Software Systems

Day 7 - Tools, Streams

Agenda

- Announcements
- Warmup
- Tools and Streams in UNIX
- Files in C

Warmup

- In Pennsylvania, license plates are typically 3 letters followed by 4 numbers. Assuming this is the only plate format and any letters/numbers are possible, how many bits of information do you get from a specific license plate number?
- What is the type of foo here?
```
int *foo(char bar, int *baz)
```
- Are there any problems with this code? Why or why not?
```
char* greet(char* name) {
  char greeting[42] = "Hello, ";
  strcat(greeting, name);
  return greeting;
}
```

Today: Tools and Streams

- In the reading, you saw two important concepts:
  - Tools should generally be small and do one thing well.
  - Streams like stdin, stdout, and stderr are abstractions and can be redirected.

Everything is a File

- Why do we care so much about streams and files?
- In UNIX, there's the concept that "everything is a file".
  - By abstracting many things in a computer system as a stream of bytes (a file), we can use the same APIs to read and write a variety of streams.
  - Keyboard input is read like a file.
  - Network connections can be written to like a file.
  - Lots of system information is represented as a file.
- An equivalent concept in Python is "everything is a dictionary".

Everything is a File: Exercise

- On your systems, what files are in the /proc directory?
- Look in a numbered directory, then print the contents of the status file there with the cat command. What do you see?
  - (If you want to search for a specific process, you can use `pgrep -f -n <command name>`)
- What is in the fd directory? What do you think this is? (ls -l might help here.)
- What is in the environ file? What do you think this is?
- What is in the limits file? What do you think this is?

Data Streams: UNIX

- Many UNIX tools read from standard input by default and write to standard output by default.
  - A fair number do both. So `wc foo.txt` and `wc < foo.txt` will both work.
  - You can redirect stderr with 2: `black --diff hello.py 2> foo.txt`
  - You can even redirect stderr to stdout: `black --diff hello.py 2>&1`
- If you ever need them, there are a few special files:
  - /dev/null does nothing so you can throw away output: `ls > /dev/null`
  - /dev/zero has only zero bytes (\0).
  - /dev/random has random bytes.

Data Streams: C

- In C, you can use fprintf to write somewhere:
  - fprintf(stderr, "The answer is %d\n", 42);
- Use fopen() to open files.
  - `FILE* text = fopen("foo.txt", "w");`
- fopen returns a NULL pointer if something goes wrong, so check:
  - ```
    if (!text) {
      perror("Couldn't open the file");
      exit(1);
    }
    ```
- Clean up afterwards with fclose.
  - `fclose(text);`
    ```

Data Streams: Exercise

- Write a C program that writes "This is standard output" to stdout, "This is standard error" to stderr, and "This is a file" to a filename of your choice.
  - Add error checking for opening the file.

Tools: Useful Tools

- A few UNIX tools are absolutely worth knowing.
- We'll cover a few today:
  - cut
  - tr
  - sort
  - uniq
  - head
  - tail

Tools: Useful Tools

- cut allows you to get specific columns out of tabular data.
- tr allows you to translate or "squeeze" characters.
- sort allows you to sort tabular data.
- uniq "squeezes" consecutive lines, or counts them.
- head and tail get the first/last N lines of a file.

Tools: Pipes

- We can use the pipe (`|`) to redirect the output of one command to the input of another.
  - `ls` displays the files in the current directory.
  - `wc` counts lines, words, and characters in input text.
  - `wc -l` only counts the lines in its input text.
  - So `ls | wc -l` tells you how many files are in the current directory.
- You can pipe multiple commands into each other, too.

Tools: Exercise

- Write a command to print the 10 largest header files (ending with .h) in your /usr/include (or equivalent) directory, along with their sizes.

Data Streams: Exercise

- Write a C program that writes "This is standard output" to stdout, "This is standard error" to stderr, and "This is a file" to a filename of your choice.
  - Add error checking for opening the file.
- Try to run your earlier C program in a way that sends just the stderr output to wc.

Open Time

- If you want to use this time to find a project partner or discuss project ideas, feel free.
- Or get started on Assignment 2 and ask us for help.