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

- New
- Ready
- Running
- Exit
- Blocked

Event Occurs
Event Wait
Timeout
Dispatch
Release
Admit

Figure 3.5 Five-State Process Model

Process Context

CPU
- PC
- TR
- CC

REGS

MEMORY
- PGM CODE

DISK
- PGM CODE FILE
- DATA FILE
- DATA FILE

PC = program counter; TR = instruction register; CC = status flags
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)

Unix Process States

![UNIX Process State Transition Diagram](image)

Figure 3.15 UNIX Process State Transition Diagram