

Week 05 Homework


New stuff we learned this week:

- shell *expansions* `*`, `?`, and `[...]`
- `*` means *any number of any character*
- `?` means *exactly ONE of any character*
- `[...]` means *exactly ONE of the characters inside the brackets*
- **TCP** — a network *protocol* where every *packet* is *acknowledged* with an `ACK`
- **UDP** — a network protocol without `ACK`s, used for streaming video, audio, gaming
- computers can play different *ROLES*, and a very common set of roles is *CLIENT* and *SERVER*

Homework plan:

- 2 days CLI practice
 - 2 days `vim` practice
 - 2 days touch typing practice
 - watch CCCS#4 (twice, ideally)
-



Homework day 1

- `vimtutor` all of Lesson 3, Lesson 4, and Lesson 6
- touch typing practice — you really got to do this guys! remember neural nets! 
<https://www.how-to-type.com/https://www.how-to-type.com/>
<http://touchtype.co/>
- CLI practice #1

Homework day 2

- CLI practice #2
- [CCCS #4](#)

Homework day 3

- touch typing practice  
- `vimtutor` — all of Lesson 2, Lesson 4, and Lesson 6 (yup, lessons 4 and 6 are repeated!)

Homework day 4

- play `time cat`, but *no more than 10 times*. send *BOTH* of the signals we talked about in week 3 (graceful exit, or `EOF`, and hard kill)
- [CCCS #4](#)
- do a short (5 minute max) TCP server/client session with someone else from the class. To *listen* (that's the `-l` part) as the server, do `nc -lp <SOMEPORT>`. To connect as a *client*, do `nc localhost <SOMEPORT>`. We'll talk more about *ports* at another time, but for now you should know that port numbers only go up to about 64000, and many of the lower ports are reserved, so I would use a port higher than 1000 and lower than 64000.

CLI Practice #1

1. `ssh` into your home dir
2. delete the whole directory structure `jared/is/cool/and/smart/` left over from last week (hint: remember the silly pronunciation “*rimraf*” to help you remember the command + flags combo)
3. use `touch` or `echo` plus a redirect to make 6 files right in your home dir: `beep.txt`, `boop.txt`, `dog.html`, `dog.css`, `cat.html`, `cat.css`
4. using the `*` star operator, write a *shell expansion* that would expand to include both `beep.txt` and `boop.txt` — test your expansion using the `echo` command
5. next, use the `?` operator to write an expansion that would select the *same two files* (test it with `echo`)
6. third, use the `[<letter><letter>]` syntax to write another expansion that would select the same two files again (hint, you may need to use the brackets more than once!) (test it with `echo`)
7. still from your homedir, make a new directory called `robot-sounds`
8. now, move both `beep.txt` and `boop.txt` into the `robot-sounds` directory, but using a shell expansion so that you don't have to type both file names
9. now make a set of directories (with one command): `animals/dog`
10. change your working directory so that you are inside `~/animals/dog`
11. now, from the `dog/` dir, use a combination of 1) relative paths, 2) shell expansions, and 3) the `mv` command to move both `dog.css` and `dog.html` into your current directory
12. now, again without leaving the `dog/` directory, make a new folder inside of the `animals/` dir (which is one above you) called `kitty`
13. still in the `dog/` dir, use 1) shell expansion, 2) the special shortcut for your home dir, and 3) the `mv` command to move both `cat.html` and `cat.css` from your home dir into the `animals/kitty/` dir.
14. move over into the `kitty/` dir
15. print your current working directory to standard out
16. **extra credit:** *without* using `vim` or `echo` create a file in your current working directory called `thisdir.txt` that contains the full path to the `kitty/` dir. 🤔

CLI Practice #2

1. `ssh` into your home dir
2. `cd` into the root directory of the whole computer (but don't you dare *rm -rf* from in there!)
3. remind yourself how to send a *signal* to halt a program, then type `tree .` to start listing out the whole filesystem, after a few seconds, interrupt `tree` and make it stop by sending a signal
4. now jump back to your home dir by typing only 2 letters
5. from your home dir, make two new directories `html` and `css` — but try to do it with only one command (meaning, use `mkdir` only once) (hint, this doesn't have to do with shell expansion)
6. use the `tree` command to print out all of your directories and files
7. now, with the info that `tree` gave you to jog your memory, stay in your home dir, and move both the `dog.css` and `cat.css` into your new `css` directory (you can use multiple commands here, and don't try to use shell expansion for this 🙌)
8. similarly, move `dog.html` and `cat.html` into your `html/` dir.
9. now `cd` into the `html/` dir
10. from the `html/` dir, use the `*` shell expansion character and the `mv` command (and relative paths) to move both `css` files from the `../css` dir into the `html/` dir
11. now you should have 4 files in your `html/` dir.
12. from the `html/` dir, delete the now empty directory that used to have the `.css` files
13. jump back up into your home dir
14. rename the `html/` dir so it is named `code`
15. now, using shell expansion, delete both of the files that end with `.css` in the `code/` dir, but use only the `?` character for your expansion, not the `*` char
16. using `echo` and a redirect character (not vim), create a new file in your home dir called `time.txt` that has one line of text: `The time is:`
17. now, try typing `date` and see how it sends the current date/time to standard out
18. now, without blowing away the text that you already put into `time.txt`, *append* the output of the `date` command to the `time.txt` file
19. repeat the last step a few times
20. now, append one more line (without using vim) that reads: `That's all folks` to the `time.txt` file
21. type a command that will barf out the contents of `time.txt` to standard out (you should see `The time is:` followed by a few lines of date strings, followed by `That's all folks`)
22. now, use `vim` to create another new file called `lime.txt` — put a few words in it and save it
23. now, see if you can figure out how to make a backup of your `time.txt` file called `backup.txt` that contains all of the exact same text. (hint, you don't need any new commands for this, you can do it by combining something that barfs to standard out with a redirect)
24. finally, use a shell expansion with only the `[.]` syntax to *delete* both `lime.txt` and `time.txt` with a single command (so, no using `?` or `*`)
25. **super extra credit:** *in one command*, create a file called `remove.txt` that contains a bunch of lines showing exactly all of the commands you ran recently using the `rm` command. you won't need `echo` or `vim` for this, it can be done in a single command by typing only around 30 characters. 🤔🤔

