Exercises:

July 4, 2020

1 Shell

Taken from `https://missing.csail.mit.edu/2020/course-shell/`, where you will also find basic instructions in how to manoeuvre in the terminal

1.1 Create a new directory called missing under /tmp.

1.2 Look up the touch program. The man program is your friend.

1.3 Use touch to create a new file called semester in missing.

1.4 Write the following into that file, one line at a time:

```
#!/bin/sh
xeyes
```

The first line might be tricky to get working. It's helpful to know that # starts a comment in Bash, and ! has a special meaning even within double-quoted (") strings. Bash treats single-quoted strings (') differently: they will do the trick in this case. See the Bash quoting manual page for more information.

1.5 File execution

Try to execute the file, i.e. type the path to the script (./semester) into your shell and press enter. Understand why it doesn't work by consulting the output of ls (hint: look at the permission bits of the file).

1.6 Using the sh intepreter

Run the command by explicitly starting the sh interpreter, and giving it the file semester as the first argument, i.e. sh semester. Why does this work, while ./semester didn't?

1.7 Look up the chmod program (e.g. use man chmod).

1.8 Changing file permissions

Use chmod to make it possible to run the command ./semester rather than having to type sh semester. How does your shell know that the file is supposed to be interpreted using sh? See this page (`https://en.wikipedia.org/wiki/Shebang_(Unix)`) on the shebang line for more information.

2 Shell tools

Taken from `https://missing.csail.mit.edu/2020/shell-tools/`, where you can find information about scripting

2.1 Read man ls and write an ls command that lists files in the following manner:

- Includes all files, including hidden files
- Sizes are listed in human readable format (e.g. 454M instead of 454279954)
- Files are ordered by recency
- Output is colorized

A sample output would look like this:

```
-rw-r--r--   1 user group 1.1M Jan 14 09:53 baz
drwxr-xr-x   5 user group  160 Jan 14 09:53 .
-rw-r--r--   1 user group  514 Jan 14 06:42 bar
-rw-r--r--   1 user group 106M Jan 13 12:12 foo
drwx------+ 47 user group 1.5K Jan 12 18:08 ..
```

2.2 Use | and > to write the "last modified" date output by semester into a file called last-modified.txt in your home directory.

3 Extra exercises

3.1 Create the following directory structure in the "answers" directory.

(First, create an 'answers' directory in "missing")

```
/answers
| -> number1
| -> number2
     | -> number2.1
     | -> number2.2
| -> number3
     | -> number3.1
          | -> number3.1.1
          | -> number3.1.2
     | -> number3.2
     | -> number3.3
```

You should be able to change into any of these directories afterwards. When you're done, you should be able to type in commands (like pwd) and have a session roughly as follows.

Output

```
[root@kali answers]$ pwd
/tmp/missing/answers
[root@kali answers]$ cd number3/number3.1/number3.1.1
[root@kali number3.1.1]$ cd number3/number3.1/number3.1.1
[root@kali number3.1.1]$ pwd
/tmp/missing/answers/number3/number3.1/number3.1.1
```

3.2 Copying files

Run the following commands in the missing directory:

```
touch file1.txt file2.txt file3.txt file4.txt
```

Copy the files from the "missing" directory to the "number3.1.1" directory. This can be done using a single command.

You should be able to see the files in the "number3.1.1" directory afterwards.

Expected results:

```
[root@kali answers]$ pwd
/tmp/missing/answers/number3/number3.1/number3.1.1
[root@kali answers]$ ls
file1.txt file2.txt file3.txt file4.txt
```

3.3 Using a text editor from the commandline

Navigate to the answers folder and edit the file1.txt so it now contains the text "I used a text editor". Do this by either using the commandline tool "vim" or "nano".

Nano is simple to use, though not as feature rich.

Vim can be more daunting to learn, though comes with more features. Quit by typing ":q!"

3.4 Bonus exercise: Basic data wrangling

Determine how many words contain an "e" in the file following file "/usr/share/-dict/words".
Hint: The tools, "cat", "grep", and "wc" are your friends.