Emori, Noboru  
CIS 045, Fall 2020

G08125504  
Quiz Week 6

2020/10/15

Quiz-6

We will use the below data file for this quiz.

**"datafile"**

----------------------------------------

Name:DateOfBirth:Salary:Hours-worked

joey:10-05-1994:65000:40

peter:04-13-1990:50000:40

sy:02-22-1999:1000000:20

-----------------------------------------

*1.****file=datafile ;echo $file***

This prints “datafile”, confirming the variable called file is “datafile”

*3. What do you expect when you execute the following command?*

***$ cut -d: -f3 $file***

This prints the 3rd field with the : as a delimiter, so it would output

Salary  
65000  
50000  
1000000

*4. Write a cut command to get the name and salary from datafile?*

**$ cut -d: -f1,3 $file**

*5. What happens when you execute this command?*

***$ cut -d: -f3 < $file***

It feeds the content of datafile as stdin to $cut -d: -f3, so in this case it will output the same as question 3

Salary  
65000  
50000  
1000000

*6.  Get the first five characters of each line in the datafile*

**$ cut -c '1-5' $file**

**Using the 'tr' command**

      [ tr - translate or delete characters ]  
**SYNOPSIS**        [ **tr** [OPTION]... SET1 [SET2] ]

  Create this names.txt file with the names listed.

--------------

joe richards

mac arther

joe richard

lynn  nguyen

fenj Leu

----------------

*7. Write a command using 'tr' to uppercase all names in the file.*

**$ file=names.txt** # this will remain effective for question 7 and 8

**$ cat $file | tr 'a-z' 'A-Z'**

*8. Replace all lower case characters in the file with an 'X'.*

**$ cat $file | tr '[a-z]' 'X'**

*9.  Write a sed or other command to mask the salary in the data file so that the output of each line looks like this below:*

***joey:10-05-1994:\*\*\*\*\*:40***

Find the pattern of numbers between the third and fourth : and replace with \*\*\*\*\*:

**$ sed 's/^\(.\*\):\(.\*\):\([0-9]\+\):\(.\*\)$/g\1:\2:\*\*\*\*\*:\4/' datafile**

*10. Write a command to count the number of lines in the datafile.*

**$ wc -l < datafile**

*11. Write a command to count the numbers of characters in the datafile.*

**$ wc -m < datafile**

*12. When you run the below command on datafile, the output seen is listed below. For full credit, explain what is happening in detail as to why you got the output below.*

***$ sed '1h;2d;3H;4g' datafile***

*name:dob:salary:hours-worked*

*peter:04-13-1990:50000:40*

*name:dob:salary:hours-worked*

*peter:04-13-1990:50000:40*

*Look at the original datafile and describe what happened to line number 4 of the datafile?*

1h puts content of line 1 into holding buffer, but still prints it out because it’s in the pattern buffer  
2d deletes the content of line 2 in the pattern buffer, so prints out nothing. 1 is still in holding   
3H appends the content of line 3 into holding buffer, but still prints out 3 from the pattern buffer  
4g replaces the pattern buffer for line 4 with the holding buffer, which is lines 1 and 3, and prints lines 1 and 3 (the new content).

Thus, the command prints line 1, line 3, and line 4, but line 4 was replaced with lines 1 and 3.

*13.  Write a sed command to produce the output below from the datafile. The command will look very similar to number 12.*

*name:dob:salary:hours-worked*

*peter:04-13-1990:50000:40*

*sy:02-22-1999:1000000:20*

*name:dob:salary:hours-worked*

**$ sed '1h;2d;4G' datafile**

14.  Write a sed command to split up the datafile so that the first line goes into a file called 'df1', the second and third line will go into a file called 'df2' and the fourth line will go into a file called 'df3'.

**$ sed -e '1w df1' -e '2,3w df2' -e '4w df3' datafile**

*15. Write a sed command to match lines in a file that does not begin with a space or a tab from a file called xfile. To get a space, just hit the space bar. To get a Tab, you should use the Contrl+letter v, then hit the Tab key.*

***$ sed -n '/^[^Space ctrl+v Tab]/p'  xfile***

As noted, the above works, actually putting the space and tab characters:

**$ sed -n '/^[^ ]/p' xfile**

*16. Write a sed command to match lines that begins with a space or a tab*

**$ sed -n '/^[ ]/p' xfile**

 Inside the brackets are space and tab characters (inserted by Space bar and ctrl+v Tab)

*17. Write a sed command to delete lines 1 to 3 in datafile*

**$ sed '1,3d' datafile**

*18. Write a sed command to delete lines 2 to the end of the datafile.*

**$ sed '2,$d' datafile**

*19. Given the two files: filea and fileb*

*filea                                                      fileb*

*I am one                                       I am one*

*I am one                                       I am one*

*I AM LINE TWO                     I AM LINE TWO*

*I am line two                             I am line two*

*I am line 3 in filea                   I am line 3 in fileb*

* *Write a command to show the differences between the two files.*

**$ diff filea fileb**

* *Write a command to show only unique lines in the each file.*
  + **$ uniq filea; uniq fileb**
* *What does the -i option do ?*

***$ uniq -i filea***The -i option ignores case, so I AM LINE TWO and I am line two are no longer unique

* *What does the -c option do?*

***$ uniq -i -c  filea***

The -c option prefixes the printed unique lines by the number of times it happens. So “I am one” and “I AM LINE TWO” will both have 2 (“I am line two” counts because of -i), while “I am line 3 in file a” will be preceded by 1.

*20.  Write a find command to find all regular files under your home directory belonging to your LOGNAME that were created WITHIN the last 3 days that are between the size of 4 to 5 Megabytes and when you find them, you should tar them up.*

**$ find $HOME -type f -user $USER -ctime -3 -size +4M -a -size -5M -exec tar -cvf myfiles.tar '{}' \;**

*21. Write a find command and use it in conjunction with xargs to find then name of any Regular file under your home directory that has the word 'xyz' inside of the file.*

**$ find $HOME -type f | xargs grep -il 'xyz'**

Remove the -i option to make it case sensitive

*22. Write a command to find which file under the $HOME directory have lines that are 80 characters or more long.*

**$ grep -ERl "^.{80,}$"  $HOME | more**

Note that that option is el, not eye

*23. Let's see the lines that actually matches*

**$ grep -ERnH "^.{80,}$" $HOME | more**

This gives both file name and line number, which are useful. | more is highly recommended here because there may be many, many lines

*24.  Write a recursive grep command to search all files in your home that has the word 'homework'*

**$ grep -ERl "homework" $HOME | more**

*25.  Write a find to find all files with inode number 12345 and move them to the /tmp/storage directory.*

**$ find / -inum 12345 -exec mv '{}' /tmp/storage \;**