

OS 3/3

Reagan Shirk

March 3, 2020

Midterm

- March 10th, a week from today
- Study the questions we wrote
 - Focus on **concepts**, don't just memorize questions and answers
- Purely multiple choice
- Up to chapter 4, we can ignore stuff that is specific to Mac/Windows OS
 - Stay with Linux/Unix stuff
 - Know the pros/cons of these architectures, we need to understand why the people who built these OS's built them the way they did
- Next project will be assigned the day of the midterm, will be due 2-3 weeks after it is assigned

Clone and Pthreads

- `vfork()` has no duplication of virtual memory (which is **unlike** `fork()`)
- Parents memory is shared until `exec` or `exit` is called
 - `exec` == starting a new program
- Any changes to the stack or heap of the parent are seen in the parent
- The child of `vfork()` is **guaranteed** to be called
- `vfork()` = virtual fork
- A file descriptor is just an integer that stores how to access the file
- Different processes can point to the same file
 - Because they're different processes, they can write to the same file at the same time
 - You can tell which process is doing what by looking at the bytes...?
- `clone()` is faster than `fork()` and `vfork()`, you can think of `clone()` as a thread

Thread Stack

- Stack and heaps have shared things in threads (libraries, memories, etc)
 - Each thread has its own execution stack, each thread is stored in an overall stack for the main thread
 - They all share program code
 - Threads **can not** see each others' stacks but they can see what is in the shared stack (the main thread stack)

Pthread

- API standard for how you should use threads
- `pthread()` = POSIX threads