# Operating Systems

## Processes

# Chapter 3: Processes

- Process Concept
- Process Scheduling
- Operations on Processes
- Interprocess Communication

# Process Concept (1/3)

## Program vs. Process

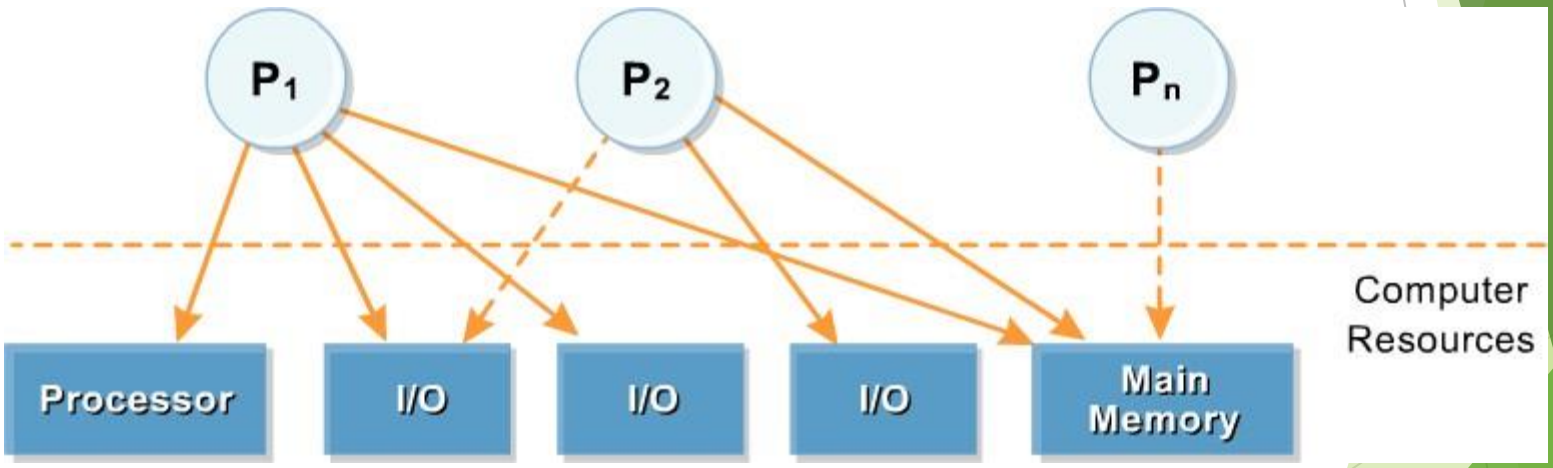
- **A program** is a **passive** entity such as the file that contains the list of instructions stored on a disk always referred to as an **executable file**.
- A program becomes a **process** when an executable file is loaded into the memory and then becomes an **active** entity.

# Process Concept (2/3)

- The fundamental task of any operating system is the **process management**.
- Processes include not only a text but also include a set of resources such as open files and pending signals. Processes also contain internal kernel data, processor state, an address space, and a data section.

# Process Concept (3/3)

- OS must **allocate resources to processes**, enable sharing of information, protect resources, and enable the synchronization among processes.



# Process Elements (1/2)

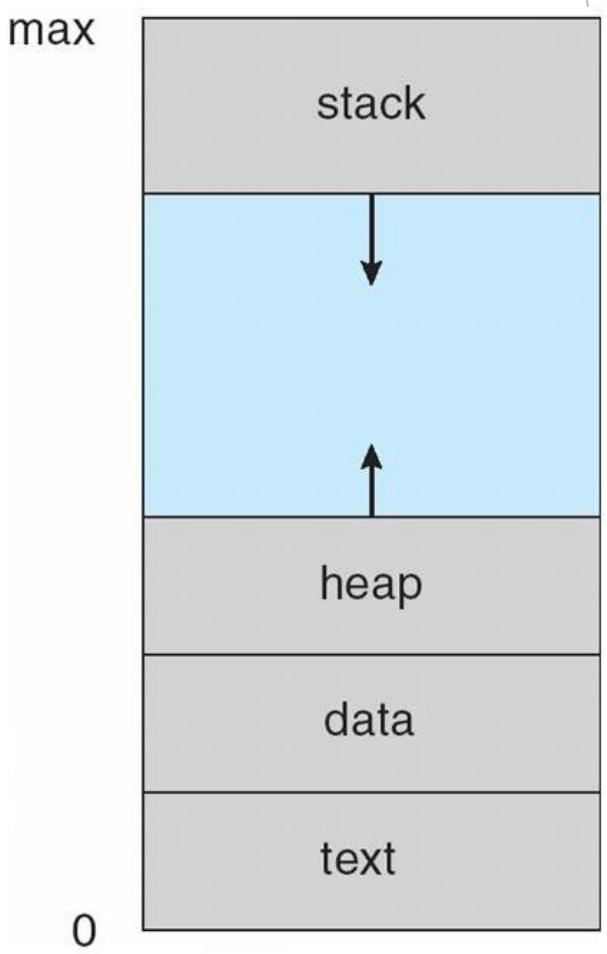
- Segments of a process represents the following

components:

- Text Section: the program code. This is typically read-only, and might be shared by a number of processes.
- Data Section: containing global variables.
- Heap: containing memory dynamically allocated during run time.
- Stack: containing temporary data.
  - Function parameters, return addresses, local variables.

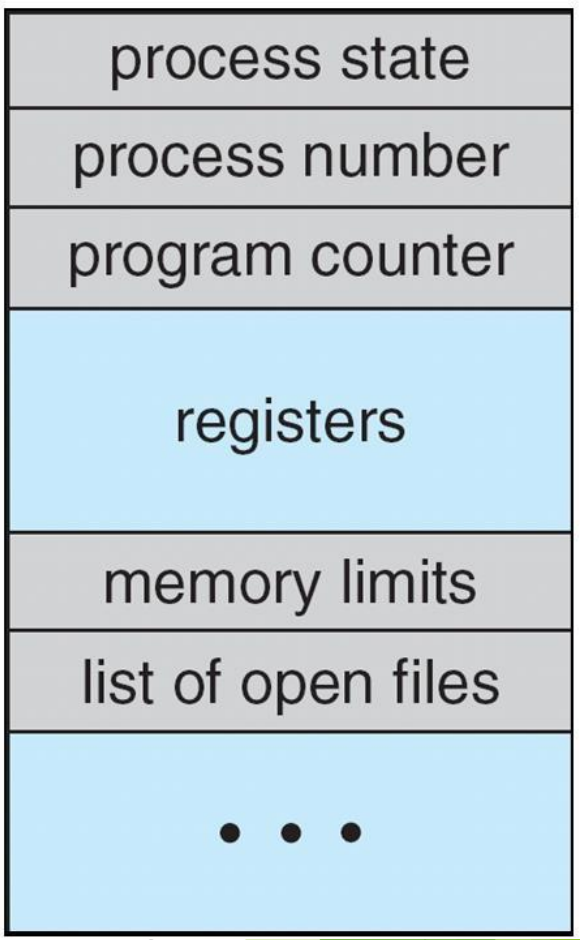
# Process Elements (2/2)

- Process in Memory



## Process Control Block (PCB)

- For better control of processes, operating systems need to consider their dynamic behaviors.
- Each process is represented in the OS by a Process Control Block (PCB).



# Process Control Block (PCB) (2/2)

- Process Control Block (PCB) (1/3)
  - **Process identification information** process number
    - Process identifier: numeric identifiers represent the unique process identifier
    - User identifier: the user who is responsible for the job).
    - Identifier of the parent process that created this process.

# Process Control Block (PCB) (2/2)

- Process Control Block (PCB) (2/3)
  - **Processor state Information**
    - Process state – running, waiting, etc
  - **Program counter**
    - location of instruction to next execute
  - **CPU registers**
    - contents of all process-centric registers

# Process Control Block (PCB) (2/2)

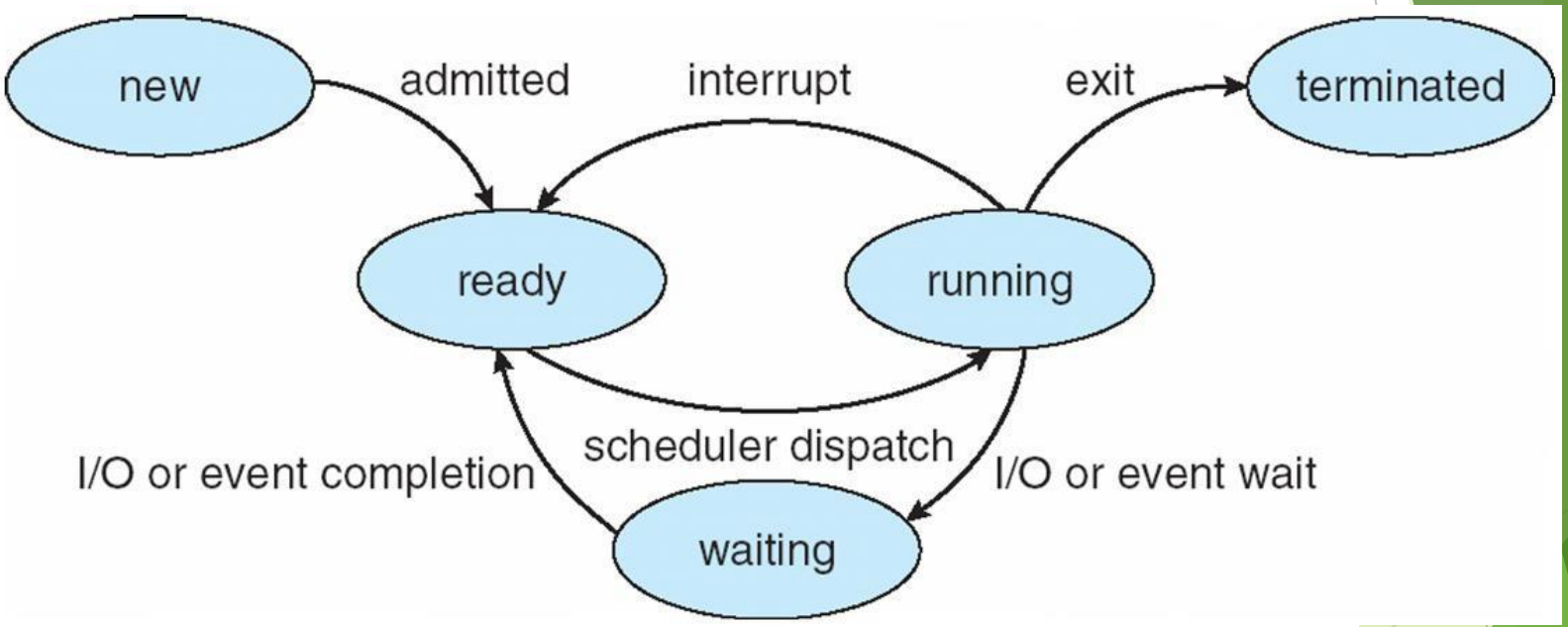
- Process Control Block (PCB) (3/3)
  - **CPU scheduling information**
    - priorities, scheduling queue pointers
  - **Memory-management information**
    - memory allocated to the process
  - **Accounting information**
    - CPU used, clock time elapsed since start, time limits
  - **I/O status information**
    - I/O devices allocated to process, list of open files

# Process State (1/3)

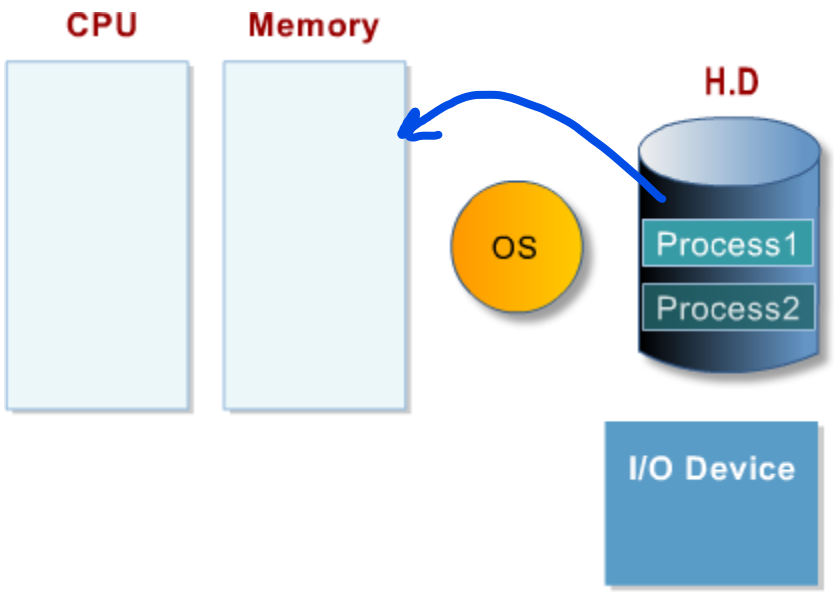
- As a process executes, it changes **state**
  - **new**: The process is being created
  - **running**: Instructions are being executed
  - **waiting**: The process is waiting for some event to occur
  - **ready**: The process is waiting to be assigned to a processor
  - **terminated**: The process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

# Process State (2/3)

- Diagram of Process State

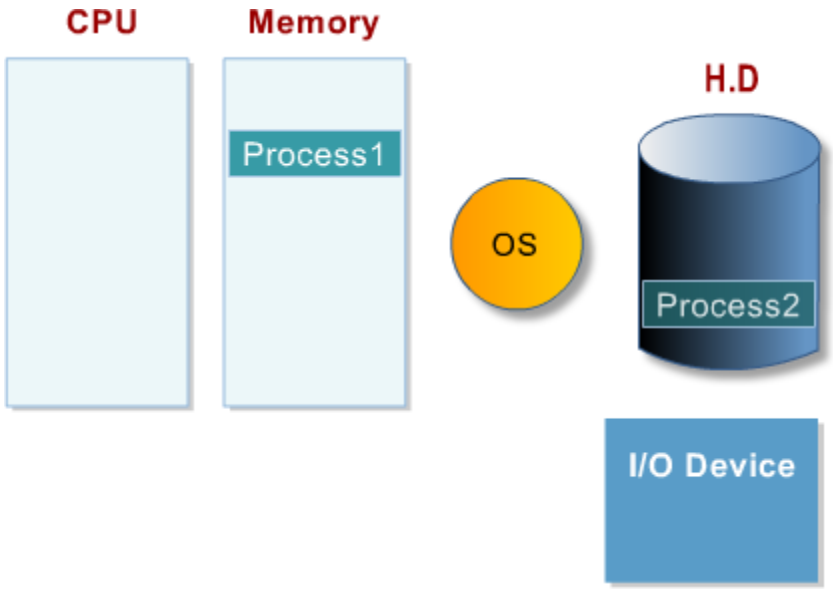


# Process State (3/3)



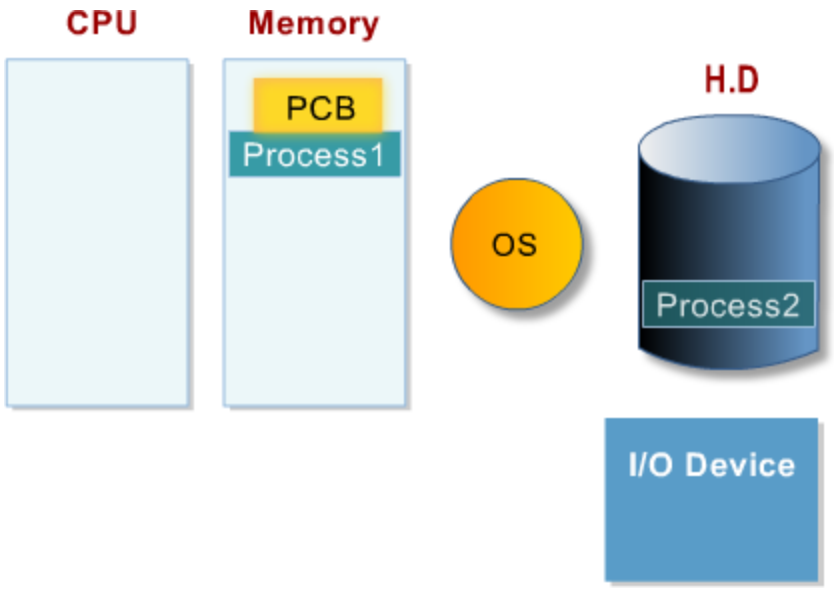
# Process State (3/3)

OS assigns the process to memory:
take the process1 from disk to memory



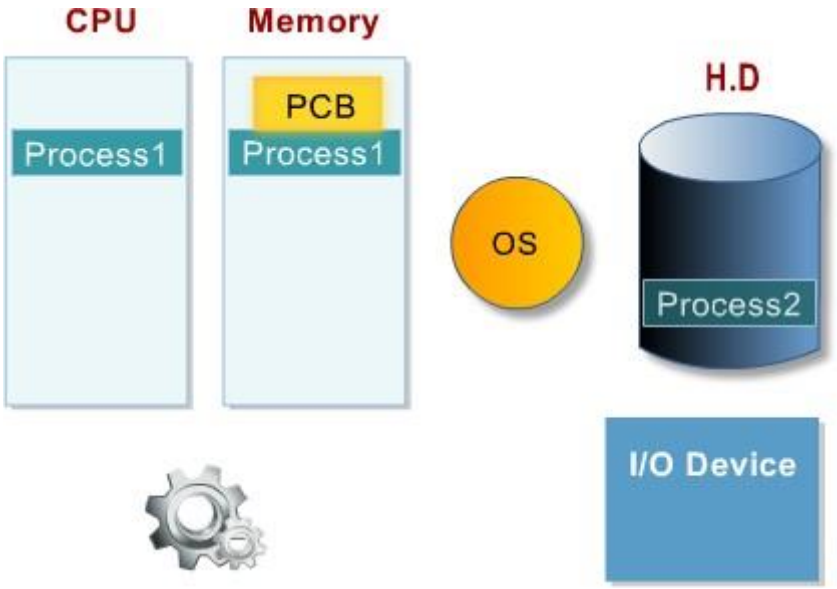
# Process State (3/3)

create PCB to process



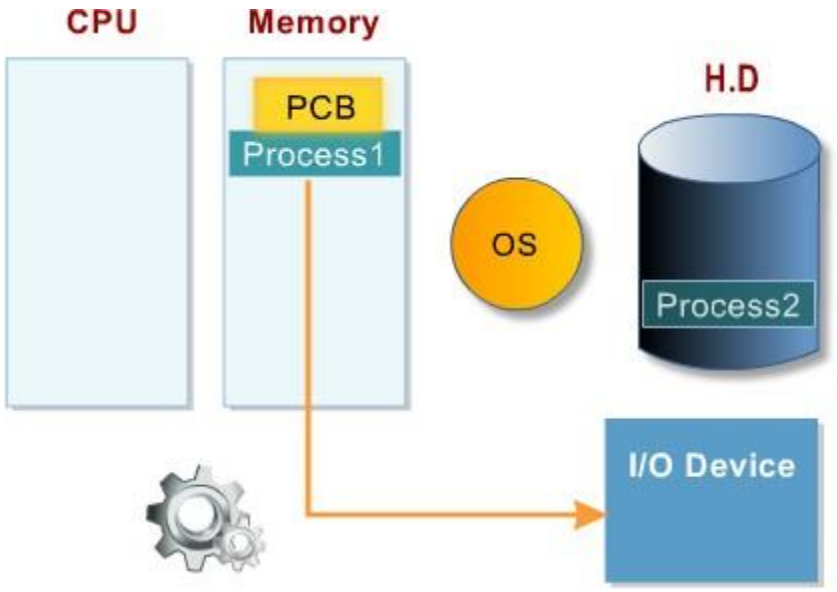
# Process State (3/3)

execute process1:
Assign process1 to the process



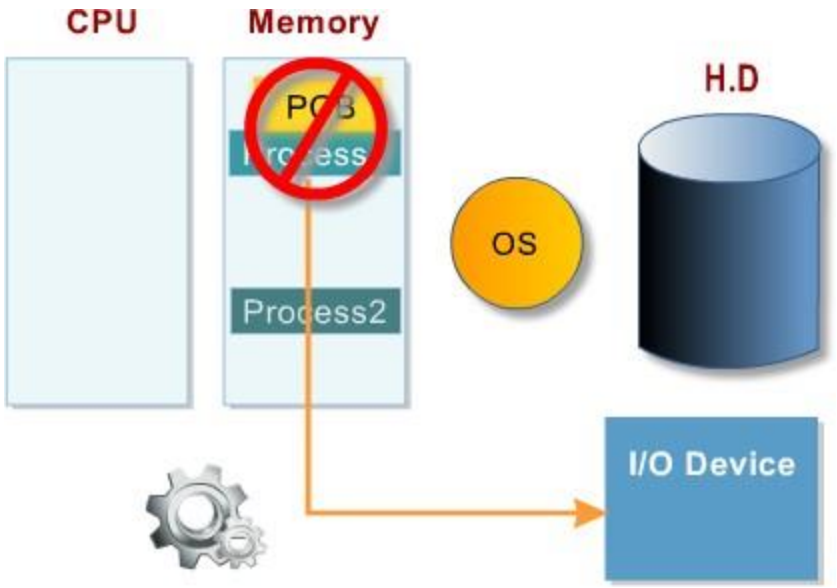
# Process State (3/3)

become waiting:
wait for i/o

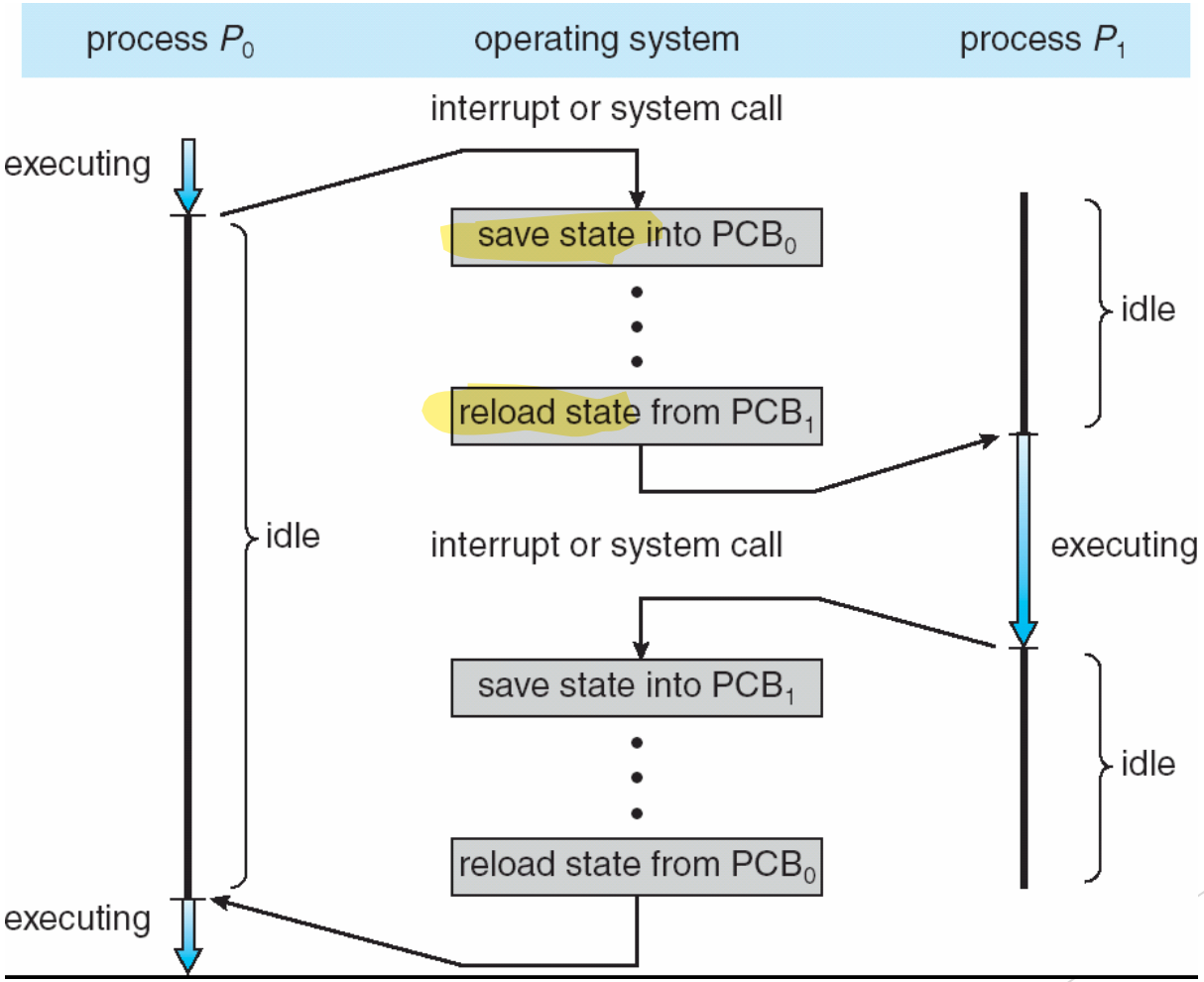


# Process State (3/3)

finish: terminate

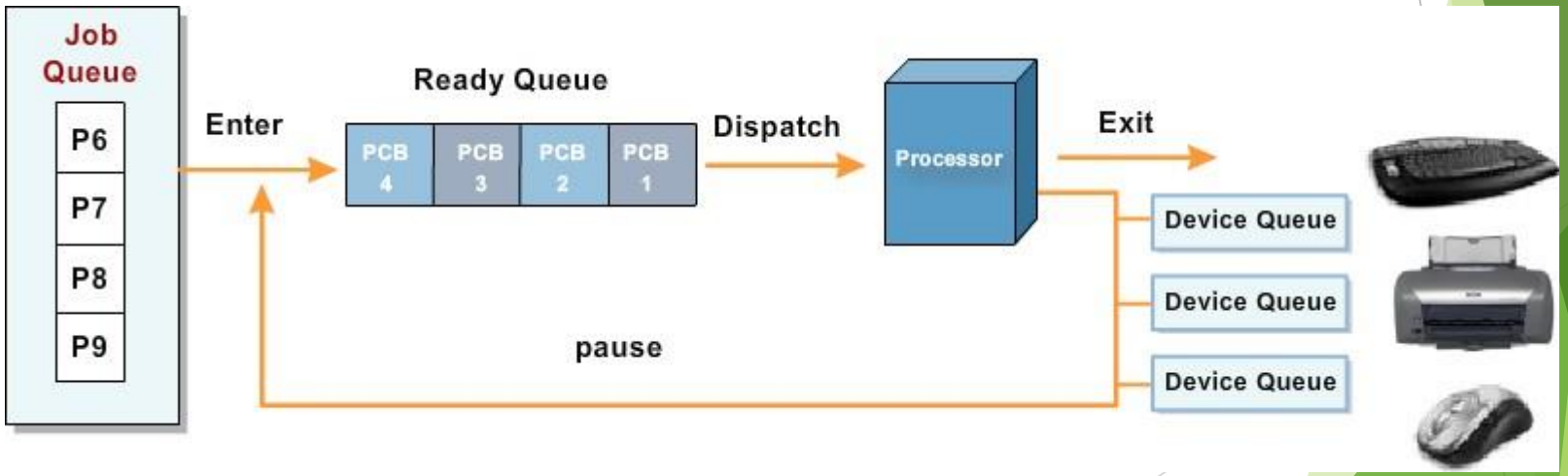


# CPU Switch From Process to Process

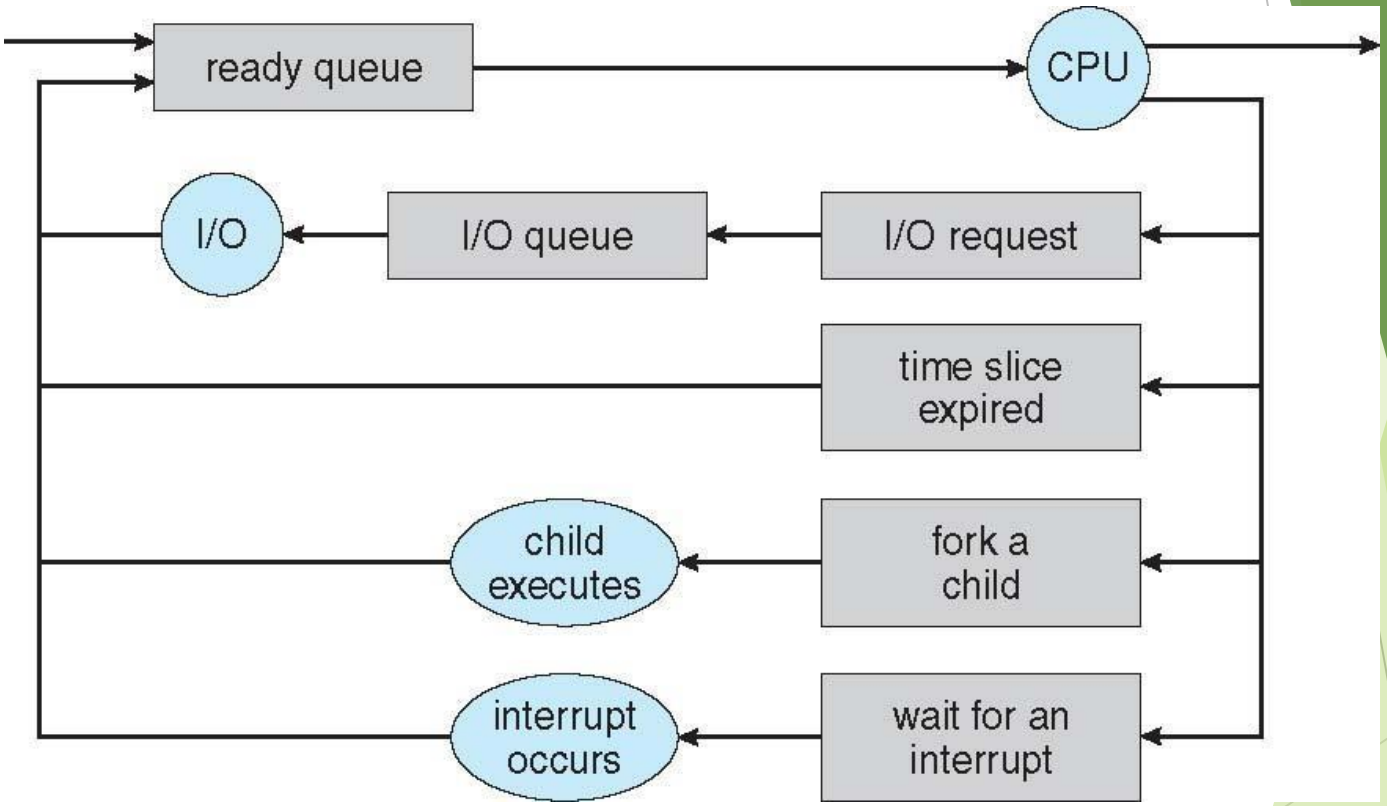


# Process Scheduling (1/2)

- **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



# Process Scheduling (2/2)



# Schedulers (1/2)

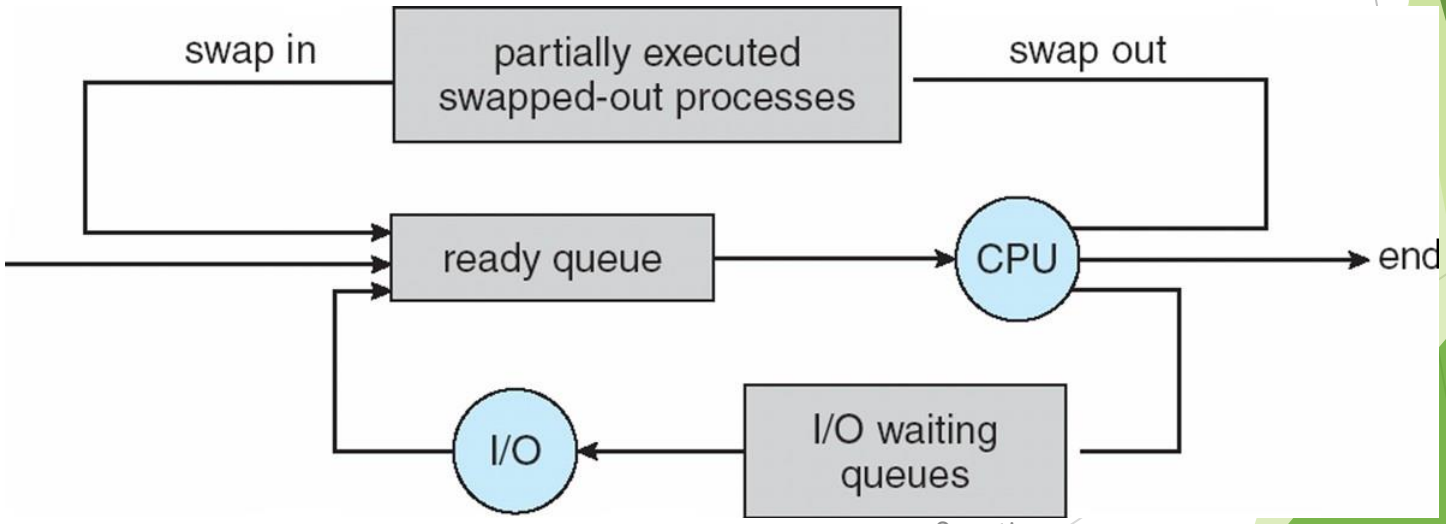
- **Short-term scheduler (or CPU scheduler)**
  - Selects which process should be executed next and allocates CPU.
  - Invoked frequently (milliseconds) → (must be fast).
- **Long-term scheduler (or job scheduler)**
  - Selects which processes should be brought into the ready queue.
  - Invoked infrequently (seconds, minutes) → (may be slow).
  - Controls the degree of multiprogramming.

move from disk to memory

switch between processes

# Schedulers (2/2)

- **Medium-term scheduler**
  - Can be added if degree of multiple programming needs to decrease
  - Remove process from memory, store on disk, bring back in from disk to continue execution: **swapping**



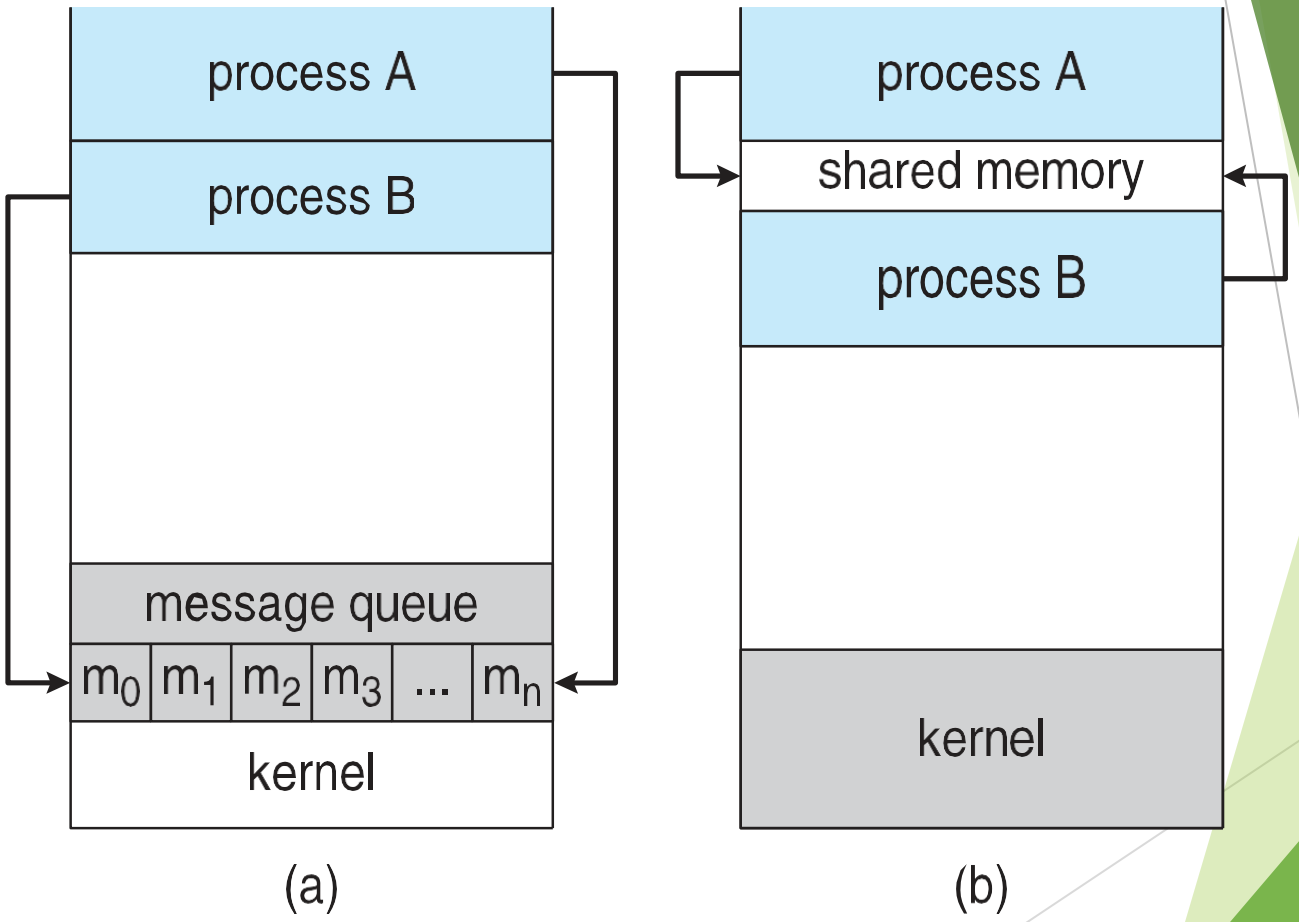
# Interprocess Communication (1/2)

- Processes within a system may be **independent** or cooperating.
- Cooperating process can affect or be affected by other processes, including sharing data.
- Cooperating process need interprocess communication (IPC).

- Two models of IPC:
  - **Shared memory**
  - **Message passing**

# Interprocess Communication (2/2)

(a) Message passing. (b) shared memory.



# Thank You