


Week 06 Homework

New stuff we learned this week:

- lots of commands take a *variable* (or *variadic*) number of arguments, which is usually represented in the `man` page as a `...` — that means you can do something like `mkdir foo bar baz` and get *three* new directories
- the `cat` command actually *concatenates*, that is, it hooks together text from files, sending them all to standard out, after concatenating them 
- the `head` command reads a fixed number of *lines* (or bytes, if you're feeling cheeky) from a file and sends it to standard out
- the `tail` command works like the `head` command, but from the end, and it has a cool flag `--follow` that allows you to watch as stuff gets *appended* to the file 🐼🐼
- the **HTTP** protocol is a plain-text based protocol built on top of **TCP** — you can even use your meaty fingers to talk HTTP using `nc`

Homework plan:

- 2 days CLI practice
- 2 days `vim` practice
- 2 days touch typing practice
- watch CCCS#5 — *definitely TWICE*

Touch Typing Links:

- <http://touchtype.co>
- <https://www.how-to-type.com>

Homework day 1

- `vimtutor` — Everything *except* Lesson 5 and Lesson 7
- touch typing practice

Homework day 2

- [CCCS #5](#)
- CLI Practice #1

Homework day 3

- `vimtutor` — Everything *except* Lesson 5 and Lesson 7 (yes, same as day 1)
- touch typing practice

Homework day 4

- CCCS #5
 - CLI Practice #2
-

CLI Practice #1

1. `ssh` into your home dir
2. delete everything in your home dir *except* the `moby.txt` file, but try to do it efficiently. `rm` is like most unix-y commands, it takes *variadic* arguments, so you can `rm` many things at once, like `rm foo.txt kitty.jpg`. Try to take advantage of this fact when cleaning up. ✨ **Extra credit:** see if you can also use *shell expansions* to be even more concise in your cleaning efforts.
3. Now, in your tidy home dir, in one command, using `touch`, create 3 empty files: `beep.txt`, `boop.txt`, and `zoink.txt`
4. using `echo` and a redirect, add content to each file containing one line of robot-sound text: `BEEP`, `BOOP`, `ZOINK`. (you can't do this in one command, you'll need to do it three times)
5. make a new directory called `sounds` in your home dir
6. in one command, move the three new files you created in step 3 into the `sounds/` dir, using either variable arguments or shell expansions (but beware of the `moby.txt` file — try not to move it)
7. change your location into the `sounds/` dir
8. use `cat` and a *shell expansion* to print the contents of all three files in this dir to standard out
9. now, use `cat` again (without a shell expansion) to print out the contents of all three files in the *reverse order*
10. next, do the same thing you did in step 9, but redirect the concatenated output into a file called `reverse-sounds.md` — but **don't re-type the command from step 9, use the up arrow** and then add the redirect (`.md` is just another plain-text file extension, don't worry about what it means, it will just simplify future steps to have these *not* end with `.txt`)
11. using the up arrow again, bring up the command you used to complete step 8, and add a redirect to the end of the command so that you redirect the concatenated output into a file called `sounds.md`
12. use the `mv` command and a shell expansion to (in one command) move both `sounds.md` and `reverse-sounds.md` up one level into your home dir
13. change directories back to your home directory, but using a command that only contains **TWO LETTERS**
14. delete the `sounds/` dir and all it contains in one command

15. use `cat` to concatenate both `sounds.md` and `reverse-sounds.md` and send the output to standard out
16. (using the up arrow to not retype) repeat the last command, but this time filter out all the lines that don't contain `ZOINK` by *piping* the std out of `cat` into the std in of `grep`, like so: `<your-cat-command-with-arguments> | grep <pattern>`
17. (using the up arrow again to not retype) repeat the last command, but this time, add a redirect after that sends the lines containing `ZOINK` into a file called `zoinks.txt`
18. type a command that will print out the **first** 10 lines of Moby Dick (in the `moby.txt` file)
19. type a command that will print out the **first** 100 lines of Moby Dick
20. type a command that will print out the **last** 50 lines of Moby Dick

CLI Practice #2

1. `ssh` into your home dir
2. without using `vim` or `echo` create a file called `where-am-i.txt` that contains the full path to your *current working directory*
3. without using `vim` or `echo` create a file called `whats-here.txt` that contains a listing of all of the files and directories in your home directory
4. using `cat` and a redirect, combine the contents of both of these files into a new file called `command-output.txt` (**🌟 Extra credit:** use shell expansion to not type out both filenames)
5. (**🌟 Extra credit step:** type a command that will write *only your last five shell commands* onto the end of the `command-output.txt` file — if you're stumped, complete the lesson and then come back, you might find it easier)
6. take a minute to skim through the `man` page for `head` and `tail` to remind yourself how they work.
7. type a command that will print out the **first** 25 lines of Moby Dick
8. type a command that will print out the **last** 65 lines of Moby Dick
9. use `head` AND `tail` (piped together with a `|`) to print to standard out lines `495-505` of Moby Dick. (Hint, you'll never need to type `495` — rather you'll use a tiny bit of super simple Math)
10. (using the up arrow to not re-type) repeat the last command, but this time, send the output to a file called `paragraph.txt`
11. Repeat step 9, but play around with some other random numbers until you find a different chunk of text that seems cool, and then share your command in Slack.
12. search the `man` pages for some of the other commands you know, looking for another one that uses *variable* or *variadic* arguments (indicated by the `...` in the man page)
13. Finally, use `vim` to create a file called `jared.txt` and inside it, ask me a question. Save and close the file.
14. Slack me and see if I'm available
15. If I'm available, review the `man` page for `tail` if necessary, and then type a command that will allow you to watch in realtime while I edit your `jared.txt` file, adding lines of text under what you added. When you're ready, let me know on Slack, and I'll edit your file, answering your question and adding new lines so you can use this feature of the `tail` command.

