The picture can't be displayed.

Chapter 4: Processes

Chapter 4: Processes

- Process Concept
- Process Scheduling
- Cooperating Processes
- Interprocess Communication

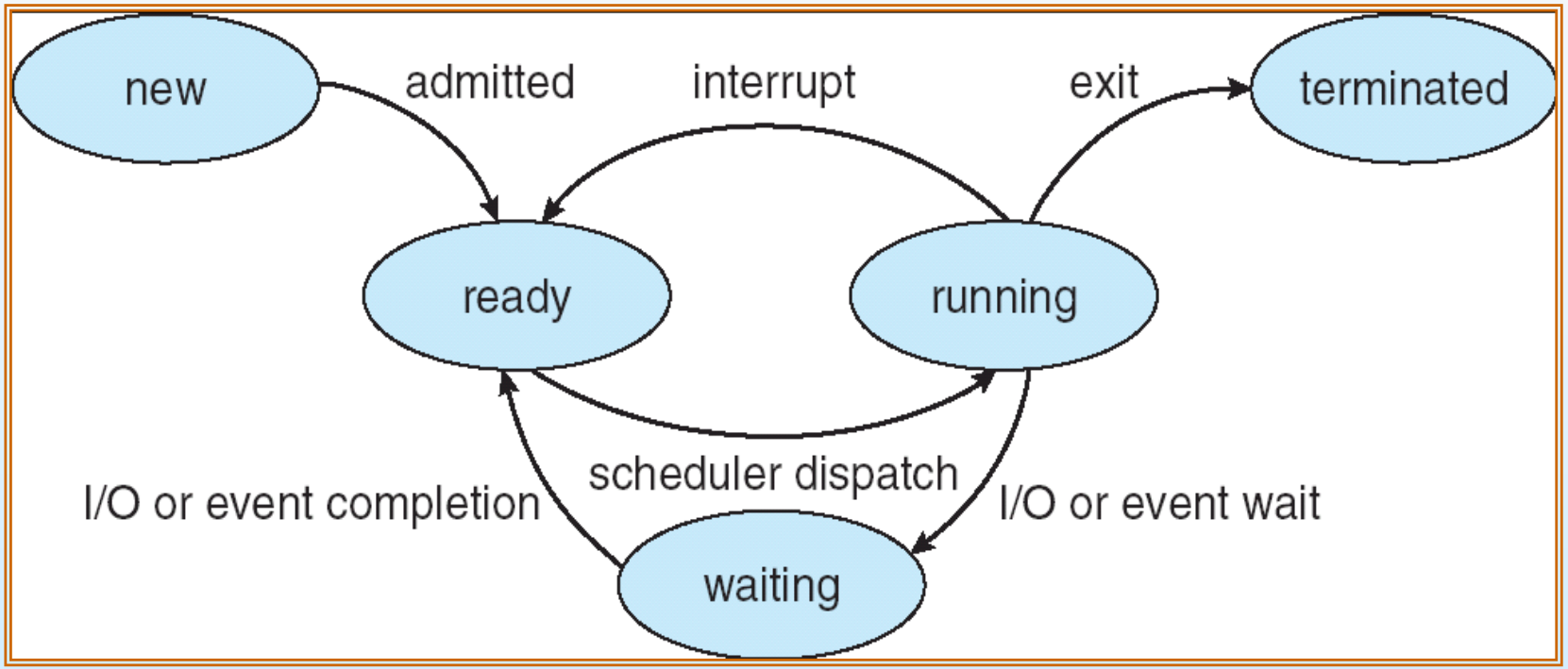
Process Concept

- An operating system executes a variety of programs:
  - Batch system – jobs
  - Time-shared systems – user programs or tasks
- Textbook uses the terms *job* and *process* almost interchangeably
- Process – **a program in execution**; process execution must progress in sequential fashion
- A process includes:
  - program counter
  - Stack (temporary data, parameters, return address, local variable)
  - data section (global variable)
  - heap (dynamically allocated variable)

Process State

- As a process executes, it changes *state*
  - **new**: The process is being created
  - **running**: Instructions are being executed
  - **waiting**: The process is waiting for some event to occur
  - **ready**: The process is waiting to be assigned to a processor
  - **terminated**: The process has finished execution

Diagram of Process State

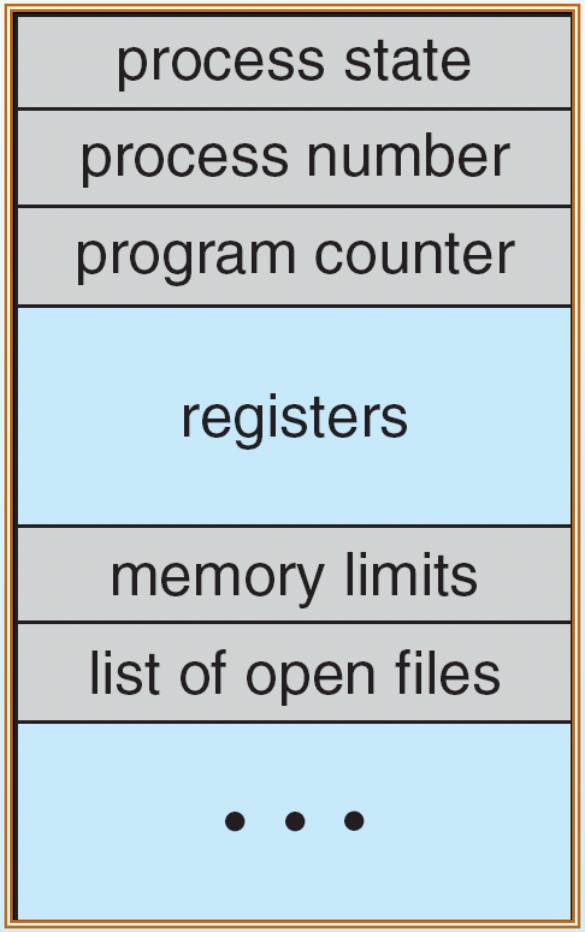


Process Control Block (PCB)

Information associated with each process

- Process state
- Program counter
- CPU registers
- CPU scheduling information
- Memory-management information
- Accounting information
- I/O status information

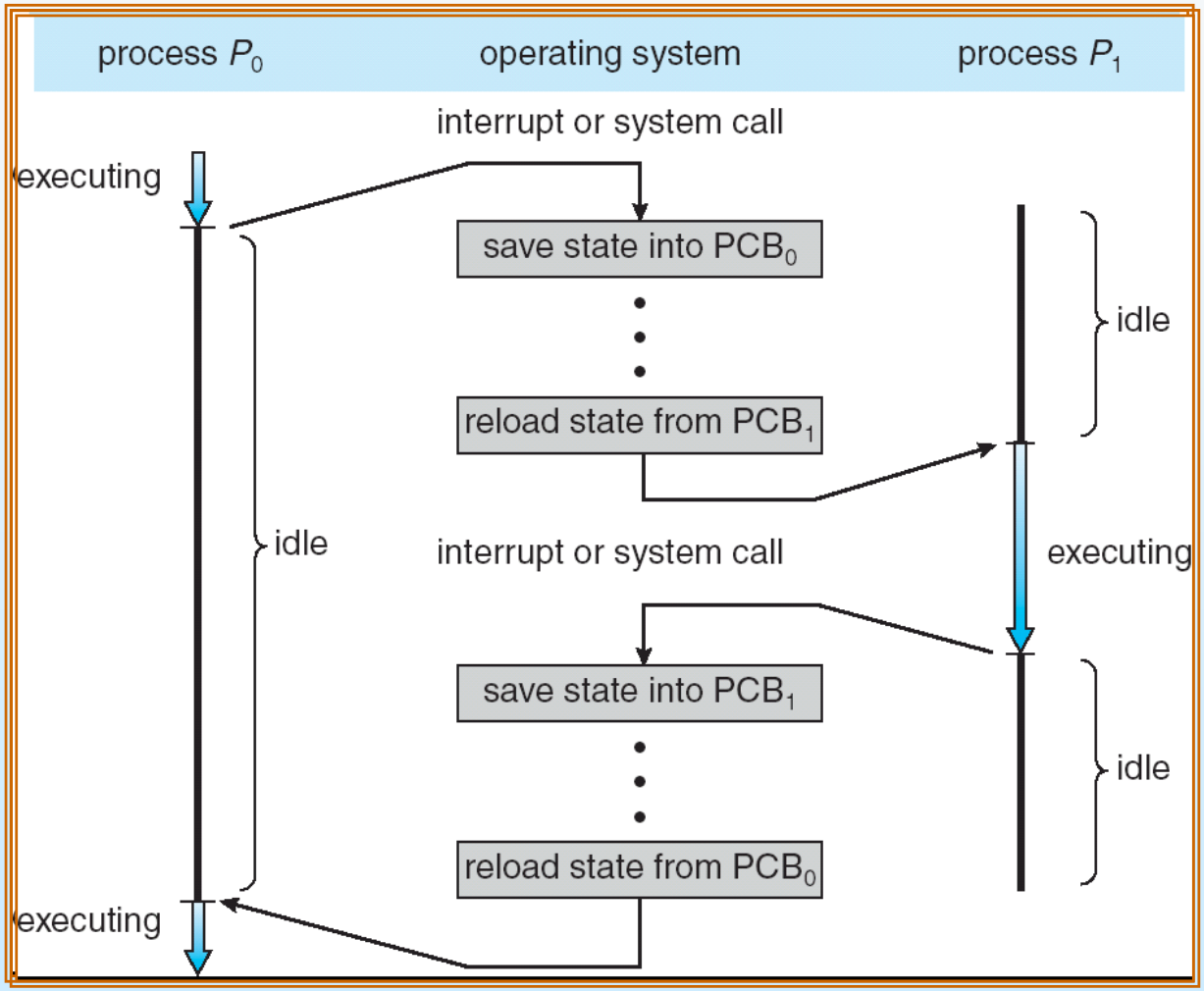
Process Control Block (PCB)



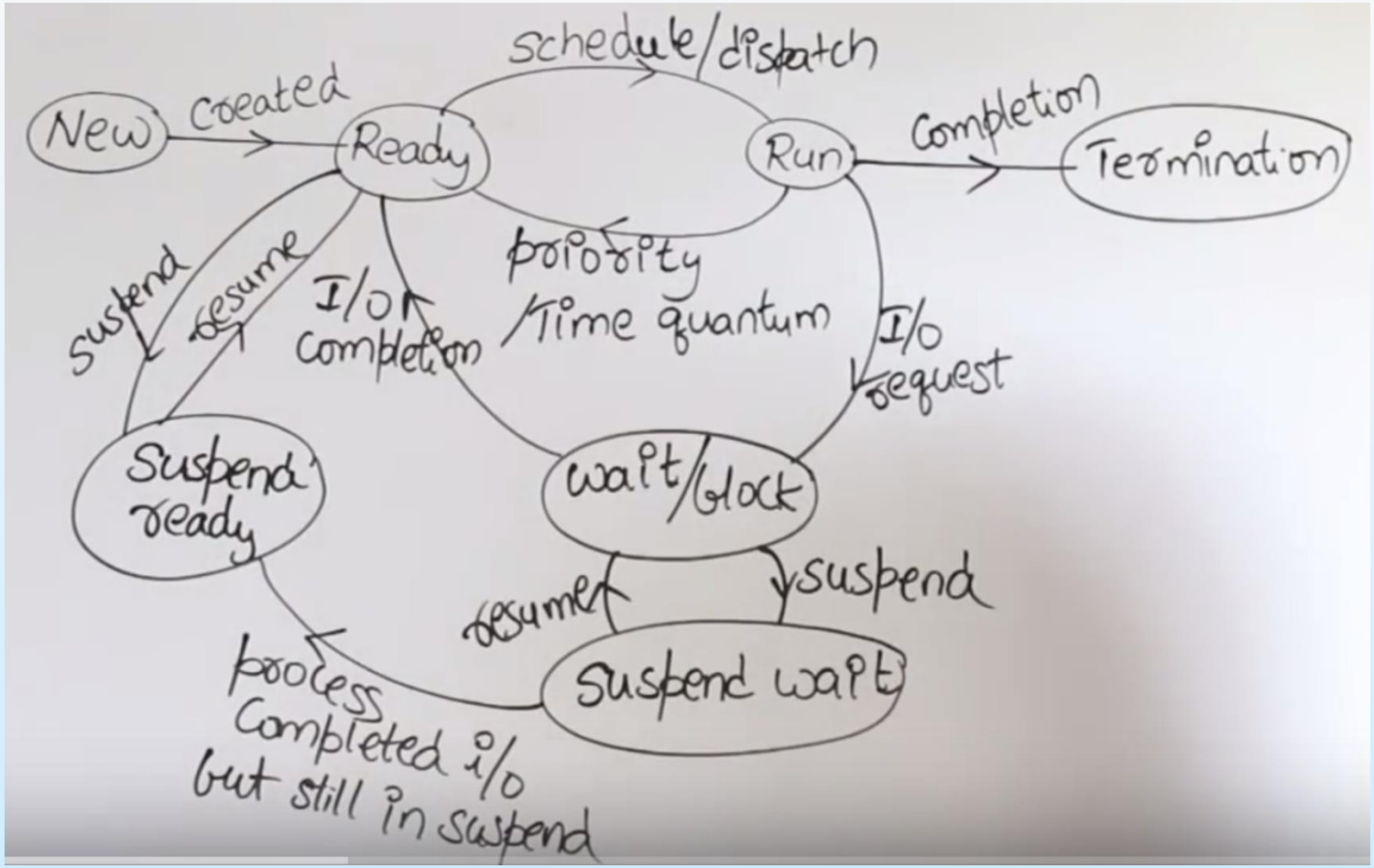
Context Switch

- When CPU switches to another process, the system must save the state of the old process and load the saved state for the new process
- Context-switch time is **overhead**; the system does no useful work while switching
- Time dependent on hardware support

CPU Switch From Process to Process



Process Scheduling



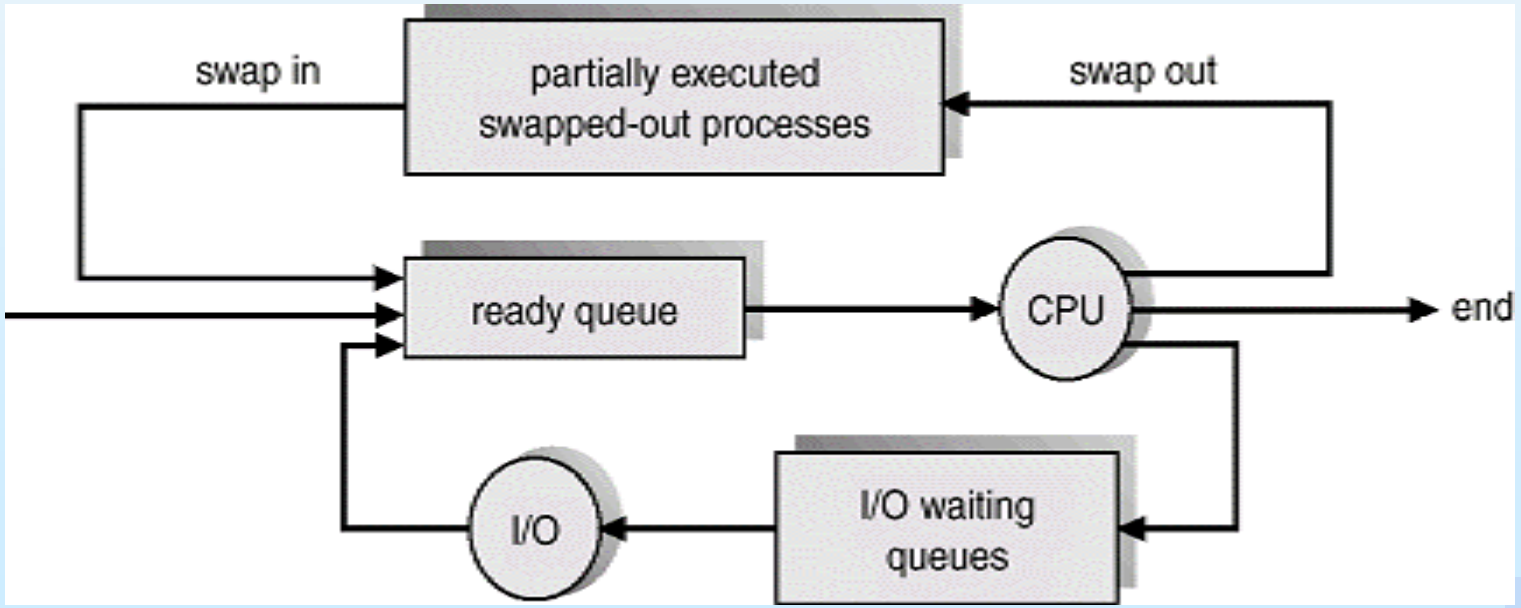
Schedulers

- **Long-term scheduler** (or job scheduler) – selects which processes should be brought into the ready queue

- **Short-term scheduler** (or CPU scheduler) – selects which process should be executed next and allocates CPU

Mid-term scheduler

- sometime it can be advantageous to remove process from the memory and thus to reduce the degree of multi-programming.



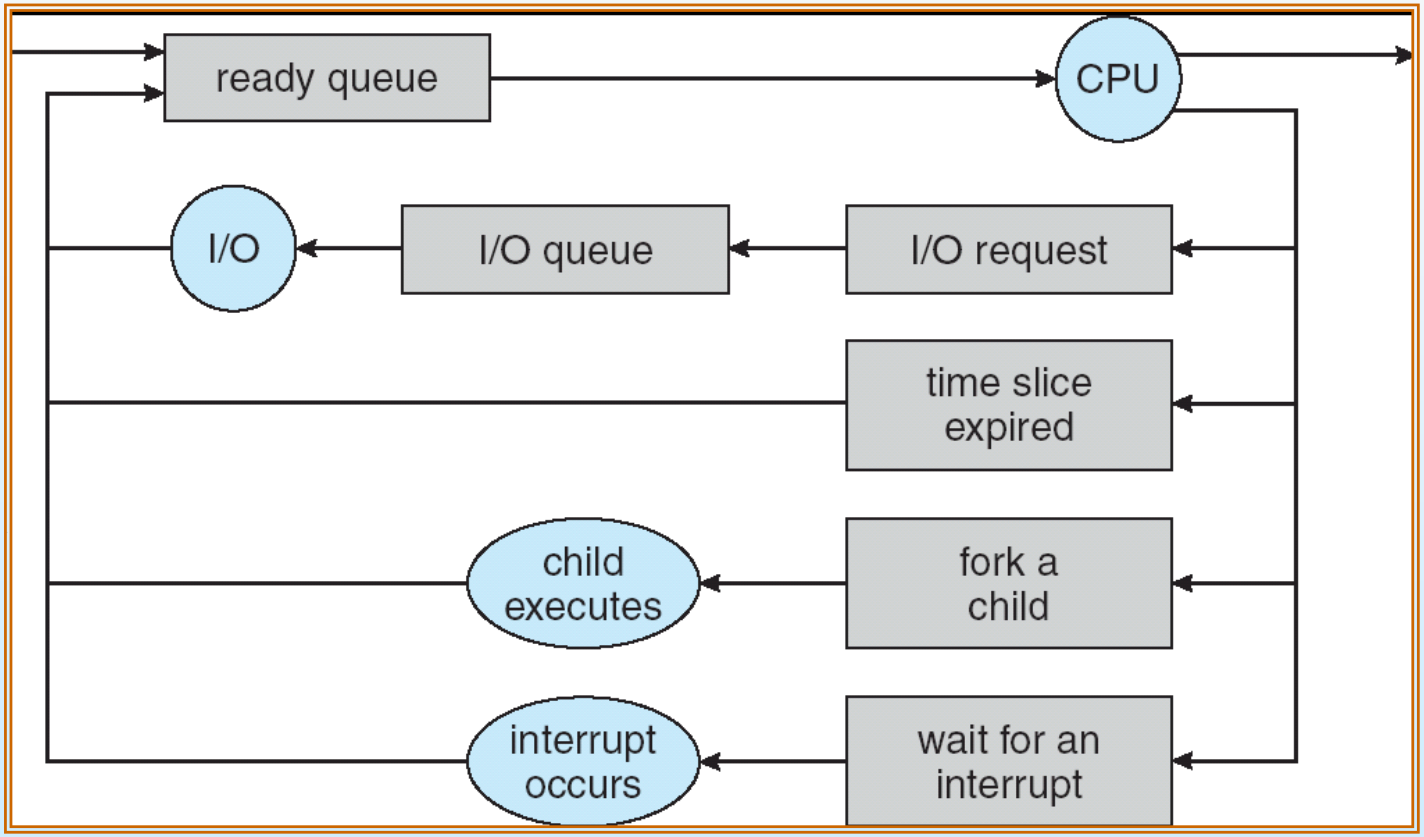
Schedulers (Cont.)

- Short-term scheduler is invoked very frequently (milliseconds) ⇒ (must be fast)
- Long-term scheduler is invoked very infrequently (seconds, minutes) ⇒ (may be slow)
- The long-term scheduler controls the *degree of multiprogramming*
- Processes can be described as either:
  - **I/O-bound process** – spends more time doing I/O than computations.
  - **CPU-bound process** – spends more time doing computations.

Process Scheduling Queues

- Maximize CPU use, quickly switch processes onto CPU for time sharing.
- Process scheduler selects among available processes for next execution CPU.
- Maintains scheduling queues of process.
- **Job queue** – set of all processes in the system
- **Ready queue** – set of all processes residing in main memory, ready and waiting to execute
- **Device queues** – set of processes waiting for an I/O device
- Processes migrate among the various queues.

(Queueing-Diagram) Representation of Process Scheduling



Process Creation

- Parent process creates children processes, which, in turn create other processes, forming a tree of processes
- **Resource sharing options:**
  - Parent and children share all resources
  - Children share subset of parent's resources
  - Parent and child share no resources
- **Execution options:**
  - Parent and children execute concurrently
  - Parent waits until children terminate

Cooperating Processes

- **Independent** process cannot affect or be affected by the execution of another process
- **Cooperating** process can affect or be affected by the execution of another process
- **Advantages of process cooperation**
  - Information sharing
  - Computation speed-up
  - Modmodularity
  - Convenience

Interprocess Communication

- Processes within a system may be **independent** or **cooperating**
- Cooperating process can affect or be affected by other processes, including sharing data, Information sharing.
- Cooperating processes need **interprocess communication** (**IPC**)
  - IPC provides a mechanism to allow processes to communicate and to synchronize their actions.
- Two models of IPC
  - **Shared memory**
  - **Message passing**

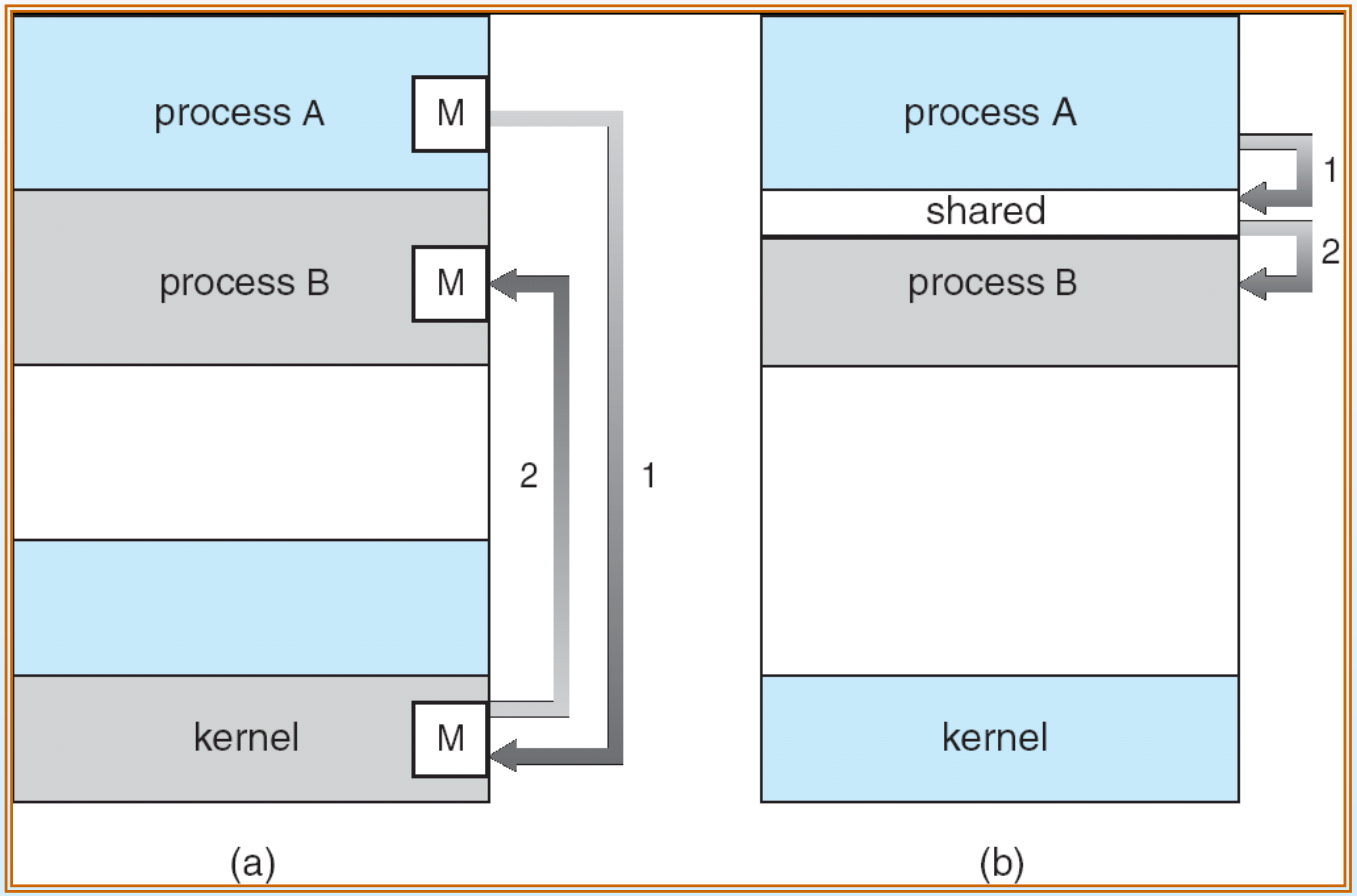
Shared-Memory

- Establish a region of shared memory
- Read and write data in the shared area
- Processes responsible for synchronization
  - Must ensure that same memory location is not being modified by multiple processes at the same time

Message-Passing systems

- Mechanism for processes to communicate and to synchronize their actions
- Message system – processes communicate with each other **without** resorting to shared variables
- IPC facility provides two operations:
  - **send**(*message*) – message size fixed or variable
  - **receive**(*message*)
- If *P* and *Q* wish to communicate, they need to:
  - establish a *communication link* between them
  - exchange messages via send/receive

Interprocess Communication (IPC) Models



a) Message Passing

b) Shared Memory

The picture can't be displayed.

End of Chapter 4