Lecture 4 (1-21-26)

Shell Scripting

Motivating Questions

- Where did I put that `.c` file?
- What programs am I **running**?
- How to **save** the results of a command?
- How do you **spam** your friend's terminal?
- how do I **terminate** a process?

how do you see the shell variables

- use the `env` command
- to access a specific variable use
  - `echo $MY_VARIABLE`

Why do you want files?

- it allows things such as variables to be **persistant**
- to see the size of a file you can use `ls -l` or `stat`
  - these return the same data in different format
- **metadata**: data about other files
  - the metadata is stored in an **inode**
    - each file has an associated **inode**
- remember file ownership is usually stored as -ownerGroupWorld
  - example: `-rw-rw-rw-`
    - the first `-` indicates the type of file
    - then each `rw-` means each group had read/write permissions
      - to convert the octal code know that:
        - `r` = 4
        - `w` =2
        - `x` = 1
      - so lets say you have. `someFile.txt`
      - to change the permissions to everyone can do everything I can do
        - `chmod 777`
        - `chmod +x` this will add the executable permission to just the **owner**

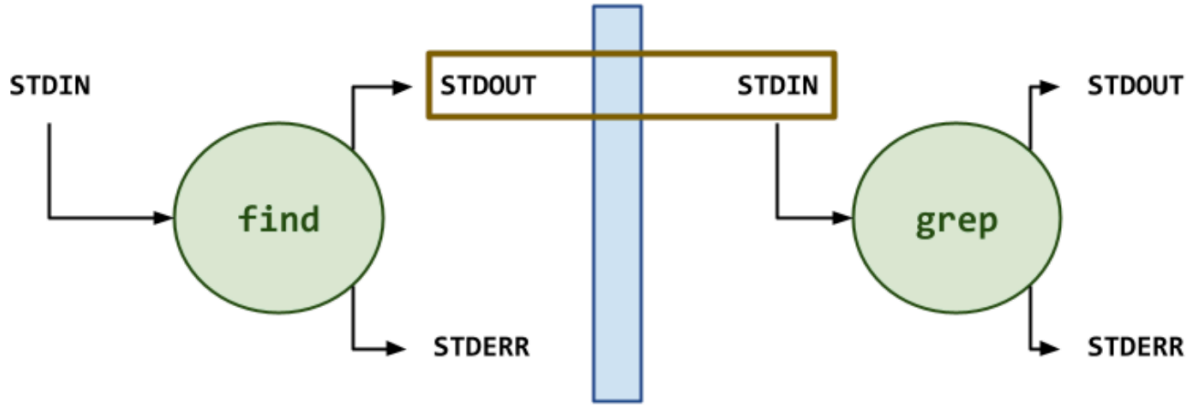
- remember if you do `which ls` to see where `ls` is stored

What makes a program a process?

- it has to be loaded from hardrive into **RAM**
- a **process** is a loaded instance of a program
- When I use control c i am sending a **signal**
- what is the program i can use to search for things in the content of files
  - `grep`
- if i use `grep '\.c$'` i will only get files that end in `.c`
  - `\.c$` is a **regular expression** (will talk more about this next week)
- Now lets see how many files end in `.c`
  - `find systems | grep '.c$' | wc - l

Process: Unix Pipeline

- Every process you make has 3 files:
  - `stdin` (0)
  - `stdout` (1)
  - `stderr` (2)
- visual of what happens when we `grep '\.c$' | wc -l`



  - the first node would represent the `find systems`
  - the blue pipe represents the `|`
  - the second node would be the `grep '\.c$'`

I might want to save the results when I run a program

- `find systems-program | tee output2`
  - what `tee` does is it outputs to the terminal **and** outputs to the file (`output2`)
- `diff output output2`

    - this command allows you to see the differences in the files
    - if it outputs **nothing** then there is nothing different
- `sha1sum output*`
    - this prints hexcodes for each file that start with output
    - if the hex codes are the same, the files are the **same content wise**

remember `/etc` stores configuration files

- find `/etc > output` throws errors into the terminal instead of `output`
    - this is because the **pipe** only redirects the **stdout** to **stdin**
        - see diagram above
    - if you do `find /etc &> combined`
        - all output will be thrown into the file `combined`
        - Now I can see if the word "denied" was in the output using:
            - `grep denied combined`
    - if you do `find /etc > combined 2> dev/null`
        - all the errors are thrown into the **black hole** (just thrown away lol)

how do i spam a friends terminal?

- back in the day pbui used to use the command `write` to hit up his friends
    - example of writing to pbui

```
write pbui
what up
lol
```

- if you use `mesg n` then nobody can send messages to you

How do i terminate a process?

- you can use control-Z to put a process to sleep
- use `ps ux` to see all processes on the system
- a process **cannot** ignore `SIGKILL` which is dash 9
    - so `kill -9 {PID}`
        - to kill the process with the given PID
            - don't use curly brace in command