Week 07 Homework 💻 🦖

New stuff learned this week:

- the `cp` command works almost exactly like `mv` except it *copies* (thus creating new files)
- with `cp` the `-r` (recursive) flag works to copy all the contents of a directory, like `cp -r dir-of-goodies/ other-dir/`
- prefix a command with `sudo` to invoke the command as a *super user*. It stands for Super User DO.
- HTTP *Requests* are composed of
  - Method (`GET` `POST` `DELETE` …etc.)
  - Path (like `/` for “root” of website)
  - protocol version `HTTP/1.1`
  - a bag of headers as Key/Value pairs `SomeKey: Awesome Value`
  - optionally some *content* (but no content when using the `GET` *method*)
- HTTP *Responses* are composed of
  - protocol version, like `HTTP/1.1`
  - status *code*: `404`
  - status *message*: `Not Found`
  - bag of headers as Key/Value pairs
  - almost always some *content* (very often *html code)*

Touch Typing Links:

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

Homework plan:

□ 1 day creating flash cards
□ 2 days CLI practice
□ 1 day `vim` practice
□ 2 days touch-typing practice
□ watch CCCS #6 — definitely twice

Homework day 1:

- do flashcard assignment (see below)
- touch typing practice

Homework day 2:

- CLI practice #1
- watch CCCS#6

Homework day 3:

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

Homework day 4:

- CLI practice #2
- watch CCCS#6

Flashcard Assignment:

Use the flashcards I handed out to make simple flash cards for all of the basic unix-y shell commands we've been learning. Put the command on one side, and what it does (in your own words) on the other side. Include any flags on the back side that you think are interesting or important. Be sure to include all of these commands:

- pwd
- ls
- cd
- rm
- mkdir
- history
- man
- touch
- mv
- cat
- echo
- head
- tail
- cp

Also, make cards for these few key concepts/operators:

- `~`
- `>`

- `>>`
- `|`
- `.`
- `..`

CLI Practice #1

1. Go carefully read and review the “new stuff we learned this week” at the top of this document.
2. `ssh` into your home dir
3. make a new directory called `week7` and change your location into it
4. redirect the output of the `pwd` command into a new file called `pwd.txt`
5. type a command that will put the entire contents of the `man` page for `ls` into another file called `ls-man.txt`
6. type a single command that will create a file called `mkdir.txt` that contains only the first ***5*** lines of the `man` page for `mkdir`
7. now, imagine you really need to back-up those three critical files you just made. Use the `cp` command 3 times to create a copy for each file named `pwd.backup.txt` , `ls-man.backup.txt` , and `mkdir.backup.txt`
8. type a few commands that will allow you to verify (by barfing to standard out) that the contents of each file is the same as it’s backup
9. make a new directory called `backups/` inside of `week7/`
10. in one command (with<u>out</u> using shell expansion), *copy* all of the backup files into the `backups/` dir
11. type a few commands to see that the backup files are now in TWO places: the `week7` dir, and the `backups` dir
12. now, (you should still be in the `week7/` dir) use `rm` plus a shell expansion to delete all of the backup files in your current working dir.
13. still in the `week7/` dir, make a new directory called `double-backups` and then use `cp` and a flag to copy the entire directory of the `backups` dir into the `double-backups` dir you just created.
14. type a single command that will combine the contents of the three files remaining in the `week7/` dir (`pwd.txt` `ls-man.txt` and `mkdir.txt`) into a SINGLE file called `combined.txt`
15. move yourself down into the `backups/` dir and *from there*, use **relative paths** and the `cp` command to copy the `combined.txt` file you just made into the `backups/` dir with the new name of `combined.backup.txt`
16. make a file called `<yourname>-was-here.txt` in each of the home directories of everyone else in the class. Put some text in it as a message, either with `vim` or using a redirect. Hint: you’ll have to use one of the new concepts we went over this week.
17. Check and see if anyone left you a message yet in your home dir

----

CLI Homework #2

1. `ssh` into your home dir
2. check for any new messages from your friends in the class from step 16 of CLI practice #1 — use `cat` to see what they say.
3. make sure that your home dir does *not* contain the file `moby.txt` from last weeks homework, delete it if it's still there.
4. one directory *above* your home dir, I put a new file called `moby-protected.txt` — stay in your home dir, and type a command to look into that dir so you can see the file.
5. ***from your home dir***, type a command that will copy that file *down into the* `week7/` dir. The file is "protected" so you'll need to use your super powers every time you interact with this file
6. change your location down into the `week7/` dir
7. type a command that will allow you to see the first 25 lines of `moby-protected.txt`
8. without using `vim` or `echo` , type a command to create a blank file called `start-end.txt` (review your flashcards if you can't remember the command)
9. in **three** commands, *using redirects*, make it so the file `start-end.txt` ends up containing the following contents: the first 15 lines of moby dick, followed by the sentence "a bunch of stuff about whales here", followed by the last 15 lines of moby dick.
10. `cat` out your `start-end.txt` file to make sure you did it right.
11. move into the user `ubuntu`'s home directory, and leave me a message, by creating and typing a short file with `vim`
12. now, from `ubuntu` home dir, move all the way back into *your* `week7/` dir using **an absolute path**. If you're stuck figuring that out, ask yourself what absolute paths always start with, and use the `pwd` command to help you orient yourself (`pwd` always prints out an *absolute* path)
13. make a directory inside of `week7/` called `files` with two directories inside of it named : `da` and `misc` so that you end up with `~/week7/files/da` and `~/week7/files/misc` (**Extra Credit** ✨: do this in a single command)
14. now, in a single command (using *variadic arguments*) create three new blank files called (take note of the file extensions!!!) `dog.md`, `dug.md`, and `dag.md`
15. use a form of shell expansion to (in one command) to move `dog.md` and `dug.md` ONLY into the `files/misc` directory
16. now, move the remaining `dag.md` file into `files/da`
17. using either relative or absolute paths, move up into *the root directory of the whole computer*
18. remind yourself how to send a kill signal, and then type `tree .` to start listing out the full file structure of the whole computer. After a few seconds, interrupt the program by sending a signal.
19. now, type a command that will print out just the first 5 lines of what `tree .` would normally print out.
20. use the up arrow to recall your last command, but this time, add something to the command to redirect the output into a file *in your home dir* called `small-tree.txt` — use the single-character shortcut for your home dir when doing this one.
21. now, type a command that will allow you to see all of the commands you've run recently. Find the one when you `cd`-ed into your `week7/` dir using an absolute path (step 12 above). Repeat this command by typing `!<command-line-number>` — so if the command appears on the line marked `231` it would be `!231`