RECAP

RECAP:

- Course split into 2 parts
  - Linux
  - C

RECAP: LINUX

- Understanding directory structure
  - General layout
  - Relative vs absolute paths
  - What is `/`, `..`, `~`, etc.
- Basic commands to navigate via terminal
  - Examples: `cd`, `mkdir`, `ls`, etc.
- Basic commands to view files
  - Examples: `more`, `cat`, `uniq`, etc.
- Remote access and file transfer

RECAP: LINUX

- Compressing/Uncompressing files
- Archiving files
- Recording shell sessions (`script`)
- `history` command
- Input/output/error redirection
- Piping
- File manipulation (ex: `cut`, `tr`, etc.)
- Basic utilities (ex: `diff`, `wc`, `grep`, etc.)

RECAP: LINUX

- git: operations, general idea, local and remote, branches, etc.
- file permissions
  - what they mean
  - viewing them, changing them
- processes:
  - viewing, killing, etc.
  - running in foreground vs background
- aliases, environment variables

RECAP: LINUX

- **Be able to use the manpages**
- Bash scripting
- sed
- gawk
- you should be able to use bash scripting, sed, awk, grep (and know what should be used when)
- regular expressions (special characters, character classes)

RECAP: C

- Harder to break down
- Understand compiled vs interpreted (what compiling means, compiling options)
- preprocessor
- header files

RECAP: C

- Be able to write, compile, and run C programs
- Know datatypes, operators, bitwise operators
- Know I/0 (stdin/stdout and file)
- loops, conditionals, functions, switch
- arrays
- pass-by-reference vs pass-by-value

RECAP: C

- pointers + pointer arithmetic
- purpose of and when to use "dereference" and "value of" operators
- stack vs heap
- how and when to allocate memory dynamically
- difference between the various memory allocation functions
- `gdb` and `valgrind`

RECAP: C

- Makefiles
- enums, unions, structs
- 2D arrays and how they are handled
- string and memory functions