Assignment 2

CAAM 519 Fall 2021

Due Friday October 15, 10:59 am

In this assignment I may ask you to do things that you haven't seen in class (for example, change the date in the title of a IATEX document). Remember: there is a wealth of useful (R) information that is readily Google-able!

1. Setting up Github

- (a) (0 points) Create a Github account: https://github.com/
- (b) (0 points) Register for the Github Student Developer Pack: https://education.github. com/pack: this gives you Github Pro for free, which will allow you to add the instructor and graders as collaborators on your private repository.
- (c) (0 points) Create a repository named "caam-519-submissions". Then add me and the graders as collaborators. On the repo main page, go to Settings \rightarrow Collaborators, and then add msarrafj, philipwxh, and AlexBalsells as collaborators.

2. Shell scripting

- (a) (15 points) Write a shell script named clean-build.sh that does the following: It takes as input a project name, project (without a file extension). The script should then:
 - 1. Make a hidden directory in the current directory named .build. Your code should not print a warning message if that .build directory already exists.
 - 2. Compiles the project.tex file using pdflatex in such a way that all the auxiliary build files are contained solely in the .build directory, but the project.pdf is contained in original directory upon completion. Commit clean-build.sh to caam-519-submissions/homework-2
- (b) (25 points) Create a shell script called linecount.sh that takes zero or more paths as parameters and reports the total number of lines of all files in all of the paths. If no parameters are given, then linecount.sh should output 0 and exit.

For each path that is a directory, the lines of all of the files in that directory or below it in the file system should be counted. (That is, if foo, foo/bar, foo/qux, and foo/bar/asdf are all directories and foo is passed as a parameter, then the output of

linecount.sh should include the lines of all files in those directories as well.)

Write a function that iterates over each item in a directory and adds the line count to a running total for each file and recurses into each directory. You may not use find.

Make sure your implementation correctly handles file names with spaces, files and directories that start with a period, and only considers normal files and directories (e.g., not symbolic links nor device files).

Report any unreadable files or unreadable/unsearchable directories on stderr (see examples below) and continue on. (The order of the error messages doesn't matter and yours may not match the order in the examples.)

The shell parameter **\$0** expands to the path to the script (e.g., ./linecount.sh) which is useful in error messages. Example outputs (the numbers are made up and just for example purposes)

```
> ./linecount.sh
0
> ./linecount.sh .
97
> ./linecount.sh "$HOME" /etc
./linecount: /etc/cups/ssl: Permission denied
./linecount: /etc/chatscripts: Permission denied
./linecount: /etc/sudoers.d/README: Permission denied
./linecount: /etc/shadow: Permission denied
# A bunch more errors removed from the example
339123
```

- (c) Download **poetry.txt** from the class website and achieve the following tasks. (The file contains favorite poems collected from student 1-12.)
 - i. (10 points) Create a shell script called **Poetry_table.sh** that takes **poetry.txt** as an argument and creates a table similar to this form (that contains all 12 students info):

```
Student_IDFavorite poemFirst_7_chars_of_poemStudent1Of that colossal Wreck ...Of thatStudent2At night they come ...At NighStudent3NANA....
```

- poetry.txt contents should not be modified at any time
- columns could be delimited with either [tab] or [space]
- third column represents the starting 7 characters of each poem
- if data for a Student is not available print NA
- awk, cut, paste and others are useful here (hint)
- ii. (10 points) Modify your Poetry_table.sh script such that (only) if a -h flag (option) has been passed into a shell script, then a help message is displayed:

```
> ./Poetry.sh -h
Poetry.sh program generates a table that contains ...
```

iii. (10 points (bonus)) Write a program Poet_finder.sh that takes the favorite poem of each student as an argument and spits out the name of its poet. The script needs to search the given poem in "https://poets.org/" website and find the corresponding url. Then get access to content of the webpage. Then use awk (or other commands) to filter name of the poet and pass it to stdout. You may use any external command (e.g. w3m, etc) to achieve the task. Output should look similar to this:

```
> ./poet_finder.sh Of that colossal Wreck
```

The poet is: Percy Bysshe

- iv. (5 points (bonus)) If you succeeded in the last problem. Now modify Poetry_table.sh to include the name of poet (of each favorite poem) as the last column.
- 3. (30 points) **Create a latex file:** Write a document named latex-submission.tex using LATEX. The document should have a title that reads: "CAAM 519, Homework #2: LATEX Submission" (with the pretty-printed latex symbol (i.e. LATEX)). The author should be your NetID, formatted in a monospaced font (hint: there's a command whose first four letters are "text" that will accomplish this.). And the date should be the date I am preparing this homework: September 30, 2021.

When you typeset your document, you may notice a missing space in the title. Find a way to force a (normal-sized) space between " LAT_FX " and "submission."

The contents of the document should be as follows. The first section will be entitled "Communicating with remote machines via ssh", and should contain a **verbatim** environment block that includes four distinct pieces:

- 1. the ssh command you used to log into the clear machine,
- 2. the message that the clear machine provided upon logging in, and
- 3. the output of the following shell commands: echo \$HOSTNAME.
- 4. Also command you use to change your shell prompt text to > In my Ubuntu running in VirtualBox, by default it says something like mo@sarraf-VirtualBox:\$.

In the second section, entitled "A script to build a latex document while hiding auxiliary files", use the listings (or minted) package to include a pretty-printed copy of your clean-build.sh code. In the body of the document, briefly describe the code line-by-line. Use the basicstyle= option for the lstlisting environment, and configure the listing for the bash language (this is another option you can set). Provide a caption describing the listing.

After this block, I want you to critique your code. Identify and briefly describe two ways that the code that you have written could lead to unexpected or undesirable behavior. In particular, imagine ways that a user could attempt to use your code in a way that is seemingly reasonable, but will nonetheless fail to produce the desired result.

Commit and push the following to caam-519-submissions/homework-2 which was created in the first part.

- 1. latex-submission.tex latex file only (no pdf output or auxiliary files. You may need to use .gitignore file)
- 2. clean-build.sh, linecount.sh, Poetry_table.sh (and maybe Poet_finder.sh) bash scripts
- 4. (0 points) **Finalizing:** Once your assignment is completed, **tag** the last commit with the homework-2 tag. Note that this references the exact version that we will grade; if you end up making changes (before the deadline, of course), make sure you delete the tag, and re-tag the latest commit!

Also, you should only have six files (including .gitignore) included in your tagged submission! You shouldn't be submitting, for example, the latex auxiliary logging files. 5. (0 points) Fill out this form so that we can track down your repository.

Note that your scripts and commands should be clearly commented, explaining each part of it. If it is not commented, you will receive no credit.