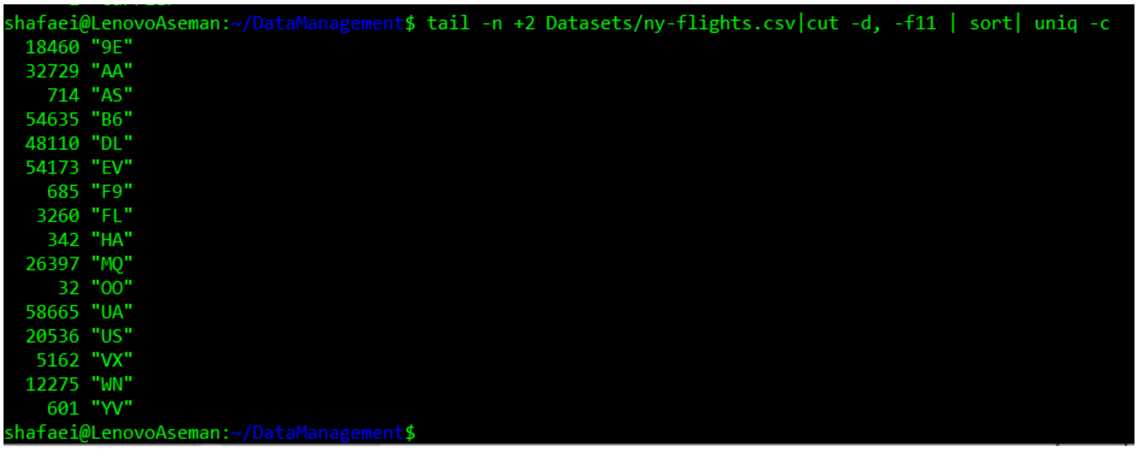
```
shafaei@LenovoAseman:~/DataManagement$ cut -d, -f11,13 Datasets/ny-flights.csv > ny_flights_part.csv
shafaei@LenovoAseman:~/DataManagement$ head -5 ny_flights_part.csv
"carrier","tailnum"
"UA","N14228"
"UA","N24211"
"AA","N619AA"
"B6","N804JB"
```

Explanation of the command:

First command (cut –d, -f11,13) will select carrier and tailnumber fields and save the result in a new file (ny_flight_part.csv)

The second command only shows the result of previous command, we represent only 5 first lines of ny_flight_part.csv

(f) (C+A) find the number of flights per carrier.

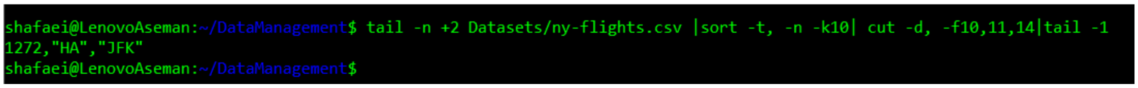


```
shafaei@LenovoAseman:~/DataManagement$ tail -n +2 Datasets/ny-flights.csv|cut -d, -f11 | sort| uniq -c
  18460 "9E"
  32729 "AA"
    714 "AS"
  54635 "B6"
  48110 "DL"
  54173 "EV"
    685 "F9"
   3260 "FL"
    342 "HA"
  26397 "MQ"
     32 "OO"
  58665 "UA"
  20536 "US"
   5162 "VX"
  12275 "WN"
    601 "YV"
shafaei@LenovoAseman:~/DataManagement$
```

Explanation of the command:

the first part of this command (tail -n +2 Datasets/ny-flights.csv) will remove the header of the dataset, then the second part (cut –d, -f11) will select the carrier field. The third part (sort) will sort value of carrier field because uniq command needs a list of sorted value as input. Finally, the last part (uniq -c) will show number of iteration for each carrier in dataset.

(g) (C+A) find the flight with the longest arrival delay. Identify carrier and origin airport.



```
shafaei@LenovoAseman:~/DataManagement$ tail -n +2 Datasets/ny-flights.csv |sort -t, -n -k10| cut -d, -f10,11,14|tail -1
1272,"HA","JFK"
shafaei@LenovoAseman:~/DataManagement$
```

Explanation of the command:

the first part of this command (tail -n +2 Datasets/ny-flights.csv) will remove the header of the dataset, then the second part (sort –t, -n –k10) will sort dataset based on the value of 10th field, the third part (cut –d, -f10,11,14) will select the arrival delay, carrier and origin airport fields. And finally the last part (tail -1) will select last line in the sorted dataset which has the longest arrival delay.

(h) (C) find the flights that departed on time (departure delay = 0).

```
tail -n +2 Datasets/ny-flights.csv|egrep "((.*),){6}0,((.*),){12}(.*)"
```

Explanation of the command:

the first part of this command (tail -n +2 Datasets/ny-flights.csv) will remove the header of the dataset, then the second part (`egrep "((.*),){6}0,((.*),){12}(.*)"`) will search for a zero in 7th field:

- ((.*),){6} : The value first six fields can be anything that are separated with a comma
- 0, : The value of 7th field should be zero that finally ends with a comma
- ((.*),){12}(.*): after departure delay we will have 13 fields which only the last one does not have a comma after it.

(i) (C) find the flights that departed before scheduled (departure delay is negative).

```
tail -n +2 Datasets/ny-flights.csv|egrep "((.*),){6}-[1-9](.*),((.*),){12}(.*)"
```

Explanation of the command:

the first part of this command (tail -n +2 Datasets/ny-flights.csv) will remove the header of the dataset, then the second part (egrep "((.*),){6}-[1-9](.*),((.*),){12}(.*)") will search for a negative number in 7th field:

- ((.*),){6} : The value first six fields can be anything that are separated with a comma
- -[1-9](.*), : The value of 7th field should start with a – then we expect a digit between 1 to 9 and then we can have any number after that finally this field will end with a ,
- ((.*),){12}(.*): after departure delay we will have 13 fields which only the last one does not have a comma after it.

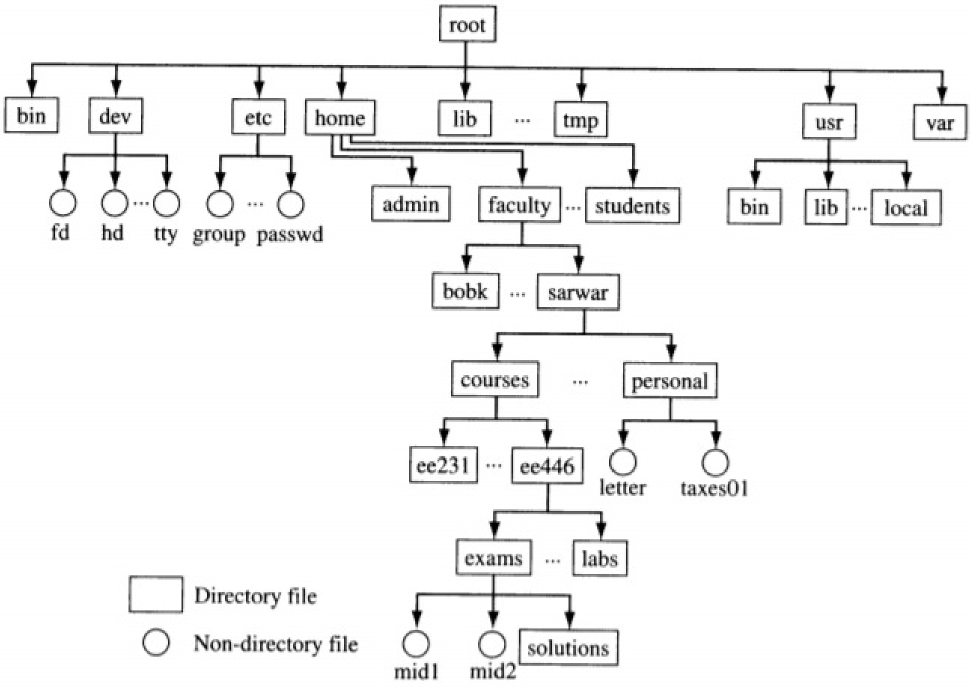


Figure 1: Example file system

3. (18 points) Suppose your file system looks like the one in Figure 1. Your home directory is /home/faculty/sarwar. You log in the system and are now in your home directory.

(a) You try to run 'ls ee446/exams'. Does this work or give an error? No it will not work because e446 is not a direct subdirectory in my home directory

(b) You try to run 'ls courses/ee446/exams'. Does this work or give an error? Yes this command will work and the output would be: solution mid1 mid2

(c) You try to run ’ls /courses/ee446/exams’. Does this work or give an error? This command will not work because courses is not a subdirectory in root

(d) You want to move file ”mid1” to your current directory. What is the command to do this?

```
mv courses/ee446/exams/mid1 ./
```

(e) You want to copy file ”mid2” to /home/faculty/bobk. What is the command to do this?

```
cp courses/ee446/exams/mid2 bobk
```

(f) How do you move from your home directory to /home/faculty/?

```
cd ..
```