Lecture 36 (4-27-26)

Exam 4 Review

- review `find` command

Written part:

- Part one: shell scripting (1.5 pts)
  - How would you use git to update some code?
  - How would you check if a file is **reachable? writeable**? **executable**?
  - How does a **shell script** process **command line arguments?**
  - Given a **pipeline**, where is **stdin, stdout, stderr, redirect** to/from
  - he will ask a short circuit question pertaining to the test command
  - no regex (i don't know if i trust this)
- Part two: **Python: Scripting** (1.5 pts)
  - How would you run a **Python** script?
  - How would you **count** the number of unique items in a file?
  - How would you use **map/filter/lambda**?
  - How do you **sort** by multiple factors?
    - probably will have to sort by counts
  - How do you write a **list comprehension**?
- Part three: C: **PAS/ Memory Allocation** (1.8 pts)
  - How would you build a **doubly linked list?** with **sentinel?**
  - How much **memory** is **allocated** on **data, heap, stack** given some **singly linked list** code?
- Part four: C MM/Linked Lists (1.2 pts)
  - How would you iterate through a **singly linked list? recursively?**
  - When should you allocate on the **stack** vs **heap**?
  - What is the difference between **unitialized memory, invalid memory access, memory leak**
- Part five: SysCalls: Files (2 pts)
  - How do you get the size of a file?
  - How do you check if a **file** is **readable, writable, executable?**
    - `access` sys call
  - How do you **list** the **components** of a **directory?**
- Part six: SysCalls: Processes (2 pts)

    - How would **execute** another program in a **child process?**
    - How would you create / fix:
        - **orphans**
        - **zombies**
        - **fork bomb**
    - How would you identify **task parallelism? data parallelism?**
- Part seven : SysCalls: Networking (2 pts)
    - What are the **components** of a **URL** such as `http://google.com`
    - What happens on both the **client** and **server** during a **HTTP** transaction?
- also remember task parallel, data parallel, embarrassingly parallel
- also concurrency and parallelism

Programming part:

1. **Using the shell**
    1. locate and copy a file (`find` and `cp`)
    2. scan remote machine for open ports (`nmap`)
    3. you will have to do something with that file
2. **Programming in Python**
    1. Sort data (maybe multi factor sorting)
    2. Use regular expressions to extract fields
        1. remember you could use `csvreader` , `.split()` if u have csv
    3. 4 components on this part:
        1. open a file
        2. extract things
        3. sort things
        4. print things
3. **Networking in C**
    1. Perform a HTTP PUT transaction
        1. uploading some file to a server
    2. Copy a file
        1. how would you copy a file:
            1. read from one, write to the other
                1. `curlit` and in lecture `copyit`

Server System Calls order

- Server Socket:
    1. `getaddrinfo`
    2. `socket`
    3. `bind`
    4. `listen`
    5. `accept`
- After each accept:
    - there is a client socket
        1. `read`
        2. `write`
        3. `close`
- the client tells the server its done with its request by a blank line