Multiprocessing Wrap

- **Today's Agenda**
  - I still need to work through the synchronization examples I didn't get to on Friday.
    - Buggy program is here.
    - Fixed program, which relies on signals masks, is right here.
  - I want to introduce the **`kill`** system call, which allows one process to send an arbitrary signal to another.
    - The function is pretty easy to understand, but I want to work through a short programming puzzle just to make sure.
  - I want to give you some high-level sense as to how the OS supports multiprocessing so that each process can operate as if it owns all of its 4GB (or 256TB, in a 64-bit world) virtual address space. This part is all concept and whiteboard illustrations, but there's no code.
  - I also want you to understand how the scheduler can round-robin through all of the active processes so that each of them gets enough processor time to make progress. Again, all concept and whiteboard drawings, but no code.