CSCI 3215 (Advanced UNIX Command-Line Tools), Fall 2020

Homework 2

Credit: 40 points.

1 Reading

Be sure you have read, or at least skimmed, the assigned readings for 9/11, 09/14, and 09/16.

2 Problems

Answer the following questions. You may write out your answers by hand and scan them, or you may use a word processor or other program, but please submit a PDF or plain text via e-mail to my TMail address. (No links to shared files on Google Drive please.) Please use a subject line that mentions the course and the assignment (e.g., "csci 3215 hw 2" or "UNIX hw 2"). Answers to most questions will involve experimentation on a UNIX or Linux system. You are free to use any appropriate system (unless a specific problem says otherwise); if you use something other than one of our classroom/lab machines please tell me what.

Shell basics and customizations

(Answer the following questions for the `bash` shell.)

1. (5 points) When you type a command (e.g., `ls`), the shell has to find an executable (program) to run. Where does it look? How could you make it also look in your directory `MyPrograms` (in addition to wherever it looks now)? How could you make it look *only* in your directory `MyPrograms`?

2. (5 points) Give the command(s) you would use to define two aliases or shell functions: `delete` to move a file to be "deleted" to a temporary directory such as `$HOME/.trash`, and `undelete` to move a "deleted" file from the temporary directory to the current directory. (Examples of use: `delete myfile`, `undelete otherfile`.)

   (It's up to you to decide whether these should be aliases or shell functions. Hint: You might be constrained by the capabilities of aliases versus functions.)

   Note that commands to define aliases or functions can be entered from the command line, in which case they apply to the current session only, or can be included in an appropriate initialization file.

I/O redirection and pipes

1. (5 points) How would you invoke the `gcc` compiler if you want to be able to page through its output (both standard output and standard error) with `less`?

   How would you capture the error output only in a file called `gcc-ERRORS`?

2. (5 points) How could you make a one-line text file without using a text editor? Could you extend this idea to make a multiline text file? Tell me about as many ways to do this as you can think of.

Filter programs and other useful commands

1. (5 points) When a new user account for the department's lab machines is created, part of the setup procedure is to create a home directory in `/users` and copy into it certain files from `/etc/skel`. What command could you use to compare the files in your home directory to the files in `/etc/skel` (including any files in subdirectories), for example to find out whether you had inadvertently changed or deleted something that might be important? (Hint: You can do this with one command.)

2. (5 points) What would you type at the command line to find all files in your home directory (and all subdirectories) that have been modified within the past 24 hours and end with `.c`?

3. (5 points) What would you type at the command line to get a sorted list, with no duplicates, of all the users running processes on the machine you're using, along with a count of processes they're running?

4. (5 points) Answer/do one of the following:

   (a) What would you type at the command line to find out how many processes are being run by user `root`?

   (b) Describe something you actually want to do (e.g., archive all files that have been changed in the last 24 hours, or find a classroom machine that's up and connect to it with `ssh`) and a solution involving a pipe and at least one of the commands mentioned in the reading for this assignment.

3 Pledge

Include the Honor Code pledge or just the word "pledged", plus *at least one of the following* about collaboration and help (as many as apply).[1] Text *in italics* is explanatory or something for you to fill in. For programming assignments, this should go in the body of the e-mail or in a plain-text file `pledge.txt` (no word-processor files please).

- This assignment is entirely my own work. *(Here, "entirely my own work" means that it's your own work except for anything you got from the assignment itself — some programming assignments include "starter code", for example — or from the course Web site. In particular, for programming assignments you can copy freely from anything on the "sample programs page".)*

- I worked with *names of other students* on this assignment.

- I got help with this assignment from *source of help — ACM tutoring, another student in the course, the instructor, etc. (Here, "help" means significant help, beyond a little assistance with tools or compiler errors.)*

- I got help from *outside source — a book other than the textbook (give title and author), a Web site (give its URL), etc.. (Here too, you only need to mention significant help — you don't need to tell me that you looked up an error message on the Web, but if you found an algorithm or a code sketch, tell me about that.)*

- I provided help to *names of students* on this assignment. *(And here too, you only need to tell me about significant help.)*

[1] Credit where credit is due: I based the wording of this list on a posting to a SIGCSE mailing list. SIGCSE is the ACM's Special Interest Group on CS Education.

4 Essay

Include a brief essay (a sentence or two is fine, though you can write as much as you like) telling me what if anything you think you learned from the assignment, and what if anything you found found interesting, difficult, or otherwise noteworthy. For programming assignments, it should go in the body of the e-mail or in a plain-text file `essay.txt` (no word-processor files please).