McGill University
School of Computer Science

COMP-206

Software Systems

Due: January 31, 2017 on myCourses at 23:30
(two late days, -5% each day)

Systems & Operating Systems

Question 1: Short answer questions

1. In the course textbook, chapter 1, there is a discussion about the getChar assembler program. How does that story demonstrate the concepts behind software systems?
2. Google the "Linux Documentation Project" and look around the web site. Then Google "Linux Documentation Project Manifesto" and for this question, summarize the manifesto in one or two sentences.
3. Browse the http://linuxcounter.net web page. Identify the three top Linux using countries AND justify why you selected the countries you chose.
4. What are Vi, Vim, Emacs, and Pico?
5. Using Google identify which of these four tools Vi, Vim, Emacs, and Pico have an active online user community. We define an active user community as a community that has recently updated their software and/or have public discussion boards with recently posted information and questions. Identify an example active web community (or an inactive community for the tool if it is not active) by writing down one example URL.
6. Given the information in the URL of ONLY one active user community you identified above, what reasons do they give for supporting the tool?

Question 2: The Unix Command-line

For each question bellow write the Unix command that performs the requested action:

1. Finger all the currently logged in users who have the string "le" anywhere in their name and redirect the output into a file named fingeredUsers.txt. Do this as a single one-line Unix command expression.
2. Assuming you are in your home directory, write all the Unix command-line commands you would use to do the following: create a sub-directory called Source. The Source directory will have two additional sub-directories: Backup and Docs, create these as well. Assume your home directory has a bunch of .doc

and .java files. Write two commands executed on a single command-line that copies all the .doc files into the Docs directory and the .java files into the Source directory. Write all the above command-line commands without moving away from your home directory.

Question 3: The Login Script

Many Unix "power users" (users who actively use a particular technology to its fullest potential) will often customize their command-line interface to distinguish themselves from "regular users". They do this by using the computer's login script. This question asks you to customize your command-line interface using the Unix login script. To answer this question you will need to do four things: (1) identify the login script that is activated when you SSH/Putty into mimi.cs.mcgill.ca, (2) identify the login script that is activated when you login from the computers on the third floor of Trottier (they are not always the same), (3) decide if these two ways of accessing the CS Unix system use the same login scripts or not, and finally (4) write/modify the appropriate login script(s) for these two ways of accessing the CS Unix computers to customize your command-line in the following ways:

- Change you command-line prompt to some interesting phrase/word of your choice
- Change the command-line prompt to display your user name and your current working directory (this includes your interesting phrase/word)
- Create a short alias for "ls -l -a", it is up to you, but maybe "lsa"
- Execute the "who" command together with the "grep" command to identify whether your peer group (or any 3 or more people) are logged into the computer. (It is always good to know if your friends are around)
- If you are on a Unix computer with a GUI then the login script must launch the browser (any browser installed on that computer). The browser must automatically load up the URL you identified in Question 1.6.
- Your login script must append to the text file MyLoginHistory.txt the date and time you just logged in.

Hand in the login script(s) you created/modified to answer this question.

For The Glory

Every assignment will have a "for the glory" section. This is not for grades. If you can do the glory question then it is for bragging rights and peer respect.

Normally glory questions are programming questions but this one is connected to a question a student asked me a while back... "How can I become a good programmer?". I will attempt to give you some direction.

I have three challenges for you. If you can do these challenges then you will become a good programmer. Challenge #1: When writing your programs do not use the Internet. If you do want to use the Internet then use it only for syntax not for programming examples. A good programmer practices dreaming up code using their imagination, education, and syntax resources. Challenge #2: Do not look at each others code but DO discuss assignment questions with your peers. Understand together but build alone. Challenge #3: When I was between the ages of 18 and 21 my peer group developed a competition to help improve our programming skills that I'd like to share with you. This challenge works best as a friendly competition between a few friends. It is... count the number of compiles. The person, in your group, that finishes their assignment with the fewest compiles wins respect from the group. Then for each subsequent assignment you should attempt to lower the number of compiles. If you are interested I can provide strategies. At one point, my peer group were submitting assignments with as low as 3, 2 and sometimes 1 compile (if you can believe it).

The glory question: count the number of times it took you to run and re-run the login script before it was perfect. Compare that number with your peer group to see who did best. Then, remember that number because it is your upper bound for assignment #2!!

WHAT TO HAND IN

Everything must be submitted to My Courses before the due date. Remember that you can hand in your assignment up to two days late but there will be a penalty of 5% each day. After that, your assignment will not be accepted. Please hand in the following:

- For question 1: A text, word or PDF file named A1Q1 with the answers.
- For question 2: A text, word or PDF file named A1Q2 with the answers.
- For question 3: The login script and other file(s) you created/edited.

HOW IT WILL BE GRADED

The TA will use the following instructions when grading your assignment. TA's are often given additional instructions not presented here but compiled from common student questions or found problems in the assignment or adjustments to the assignment procedures.

- ASSIGNMENT: is worth a total of 20 points.
- QUESTION 1: is worth 6 points
  - 1 point for each question
- QUESTION 2: is worth 4 points.
  - 1 point for each question
- QUESTION 3: is worth 10 points.
  - Graded proportionally.

GRADING RULES

The following rules are followed by the TA when grading assignments:

- A program must run in order to get a grade (even if it does not run well). If it does not run (does not compile) it will receive a zero. (Make sure to run your programs from Trottier – they sometimes do not run the same from home when logging in using putty. Mimi is not the same server as the one you use in school.)
- All questions are graded proportionally (assuming the program runs at all). This means that if 40% of the question is correct, you will receive 40% of the grade.