COSC94: Accelerated Introduction to Systems Programming

Grading

This course's grade is determined entirely by the student's performance on weekly programming assignments and the final project.

Each programming assignment is assessed for correctness using an autograder, and for understanding through an interview.

Week 1: Intro to C and Unix

Material Covered

- Some indispensable Unix commands: Further commands are covered throughout the course, but not assessed directly.
  - `ssh`
  - `scp`
  - `cd`
  - `ls`
  - `mv`
  - `cp`
  - `vi`
  - `man`
- The integer types
- Structs
- Header files
- Invoking the compiler driver
  - And what the compiler driver actually does (`cpp` -> `cc1` -> `as` -> `ld`)
- Invoking `clang-format`
- Exit statuses

Assignment

- Make a simple vec3 library. Using any external functions is forbidden.
  - Addition
  - Scalar multiplication
  - Dot product
  - Cross product

Week 2: Pointers, System Calls

Material Covered

- ASCII
- Null-terminated strings
- `argv`
- Pointer types
- The `&` and `*` operators
- `.` and `->`
- System calls
  - read, write, open, close, and exit (not the libc wrappers! Only with libc's `syscall`)
- UBSan and ASan

Assignment

- Implement a bunch of libc functions. Using any external functions is forbidden, except `syscall(2)`.
    - `close`
    - `getchar`
    - `puts`
    - `strcat`
    - `strcpy`
    - `strncpy`
    - `exit`
    - `open`
    - `read`
    - `strcmp`
    - `strlen`
    - `strtol` (This one is hard!)
    - `write`

Week 3: Data Representation and Debugging

Material Covered

- From this point in the class, students are allowed to use libc.
- Binary, octal, and hexadecimal
- Integer representation
    - Endianness
    - 2's complement
- Bitwise operations
- `gdb`

Assignment

- Write an IP packet parser.

Week 4: Assembly

Material Covered

- What a CPU does
    - Fetch, decode, execute
- What an instruction is
- What memory is
- What a register is
- Handout with a bunch of instructions
- Making system calls in assembly

Assignment

- Implement `hexdump` in assembly.

Week 5: More Assembly

Material Covered

- The ABI

- The stack

Assignment

- Write a simple stack unwinder in assembly that is callable from C

Week 6: Memory Model, Dynamic Allocation, Heaps

Material Covered

- Virtual memory
- The address space layout (high-level)
- `malloc`, `calloc`, and `realloc`
- `free`
- When to use dynamic memory, and when not to
- `mmap`

Assignment

- Implement a memory allocator
  - One free list, singly-linked

Week 7: Threads and Processes

Material Covered

- Threads
- fork
- exec

Assignment

- Implement a mini shell.

Week 8: IPC

- Signals
- Sockets

Final Project: Game Boy Emulator (Goes to the end of the term)

- We give them a wrapper around SDL2 that provides a “plot pixel” interface.
- They are responsible for implementing the Game Boy’s little variable-width instruction set (subset of Z80)
- Link cable over a socket.