Chapter 2: Processes Management

Objectives

► To introduce the notion of a process -- a program in execution, which forms the basis of all computation

► To describe the various features of processes, including scheduling, creation and termination, and communication.

► OS must provide various provisions for synchronization, communication, and deadlock handling.

Process Concept

- An operating system executes a variety of programs:
  - Batch system – **jobs**
  - Time-shared systems – **user programs** or **tasks**

- **Process** – a program in execution; process execution must progress in a sequential fashion.

- Multiple parts of a process itself:
  - *The program code*, also called the **text section**
  - *Current activity represented as a* **program counter**
  - Contents of the processor's registers.
  - **Stack** containing temporary data
    - Function parameters, return addresses, local variables
  - **Data section** containing global variables
  - **Heap** containing memory dynamically allocated during run time

A

Process Concept (Cont.)

- ► Program is a *passive* entity stored on disk (**executable file**), the process is *active*
  - ► Program becomes process when executable file loaded into memory

- ► Execution of program started via GUI mouse clicks, command line entry of its name, etc

- ► One program can be several processes
  - ► Consider multiple users executing the same program

Process in Memory
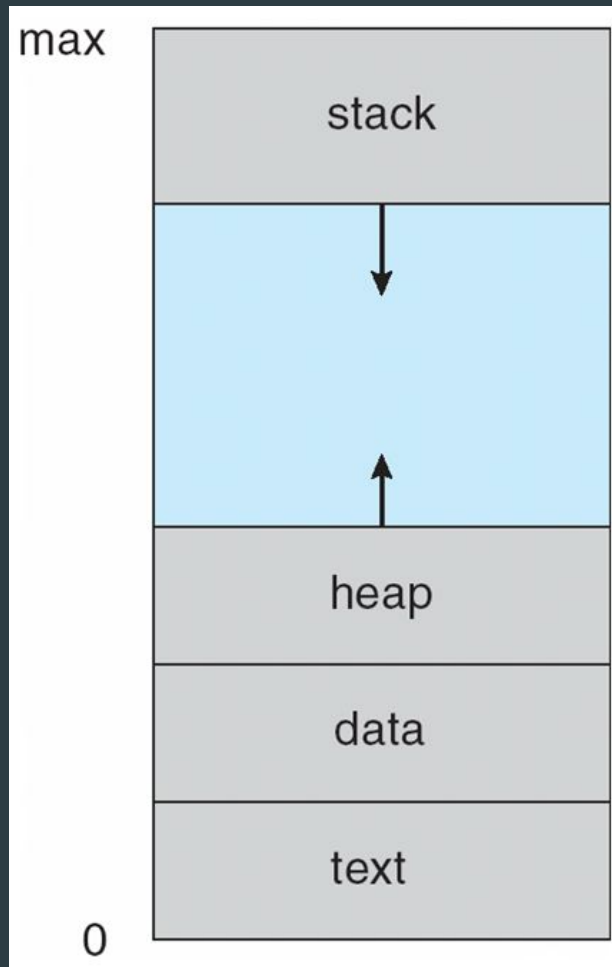
Stack: contains temporary data(function parameters, return addresses, local variables)

Heap: memory allocated dynamically.

Data: global variables.

Text: Pgm instructions

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
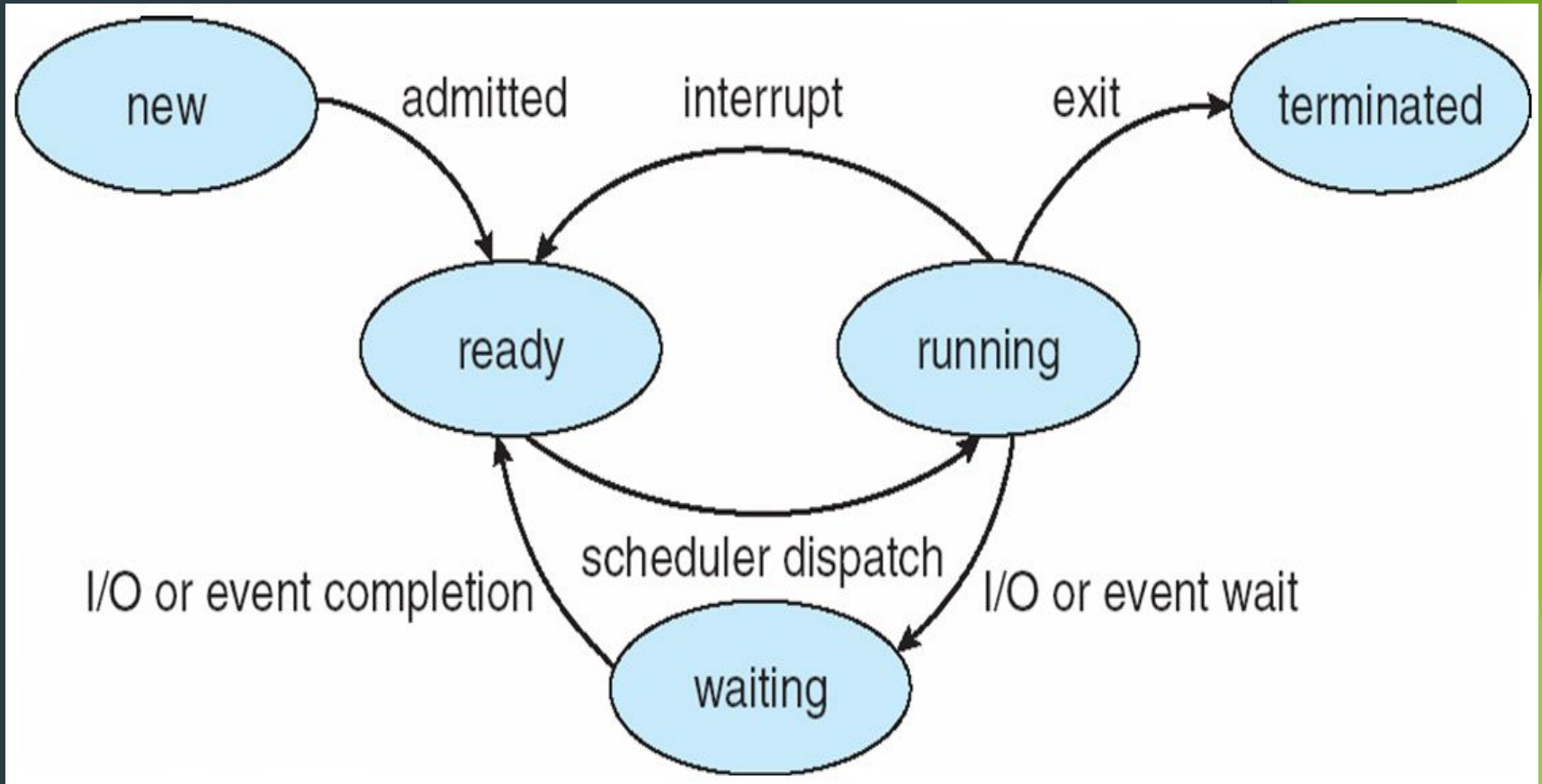
Process Control Block (PCB)

Information associated with each process

(also called **task control block**)

- ► <u>Process state</u> – running, waiting, etc

- ► <u>Program counter</u> –address/ location of instruction to execute next

- ► <u>CPU registers</u> – contents of all process-centric registers, contain addresses to help the process start where it had left.

- ► <u>CPU scheduling information-</u> priorities, scheduling queue pointers

- ► <u>Memory-management information</u> – memory allocated to the process

- ► <u>Accounting information</u> – CPU used, clock time elapsed since start, time limits,no. of processes

- ► <u>I/O status information</u> – I/O devices allocated to process, list of open files

process state
process number
program counter
registers
memory limits
list of open files
• • •

CPU Switch From Process to Process
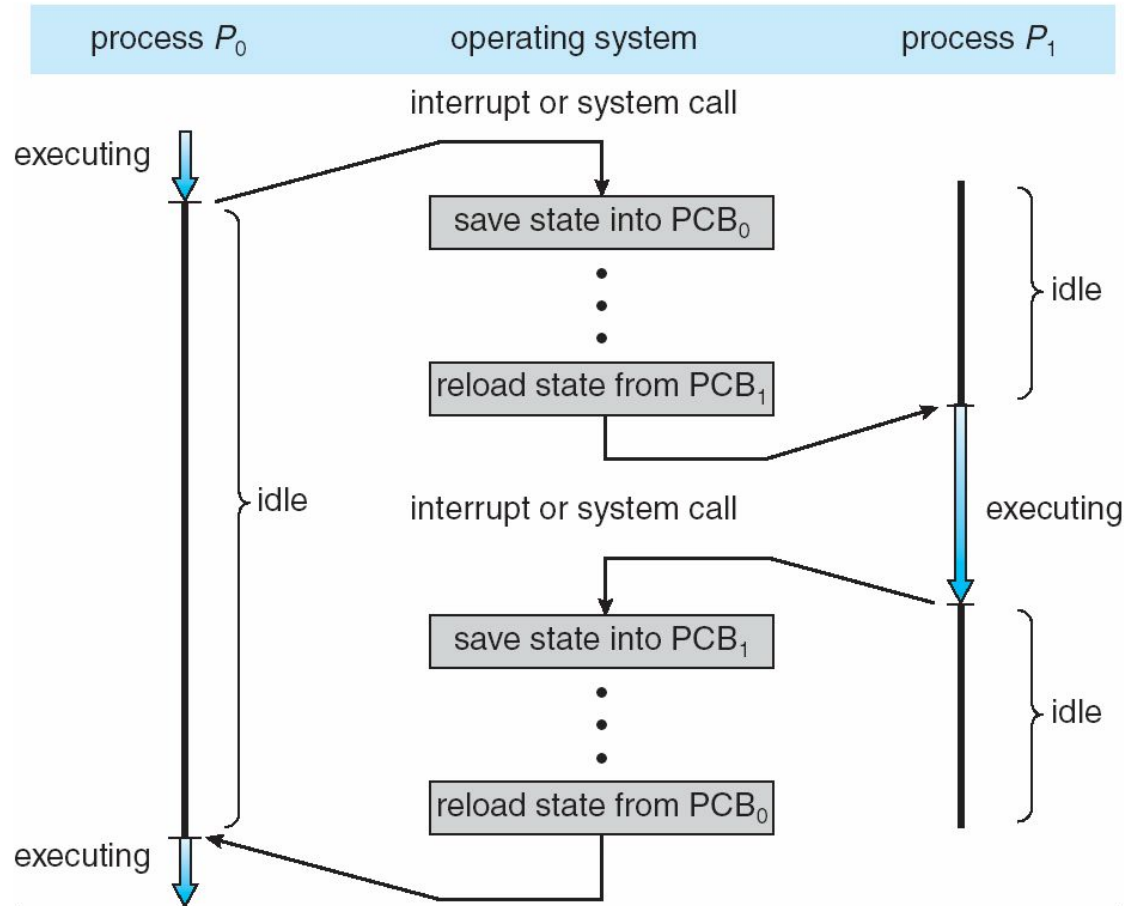
Process Scheduling

- Maximize CPU use, quickly switch processes onto CPU for time sharing

- **Process scheduler** selects among available processes for the next execution on CPU

- Maintains **scheduling queues** of processes
  - **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** – a set of processes waiting for an I/O device
  - Processes migrate among the various queue time-sharing

Ready Queue And Various I/O Device Queues
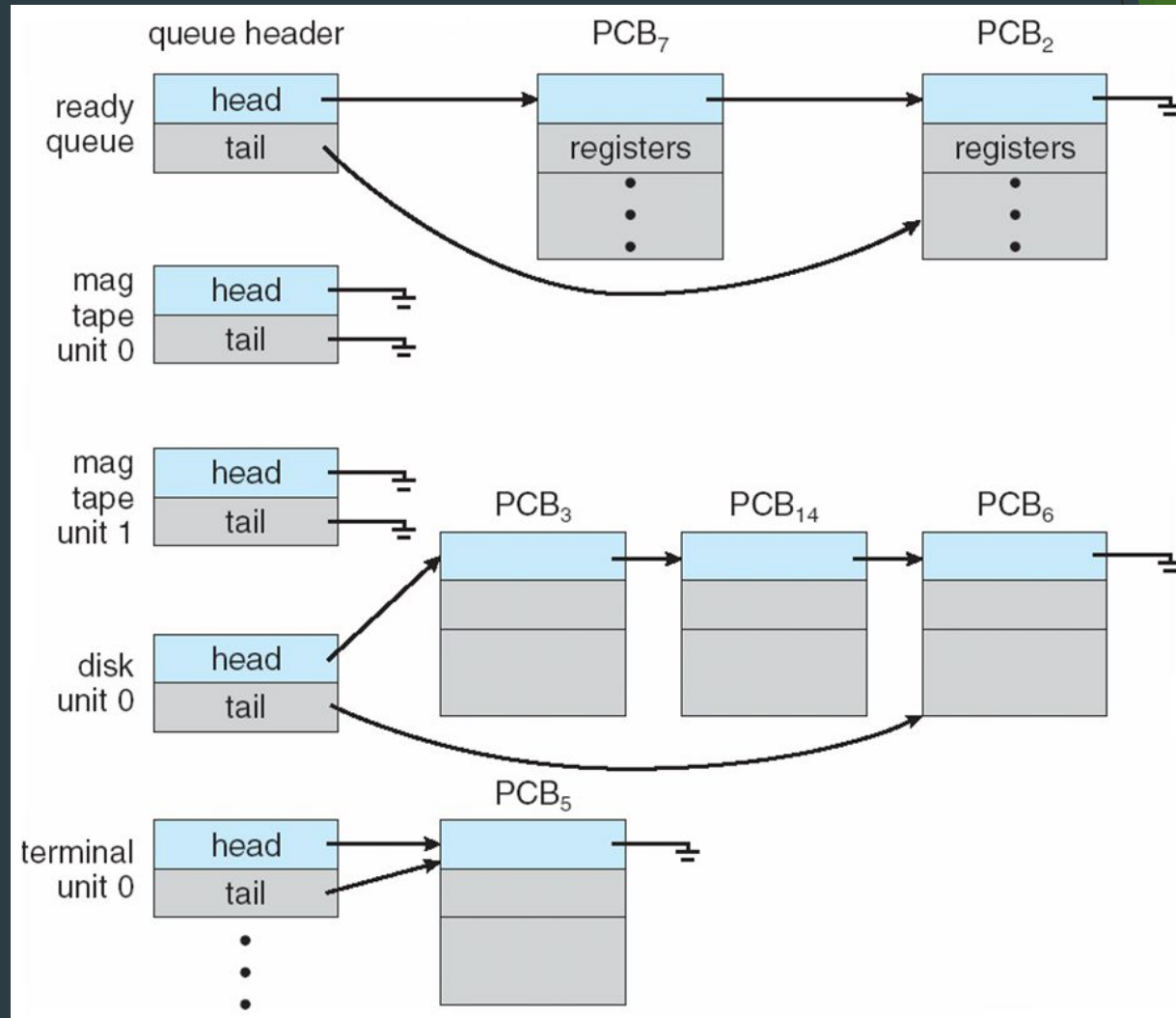
Representation of Process Scheduling

- **Queueing diagram** represents queues, resources, flows
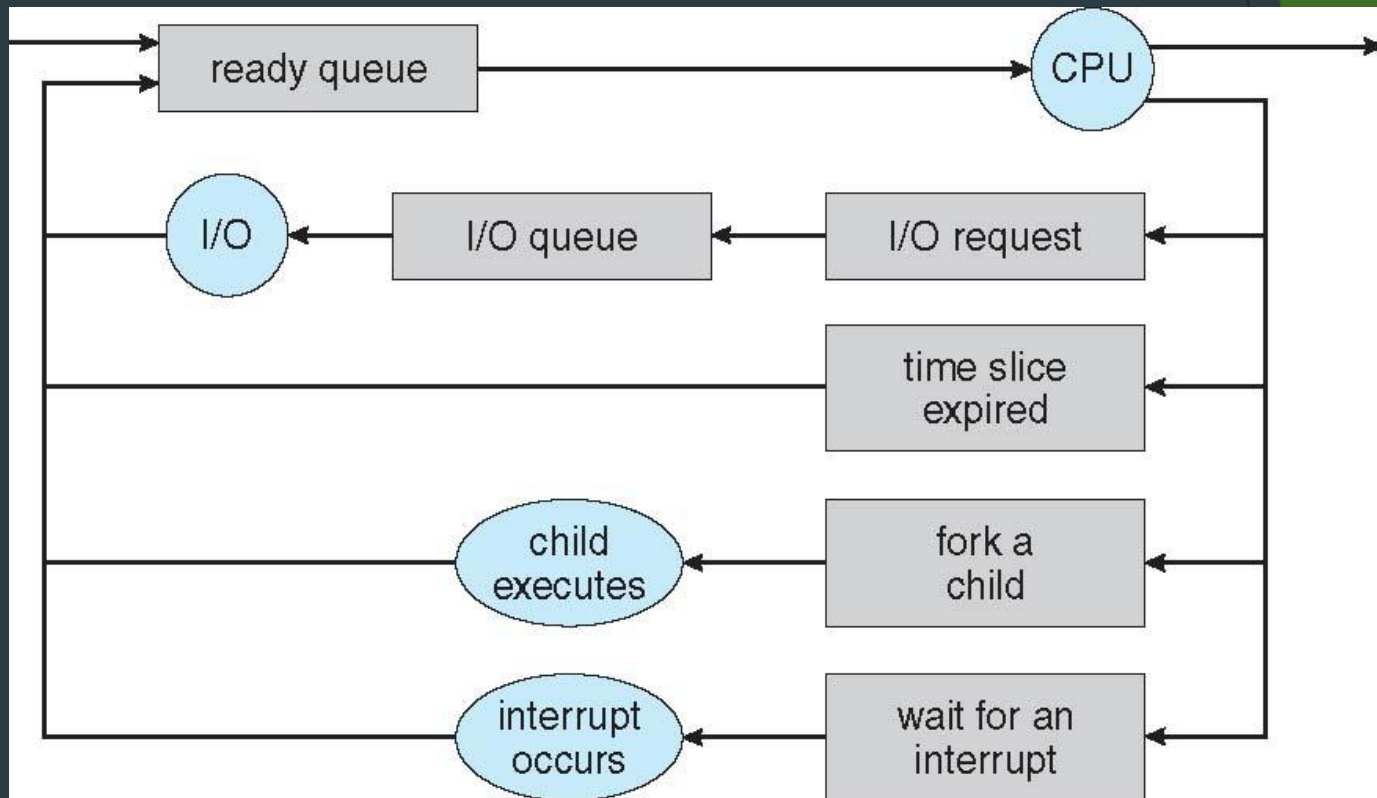
Schedulers

- **Short-term scheduler** (or **CPU scheduler**) – selects which process should be executed next and allocates CPU
    - Sometimes the only scheduler in a system
    - Short-term scheduler is invoked frequently (milliseconds) $\Rightarrow$ (must be fast)
- **Long-term scheduler** (or **job scheduler**) – selects which processes should be brought into the ready queue
    - Long-term scheduler is invoked infrequently (seconds, minutes) $\Rightarrow$ (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, many short CPU bursts
    - **CPU-bound process** – spends more time doing computations; few very long CPU bursts
- Long-term scheduler strives for good *process mix*

Addition of Medium Term Scheduling

- **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**
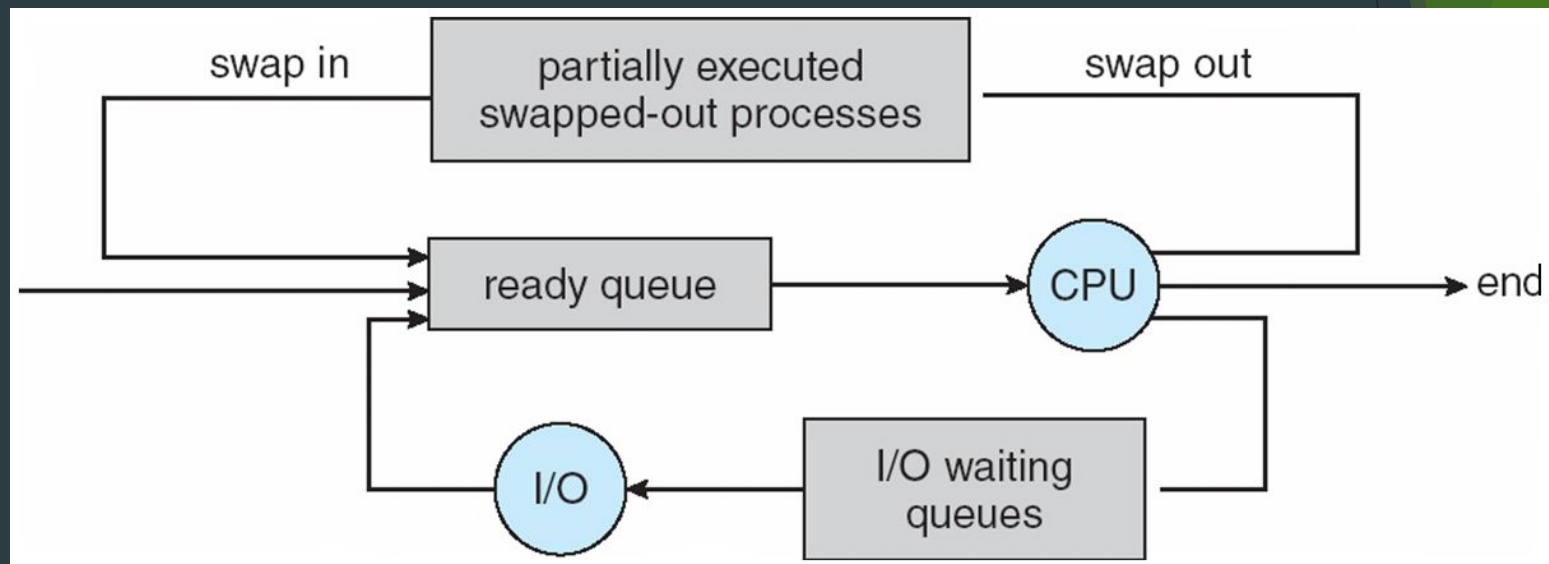
Context Switch

- 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 via a **context switch**

- **Context** of a process represented in the PCB

- Context-switch time is overhead; the system does no useful work while switching

  - The more complex the OS and the PCB ▫ the longer the context switch

- Time dependent on hardware support

  - Some hardware provides multiple sets of registers per CPU ▫ multiple contexts loaded at once