Lecture 7, Jan 23, 2024

Context Switching

- Older OSes are *uniprogramming* operating systems, which can only run one process at a time
- All modern operating systems are *multiprogramming* operating systems, which allow processes to run in parallel or concurrently
- The scheduler decides when to switch processes, which involves a process of:
  1. Pausing the current process
  2. Saving the current process state
  3. Getting the next process to run from the scheduler
  4. Load the next process' state and let it run
- We can have processes pause themselves voluntarily via a syscall (*cooperative multitasking*), or the OS will take control and choose when to pause processes (*true multitasking*)
  - Generally all general-purpose operating systems use true multitasking
  - For true multitasking the OS can give processes time slices and wakes up periodically using interrupts to do scheduling
- Swapping between processes is called *context switching*, which is overhead – this is wasted time that we want to minimize

Pipes

- `int pipe(int pipefd[2]);` forms a one-way communication channel using two file descriptors
  - The first file descriptor is the read end of the pipe; the second is the write end of the pipe
    - Note the `pipefd` is an output parameter
  - Any data written to `pipefd[1]` can be read from `pipefd[0]`
  - The pipe is managed by the kernel using an internal buffer
  - Returns 0 on success, `-1` and sets `errno` on failure
- One usage of this is to create a pipe and then fork, which will also give the child access to the pipe, allowing IPC between the child and parent
- The pipe closes if no process has the read or write end of the pipe open
  - If both ends of the pipe are open by at least one process, the pipe won't close, even if both ends are opened by the same process
  - Best practice is to close the file descriptors when we no longer need them, otherwise the pipe won't close
  - e.g. if the parent needs to send data to the child, the parent should close the read end of the pipe and the child close the write end of the pipe