CS 3305A

Process

Lecture 2

Sept 13th 2023

Topics

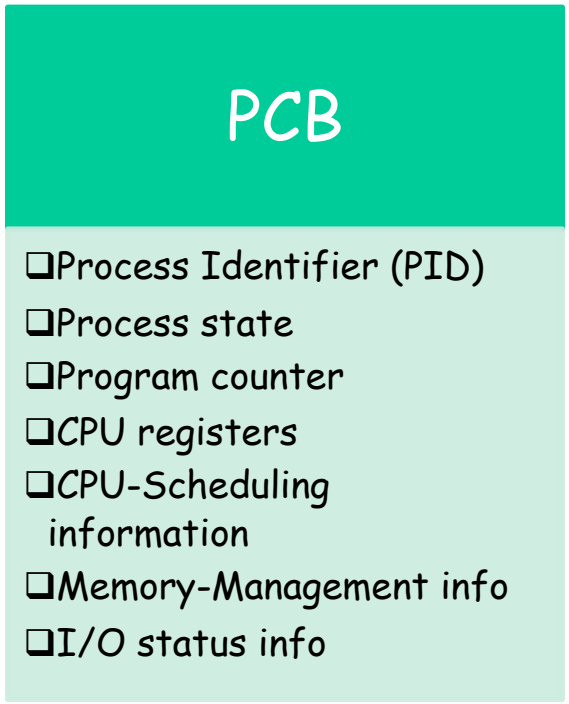
- Process Basics
- How to create a process using C
  - fork() system call
- Programming Demo: Parent & Child process
  - fork1.c
  - fork2.c

Process

- A process is an instance of an application running
- As a process executes it changes state.
- Possible states:
  1. New: The process is created
  2. Running: Instructions are being executed
  3. Waiting: The process is waiting for some event to occur e.g., I/O completion
  4. Terminated: Process has finished execution

Process Control Block

- Each process is represented in the operating system by a process control block (PCB)
  - PCB includes information such as:
    - Process Identifier (PID)
    - Process state
    - Program counter
    - CPU registers
    - CPU-Scheduling information
    - Memory-Management information
    - I/O status information



The Concept of Fork

- The Unix system call for process creation is fork()
- The fork system call creates a child process that is a duplicate of the parent.
  - Child inherits state from parent process
    - Same program instructions, variables have the same values, same position in the code.
  - Parent and child have separate copies of that state
    - They are stored in separate locations in memory
    - Important: updating the value of a variable inside the child will NOT update that variable in the parent, and vice-versa.

Fork System Call

- If fork () succeeds it returns integer value > 0 (which is the ID of the child) to the parent and returns 0 to the child
- If fork() fails, it returns -1 to the parent (no child is created)
- pid_t data type represents process identifiers
- Other calls:
  - pid_t getpid() – returns the ID of calling process. Call is always successful.
  - pid_t getppid() – returns the ID of parent process. Call is always successful.

fork() example