OS 2/18

Reagan Shirk

February 18, 2020

ASCII

- ASCII is a mapping between bytes, characters, glyphs
- Unicode helps us because... something about having too many things for ASCII?
- Unicode = [bytes/codepoints] → glyphs
- We're using UTF-8 in this class, it transforms codepoints to bytes
  - We're paying attention to the first bit in an 8-bit string
    - If the first bit is 0, the whole thing is an ASCII character
    - If the first bit is 1, we need to look at a few other things
      - If the first bit is 1 and the next is 0, something about not needing any more space...?
      - If the first bit is 1 and the next is 1, you have a two byte character where the first bit of the second byte is also 1
      - The process continues for needing more bytes, i.e. if you need 3 bytes your first 3 bits of the first byte will be 1, then you'll have two more bytes where the first bit is 1

C Stuff

- `syscall` is a wrapper
- `syscalls` is a list of different system calls, I think
- `size <filename>` returns the text, data, bss, dec, hex sizes for the file
- `strace <filename>` let's you see "what's going on under the hood" according to Grant
- `exit` is a wrapper that does some cleanup before calling `_exit`
- `_exit` is a syscall

Forks

- You have a zombie process and an orphan process
- An orphan process happens when the parent dies
  - The orphan is adopted by `pid = 1` init/systemd