

COMP2123 Programming technologies and tools

Assignment 1 - Linux commands and shell script

Due: Dec. 13, 2017

Dear Students,

Hello! If you have any questions w.r.t. this assignment, **please first visit our Assignment 1 Discussion Forum to see if you can find our answer there.** If not, please post your questions in the Forum.

- For Question 1, you must complete it online through Moodle.
- For Question 2, please submit it through Moodle, and click the submit button before the deadline.
- Please double check your submission, check ALL of these to ensure that your assignment is submitted properly:
 - Check your email for acknowledgement from Moodle, keep it for record.
 - Check the assignment page again and check if the status is changed to “Submitted for grading”.
 - Check the assignment page to see all files you have submitted.
- Please contact us as soon as possible if you encounter any problem regarding the submission.

Enjoy this assignment and let us know if you need our help.

Some of the questions require techniques not covered in the self-learning materials. You may need to search for the techniques to solve those problems.

Question 1 [20%] Please answer the following question on Moodle. When answering the following questions, assume that you have logged in Fedora and started a bash shell.

You need to choose ALL correct answers.

a. Suppose the command “ls -l” gives the following result:

```
drwx-r--r-- 1 c1123 c1123 4096 Sep  4 10:30 myDir
-rw-r--r-- 1 c1123 c1123    1 Sep  4 10:29 myFile
```

Which of the following commands will have effect to the information shown in the above result?

- o touch myDir
- o mkdir myDir
- o rm myFile
- o cp myFile myDir

b. Suppose the command “ls -l” gives the following result:

```
drwxrwxr-x 1 c1123 c1123 4096 Sep  4 10:30 myDir
-rw-rw-r-- 1 c1123 c1123    1 Sep  4 10:29 myFile
```

If you are c1123, which of the following commands, when issued, will create a copy of “myFile”, in “myDir”? (“myFile” in the current directory must stay intact after the command)

- o mv myFile myDir
- o cp * myDir
- o cp -t myDir myFile
- o cat < myFile > myDir/myFile

c. Which of the following commands lists the names of all folders and files in the current directory, whose name contains characters 'ab'. **Hidden files and files in subdirectories do not need to be listed.**

- o ls *ab*
- o ls | grep ab
- o ls | grep *ab*
- o find . -name 'ab'

d. Suppose you are in your home directory and issued the command “pwd”. You get the following result:

`/home/c1123`

After some operations, you forget where you are and so you issued the command “pwd” again. You get the following result:

`/home/c1123/A1/Q1d`

Which of the following commands will send you back to your home directory?

- o cd
- o cd ~
- o cd ~/c1123
- o cd /home/c1123
- o cd ../../

e. Which of the following commands will try to compile a C++ program without printing anything on screen? There may be compilation error in the C++ program.

- o g++ a1.cpp -o a1.out > output.txt
- o g++ a1.cpp -o a1.out 1> output.txt
- o g++ a1.cpp -o a1.out 1> output.txt 2> output.txt
- o g++ a1.cpp -o a1.out > output.txt 2>&1
- o g++ a1.cpp -o a1.out 2>&1 > output.txt

f. Which of the following commands will correctly print the total number of files and folders in the “asg1” directory? File names may consist of spaces. Hidden files need not be counted.

- o ls -a asg1 | wc -l (**LS dash A, WC dash L**)
- o ls -l asg1 | wc -l (**LS dash L, WC dash L**)
- o ls -1 asg1 | wc -w (**LS dash one, WC dash W**)

o `ls -l asg1 | wc -l` (**LS dash one, WC dash L**)

g. Suppose the command “`ls -l`” gives the following result:

<code>-rw-r--r--</code>	<code>1</code>	<code>c1123</code>	<code>c1123</code>	<code>1</code>	<code>Sep 4 10:25</code>	<code>test1.txt</code>
<code>-rw-r--r--</code>	<code>1</code>	<code>c1123</code>	<code>c1123</code>	<code>304</code>	<code>Sep 4 10:26</code>	<code>test2.txt</code>
<code>-rw-r--r--</code>	<code>1</code>	<code>c1123</code>	<code>c1123</code>	<code>594021</code>	<code>Sep 4 10:28</code>	<code>test3.txt</code>
<code>-rw-r--r--</code>	<code>1</code>	<code>c1123</code>	<code>c1123</code>	<code>9123</code>	<code>Sep 4 10:30</code>	<code>test4.txt</code>

In order to order the result of “`ls -l`” in **ascending** order of file size, what options need to be added?

- o `-S`
- o `-n`
- o `-d`
- o `-R`
- o `-r`

h. Which of the following patterns, when used in a “`grep`” command with the “`-E`” option, will match with the whole string “`ababcbabcd`”?

- o `[abc]+d`
- o `[abcd]{5,9}`
- o `(ab)*(bc)*(cd)`
- o `([ab]*[bc])*[cd]`

i. Suppose we have the following files under the current directory:

a.txt	b.txt	c.txt	d.txt
apple	orange	apple	apple
apple	apple	orange	apple
apple	apple	apple	orange
	apple	apple	apple

A command is issued and the following result is printed:

<code>2d1</code> <code>< orange</code> <code>3a3</code> <code>> orange</code>
--

What is the command?

- o `diff a.txt c.txt`
- o `diff b.txt c.txt`
- o `diff c.txt d.txt`
- o `diff c.txt b.txt`
- o `diff d.txt b.txt`

j. Which of the following commands will show the total number of “apple” and the total number of “orange” in c.txt (in any order) in part i?

```
o grep -c -E '(apple|orange)' c.txt
o uniq -c c.txt | sort
o sort c.txt | uniq -c
o sort -u c.txt | uniq -c
```

Please complete Question 1 on Moodle. You do not need to submit any file for this question.

Question 2 [80%] Shell programming

7-Eleven is the largest convenience store group in Hong Kong, which offers a wide variety of products and convenient services. Sales records in each convenience store are collected from a central backend system for analysis. The system generates log files every day and each line in the log file represents a transaction made through a specific convenience store and has the following format:



[Timestamp], [ProductID]

- Timestamp has the format YYYY-MM-DD HH:MM:SS.Millisecond.
- ProductID is an 8-digit value identifying a product provided by the company.
- Comma “,” is used as field separator in the log file.
- The log files of 7-Eleven company have “7Eleven_” as prefix in their filenames, followed by the store ID of 5 digits identifying a store location, followed by the date in YYYY-MM-DD, followed by “.log”.

Consider an example 7Eleven_00123_2016-01-01.log file that records the transactions of the store with ID ”00123” on 2016-01-01:

2016-01-01 15:44:41.134203,57553901
2016-01-01 15:50:36.294620,86782909
2016-01-01 15:50:36.294620,86782909
2016-01-01 15:50:36.294620,86782909
2016-01-01 15:50:36.294620,86782909
2016-01-01 15:50:36.294620,86782909
2016-01-01 16:02:14.628319,15527051
...
2016-01-01 17:14:29.360980,86782909
2016-01-01 17:17:27.849505,15461586
...

- You can assume that the transactions are sorted in ascending order of the Timestamps.
- From the log file we can trace a product’s transactions in a day. E.g., the six transactions of product 86782909 are highlighted as a running example. The first five transactions have the same timestamp, which means the customer bought these 5 items at the same time.

1. Create a shell script **analyze.sh** that do the following:
 - For each “7Eleven_00123_” log file, output item counts and the top three product IDs with the most amount of items sold in descending order of item counts.
 - You can assume that all the “7Eleven_00123_” log files are stored in the same directory of **analyze.sh**.
 - If the number of items were the same for two different product IDs, we output them in descending order of the product IDs. Therefore “9 81060085” is put ahead of ”9 78588321” under “7Eleven_00123_2016-01-01.log” in the example below.
 - If we run **analyze.sh** on the directory that consists of log files from 2016-01-01 to 2016-01-05 we provided, the following will output:

```
7Eleven_00123_2016-01-01.log:
  10 28468614
  9 81060085
  9 78588321
7Eleven_00123_2016-01-02.log:
  10 91743551
  10 55056411
  10 37290775
7Eleven_00123_2016-01-03.log:
  10 54293095
  10 45573708
  9 41965222
7Eleven_00123_2016-01-04.log:
  10 28445733
  9 82882899
  9 78096304
7Eleven_00123_2016-01-05.log:
  10 22184530
  9 93683090
  9 91743551
```

Hints:

- You can make use of the commands “sort” and “uniq” to get the number of occurrences of each product.
- For the “uniq” command, using the flag -c can return the row count.
- You can make use of the command “head -n x” to return the first x row.
- After you have finished the shell script, you are strongly suggested to create some other input data to test if you analyze.sh is correct. E.g., Updating 7Eleven_00123_2016-01-01.log to make 10 transactions for product 86782909, to see if you are able to output the correct result.

2. Suppose that besides the “7Eleven_00123_” log files, we also get the sales record of the store with ID ”00456”. The log files of the store 00456 have the same format as those

of store 00123. For instance, consider the 7Eleven_00456_2016-01-01.log file that consists of all the sales records made in the store 00456 on 2016-01-01:

```
...
2016-01-01 14:54:27.733836,57553901
2016-01-01 14:8:46.747141,86782909
2016-01-01 14:8:46.747141,86782909
2016-01-01 14:8:46.747141,86782909
2016-01-01 14:8:46.747141,86782909
2016-01-01 14:8:46.747141,86782909
2016-01-01 14:8:46.747141,86782909
2016-01-01 15:1:7.106278,90056244 ...
```

- You can assume that the transactions are sorted in ascending order of the Timestamps.
- You can also assume that the product ID is unique for each product across the log files of all 7-Eleven stores.
- From the log files we can trace a product's transactions in various stores. E.g., the twelve records of product 86782909 (Six in 7Eleven_00123_2016-01-01.log, and six in 7Eleven_00456_2016-01-01.log) are highlighted as a running example.

Create a shell script **trace.sh** that does the following:

- **trace.sh** takes one command line input argument. The input argument represents the ProductID that we would like to trace.
- You can assume that the “7Eleven_00123_” and “7Eleven_00456_” log files are all located in the same directory as trace.sh. If we run **trace.sh** as follows:

```
$ ./trace.sh 86782909
```

trace.sh generates a file 86782909.log that contains a list of sales records with productID 86782909 in all “7Eleven_00123_” and “7Eleven_00456_” log files, sorted in ascending order of the date and timestamp.

- As an example, 86782909.log consists of the following transactions:

```
2016-01-01 14:8:46.747141,86782909
2016-01-01 14:8:46.747141,86782909
2016-01-01 14:8:46.747141,86782909
2016-01-01 15:50:36.294620,86782909
2016-01-01 15:50:36.294620,86782909
2016-01-01 15:50:36.294620,86782909
2016-01-01 15:50:36.294620,86782909
```

We only show the seven transactions of product 86782909 in our running example.

- Note that **trace.sh** will only create the trace log but output nothing except the following cases:
 - If the number of input arguments is not 1, then **trace.sh** outputs “Usage: ./trace.sh (ProductID)” in shell prompt.
 - If there are no sales records found, then 86782909.log will be deleted. trace.sh will output “No records found for 86782909” in shell prompt.

- Please run **trace.sh** multiple times and check the correctness of the created log file.

–END–