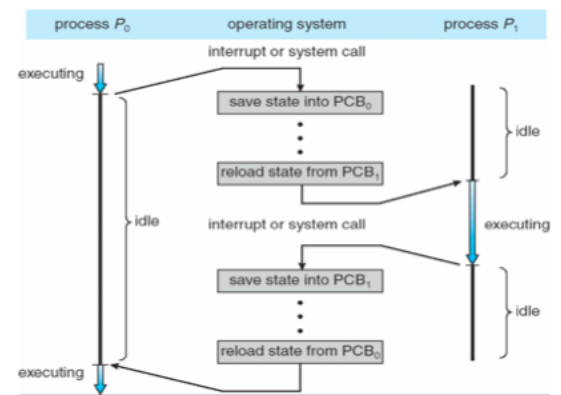
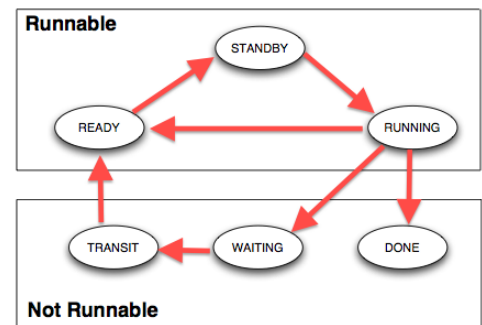


## Ch.3 – Processes

- Process contains a program counter, stack, and data section.
  - Text section: program code itself
  - Stack: temporary data (function parameters, return addresses, local variables)
  - Data section: global variables
  - Heap: contains memory dynamically allocated during run-time
- Process Control Block (PCB): contains information associated with each process: process state, PC, CPU registers, scheduling information, accounting information, I/O status information
- Types of processes:
  - I/O Bound: spends more time doing I/O than computations, many short CPU bursts
  - CPU Bound: spends more time doing computations, few very long CPU bursts
- When CPU switches to another process, the system must save the state of the old process (to PCB) and load the saved state (from PCB) for the new process via a context switch
  - Time of a context switch is dependent on hardware
- Parent processes create children processes (form a tree)
  - PID allows for process management
  - Parents and children can share all/some/none resources
  - Parents can execute concurrently with children or wait until children terminate
  - fork() system call creates new process
    - exec() system call used after a fork to replace the processes' memory space with a new program
- Cooperating processes need interprocess communication (IPC): shared memory or message passing
- Message passing may be blocking or non-blocking
  - Blocking is considered synchronous
    - Blocking send has the sender block until the message is received
    - Blocking receive has the receiver block until a message is available
  - Non-blocking is considered asynchronous
    - Non-blocking send has the sender send the message and continue
    - Non-blocking receive has the receiver receive a valid message or null



## Ch.4 – Threads

- Threads are fundamental unit of CPU utilization that forms the basis of multi-threaded computer systems
- Process creation is heavy-weight while thread creation is light-weight
  - Can simplify code and increase efficiency