Intro to Processes

Process Definitions

- **Definitions/terms**
  - **Process** - a program "in execution"; The smallest unit of work the OS explicitly keeps track of.
  - **Job** - the work submitted by the user to the OS. It can be broken down into one or more tasks which become processes when they become "in execution" state.
  - **Program** - code that a process executes.
    - Program is passive. Process is active.
    - Several processes can use the same program.

Process States

Unix Process States

Process Context

Process Data Structures

- **Process Control Block** - one per process
- **Process Table** - one for the whole OS, contains all the PCBs
- **PROCESS CONTROL BLOCK**
  - PCB = OS data structure to keep track of a process
Process Control Block

- PCB Components (partial list)
  - Process state (ready, waiting, running)
  - Program counter (PC points to the next instruction to execute)
  - CPU registers
  - CPU scheduling information
  - Memory management information
  - Open files
  - I/O status information
  - Accounting information

Process Code Example

Example:
- User-submitted job: `cc myprog.c`
- Resulting tasks: `cc myprog.c myprog.a`  
  `as myprog.a myprog.o`  
  `ld -lc myprog.o myprog`
- Resulting processes:
  - process 1 (PCB1 + code for C compiler + files)
  - process 2 (PCB2 + code for assembler + files)
  - process 3 (PCB3 + code for loader + files)
  - process 4 (PCB4 + code for profiler + files)