CS 3305A
Process
Lecture 2
Sept 11 2019
Process

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

- Each process is represented in the operating system by a **process control block (PCB)**
  - **Context information including:**
    - 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 called `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
Fork System Call

- If `fork()` succeeds it returns the child PID 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 PID of calling process.
  - `pid_t getppid()` - returns the PID of parent process.