

Kathleen Anderson, Yizhen Wu, Ziyang Zhang, Jihoon Jang

Professor Szafir

INFO3401

11 February 2020

Problem Set 2

Part One: Shell Commands

1. Assume all of the directories below are valid directories (e.g., they exist and we have access to them). What would the following print out?

```
cd /usr/INF03401/./homework/./problem1
```

```
pwd
```

‘cd /usr/INF03401’ would change the directory to the INF03401 directory. This directory would be stored under the "usr" directory. So the path here will be: root -->usr -->INF03401. And the ‘.’ will put you back to the upper two directories, before we are going to ‘homework’ directory which is under the ‘usr’ directory.

The "tree" path is: root → usr → INF03401.

Next, the "../homework/" section is going to bring you back to the parent directory, "usr" (due to the two dots, which sends you back up two directories) before sending us to a different directory within "usr" called "homework."

The tree path is = root →usr →homework.

The "./problem1" section will send us to a directory called "problem1" that's located within the "homework" directory.

(The "tree" path would go root → usr → homework → problem1). Finally, the "pwd" would print the working directory, which is "/usr/homework/problem1".

2. What set of commands would you use to achieve the following:

You've found yourself in a situation where your python program is trying to parse files in a directory and is throwing an error. You think it might be a permissions issue (i.e., you don't have permission to access the files in that directory) and need to verify the permissions on all files in that directory. The directory is in your home directory under the "datasets/activedata" directory. Make no assumptions about what directory you are currently working in.

cd ~ /datasets/activedata

pwd

Use sudo cd ~ /datasets/activedata

or

cd ~

cd datasets

ls -l -activedata

pwd

3. Briefly describe what the following set of commands would achieve. What process would happen and what would be printed to the command line?

cd ~ -- Change to the home directory

mkdir ./problem_set_1 -- Make a new directory named ‘problem_set_1’ in the ./ portion. At this point there is nothing printed out.

touch submission.txt -- create a blank text file named submission.txt in the home directory. The command line won’t print anything.

cd .. -- Go to the parent directory, which is the ‘users’ directory. The command line won’t print anything.

pwd -- print the working directory.

First, change our directory to the home directory. And create a new directory named “problem_set_1”. Then, create an empty txt file named ‘submisson.txt’ inside of the ‘problem_set_1’ directory. And go back to our parent directory, print the working directory. So it brings us to the ‘users’ directory.

4. What set of commands would you use to achieve the following:

Copy a document called config.txt from your home directory to the root directory. Then, create a new directory in the root called preferences. Within preferences, copy prefs.txt from a matching directory called preferences in your home directory. Finally, determine that the start and end of the config.txt document in both root and home match.

cd / --> change your directory to the root directory

cp ~/config.csv → Start at home directory: copy the config.txt to the root directory.

mkdir /preferences → In the root directory, making a directory named 'preferences'.

cp ~/preferences/prefs.txt → Copying prefs.txt from the root directory to the preferences directory.

head /config.txt → Find out the start of the config.txt in the root directory.

tail /config.txt → Find out the end of the config.txt in the root directory.

tail ~/config.txt → Find out the end of the config.txt in the home directory.

head ~/config.txt → Find out the start of the config.txt in the home directory.

5. You have accidentally moved a file (~/Documents/datafile.csv) to your current directory rather than copying it. What command did you use to do that? What command should you have used to do that? What command might you use to undo it?

If a file was accidentally moved, the person might have used the "mv" command to move it to the current directory. → mv ~/Documents/datafile.csv ./datafile.csv

If not that, the person could've used a "." to move it to the current directory: mv ~/Documents/datafile.csv .

- The command the person should have used to copy a file is "cp". --> cp ~/Documents/datafile.csv ./datafile.csv

If not that, then cp ~/Documents/datafile.csv .

**To undo it the person could re-copy it OR move it back to the ~/Documents directory.
→ cp ./datafile.csv ~/Documents/datafile.csv**

6. Download and unzip [asciify-master.zip](#) from Canvas. Then, navigate to the directory and use the command line to run asciify.py. What does this script do? Include a screenshot of your results.

This script opened this file and there is a cute cat?.

[illegible]

[illegible]

7. Install the Delorean Python package. What command did you use? Then, from the command line, launch Python. Use it to execute the following commands:

```
from delorean import Delorean
EST = "US/Eastern"
d = Delorean(timezone=EST)
```

```
print(d)
```

Use **pip install Delorean** to install the Delorean Python package.

What prints to the command line?

```
Delorean(datetime=datetime.datetime(2020, 2, 10, 20, 53, 42, 734801),  
timezone='US/Eastern')
```

8. We can scrape webpage content or download collections of data files using wget. NASA, the USDA, and the NCBI all recommend using wget to download data from their repositories because it is faster and allows you to download data in bulk. We'll experiment with that on a smaller scale: use the command line to download data on the World Bank's budgetary expenditures at

<https://finances.worldbank.org/api/views/yu93-ayrw/rows.csv?accessType=DOWNLOAD>

[\(Links to an external site.\)](#)

[Links to an external site.](#)

. What command did you use? Make sure to include this file as part of your submission.

Use wget

<https://finances.worldbank.org/api/views/yu93-ayrw/rows.csv?accessType=DOWNLOAD>

9. The above budgetary data contains both individual items and aggregated totals across different sectors. Let's distill the data down to different aggregated totals. In this dataset, you can use the keyword "Total" to identify only those rows containing aggregate budget items. Use the command line to identify these rows. What command did you use? How many rows did this find (hint: the -c option can be helpful here)?

Use **grep Total World_Bank_Expenditures_by_Organizational_Unit.csv** to identify these rows. It prints out random lines that include 'Total' inside.

Use **grep Total World_Bank_Expenditures_by_Organizational_Unit.csv -c** to count the total number of lines that have Total.

There are 33 lines.

10. Now, print the lines containing the word "Total" to a file to "distilledExpenditures.csv". What command did you use? Include this file as part of your submission.

grep Total World_Bank_Expenditures_by_Organizational_Unit.csv > distilledExpenditures.csv