Amirkabir University of Technology
(Tehran Polytechnic)


# Operating Systems

# Processes-Part1

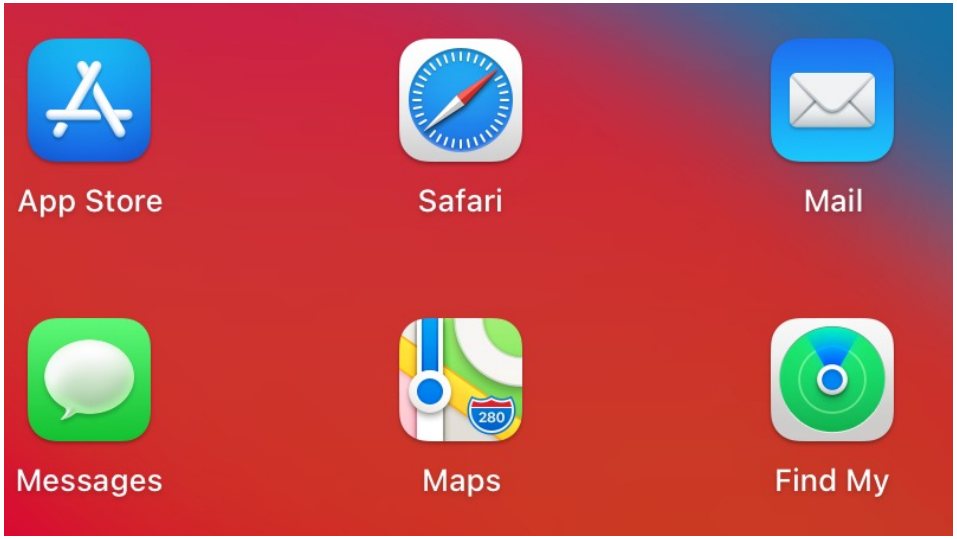
Seyyed Ahmad Javadi

sajavadi@aut.ac.ir

Fall 2021

# Process Concept

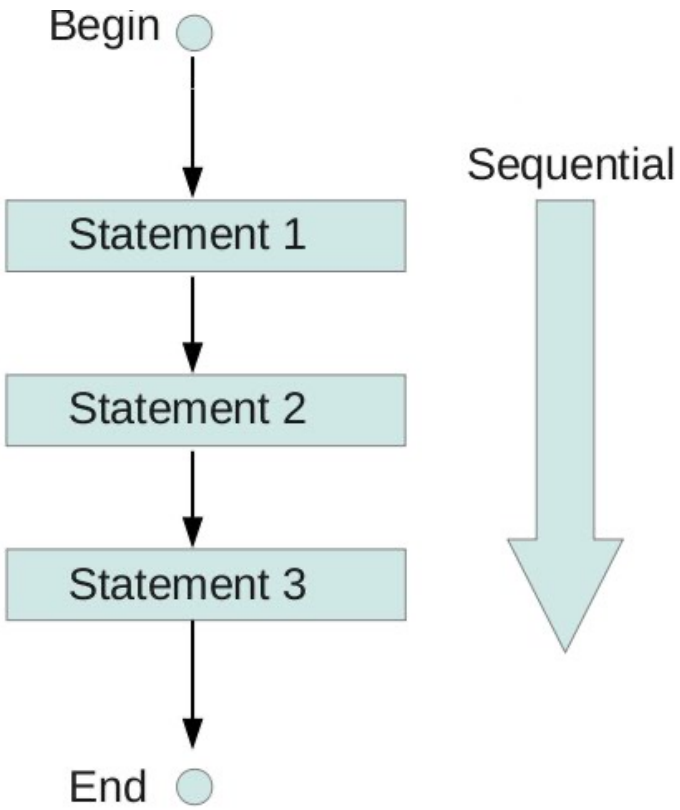
- An OS executes a variety of programs that run as a ***process***.




***Process, a program in execution***

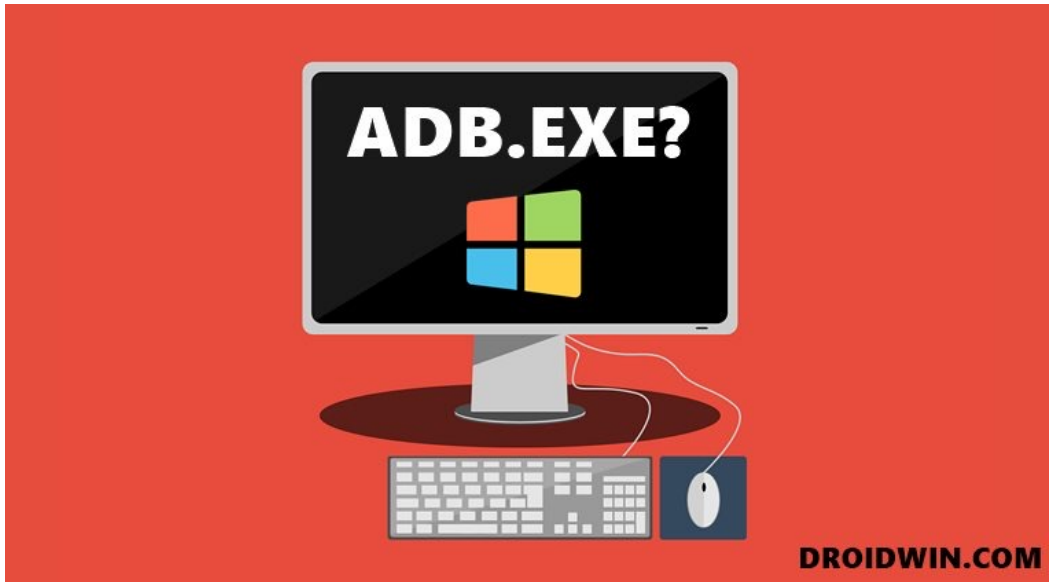
# Process Concept (cont.)

- Process execution must progress in ***sequential fashion***.
  - ***No parallel execution*** of instructions of a single process.



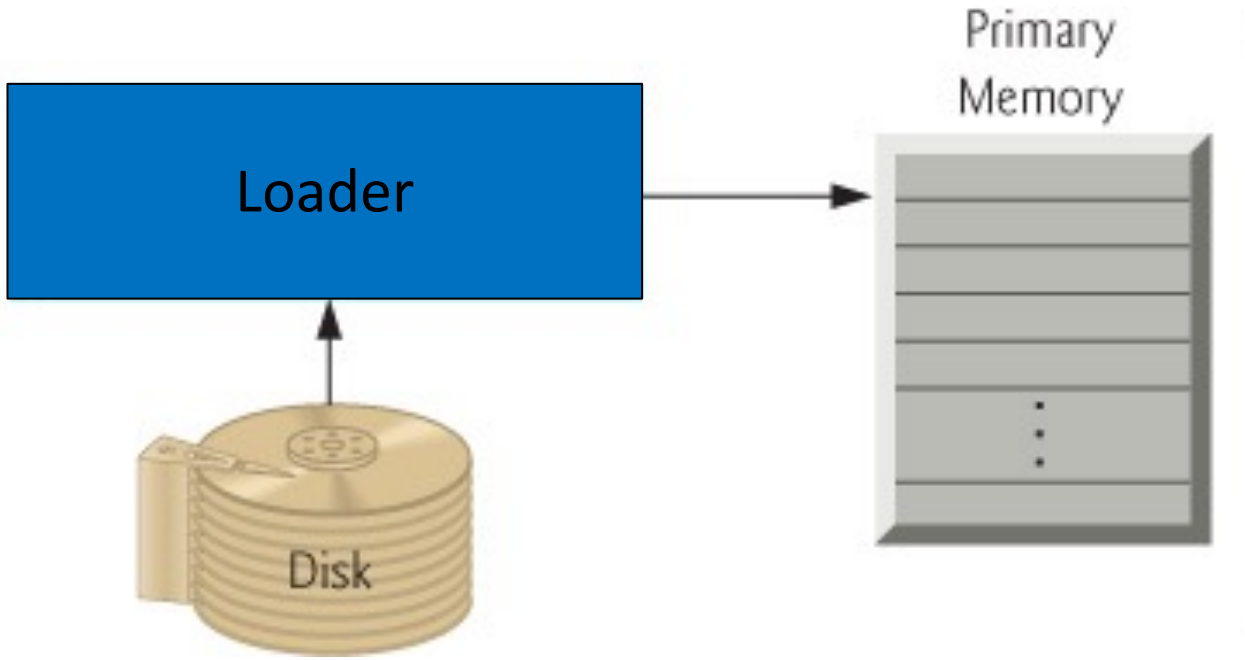
# Process versus Program

- Program is **passive** entity stored on disk (***executable file***).



# Process versus Program (cont.)

- ***Process** is active.*
  - ***Program becomes process*** when an executable file is loaded into memory.



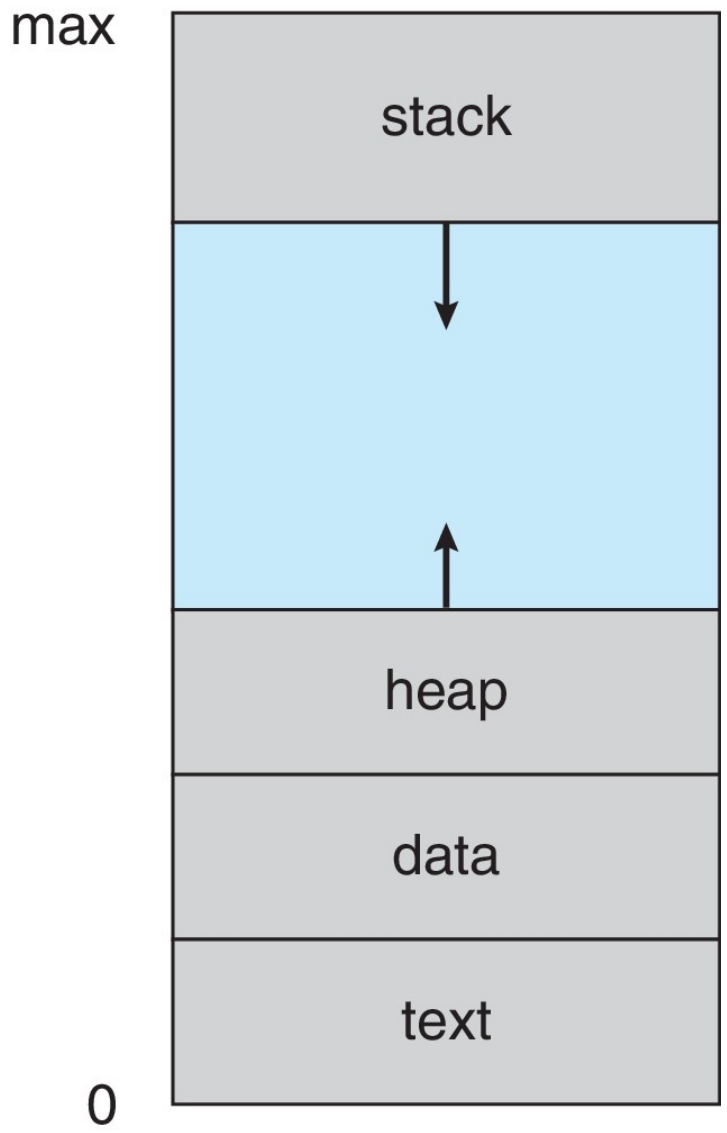
# Process versus Program (cont.)

- 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.

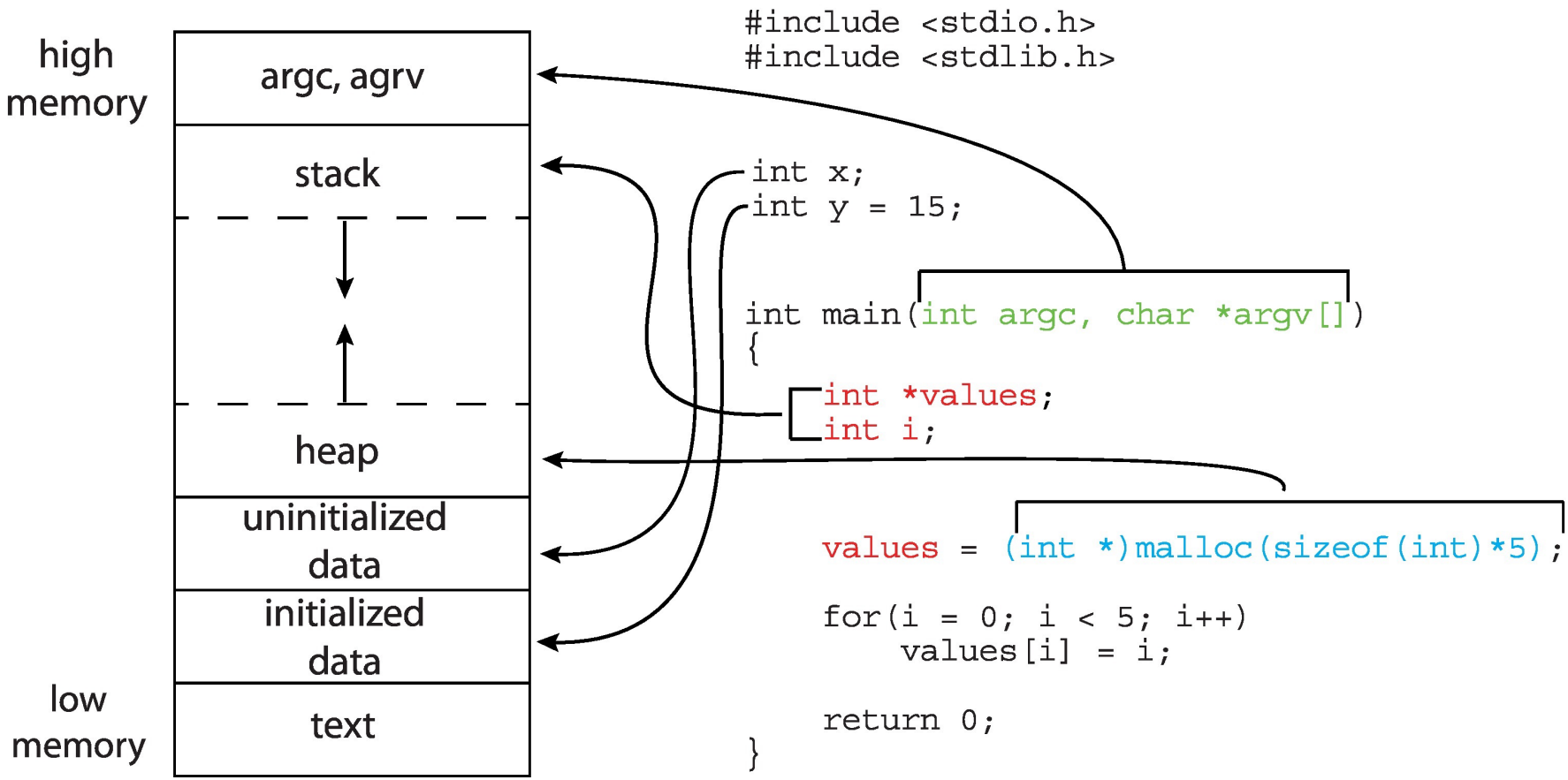
# Multiple Parts of Process

- The program code, also called ***text section***
- Current activity including ***program counter***, processor registers
- ***Stack*** containing temporary data
  - Function parameters, return addresses, local variables
- ***Data section*** containing global variables
- ***Heap*** containing memory dynamically allocated during run time

# Process in Memory



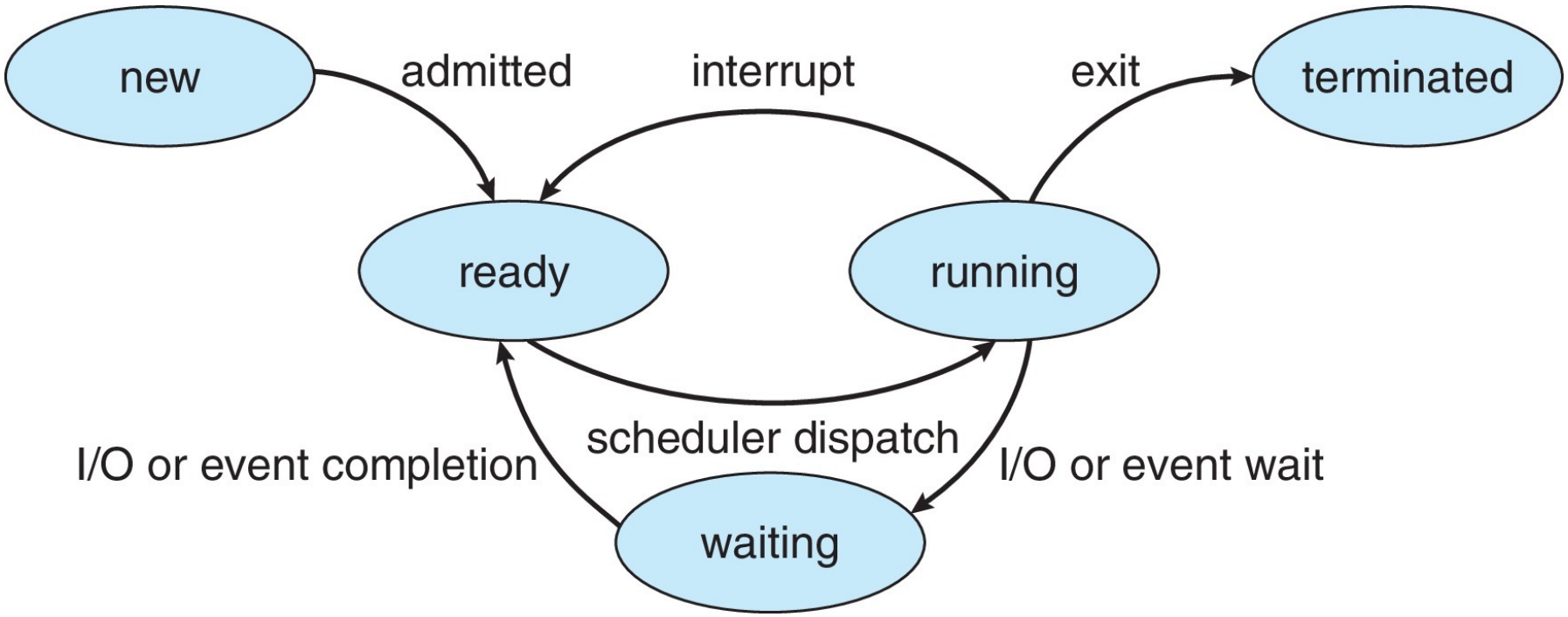
# Memory Layout of a C Program



# 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

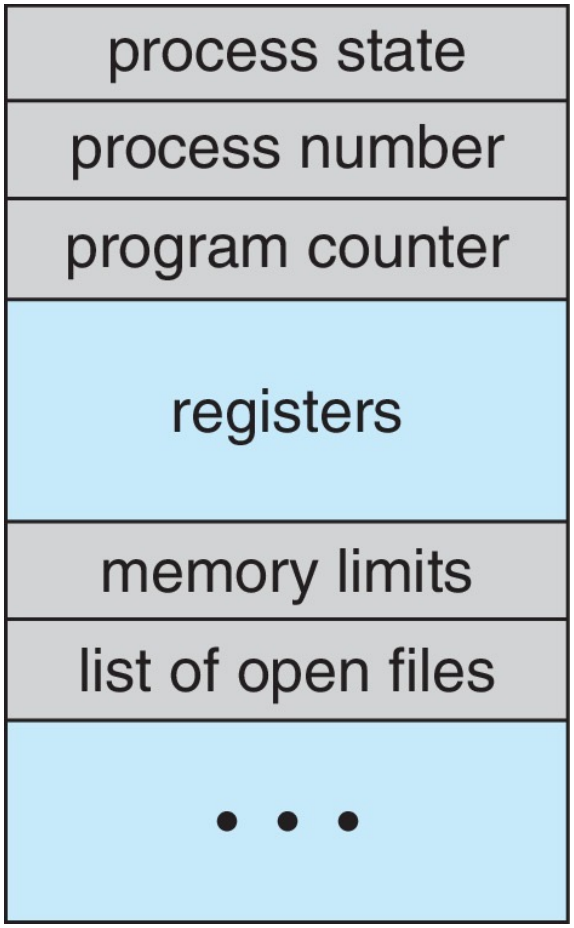


# Process Control Block (PCB)

Information associated with each process

- Process state: running, waiting, etc.
- Program counter: location of instruction to next execute.
- CPU registers: contents of all process-centric registers.
- CPU scheduling information: priorities, scheduling queue pointers.
- Memory-management information: allocated memory
- Accounting information: CPU used, clock time elapsed since start, etc
- I/O status information: allocated I/O devices, list of open files.

# Process Control Block (PCB) (cont.)



# Threads

- So far, process has ***a single thread of execution***.

- Consider having ***multiple program counters*** per process.
  - Multiple locations can execute at once
    - Multiple threads of control -> **threads**

- Must then have ***storage for thread details***
  - Multiple program counters in PCB.

- Explore in detail in Chapter 4.

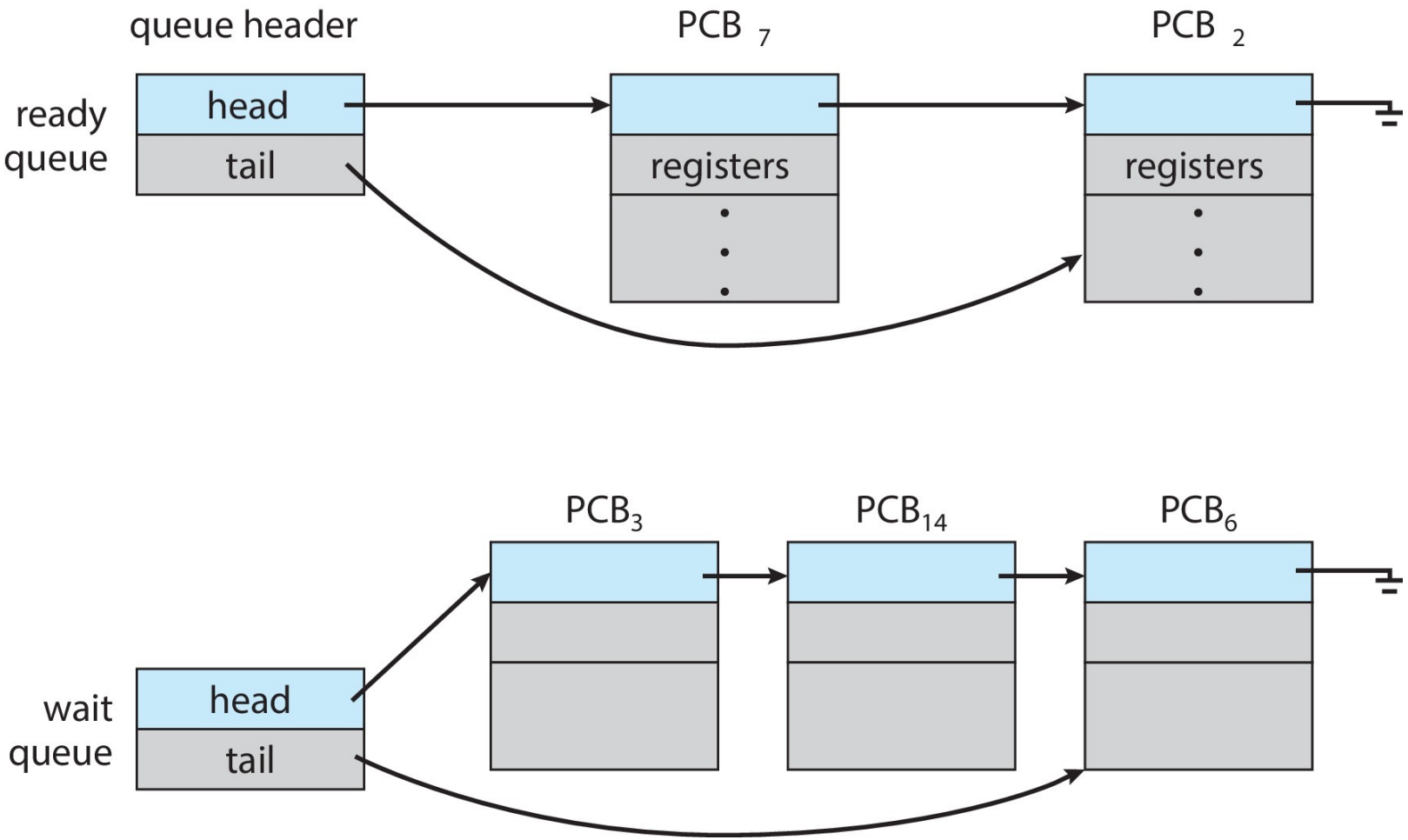
# Process Scheduling

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

- Goal: Maximize CPU use, quickly switch processes onto CPU core.

- Maintains ***scheduling queues*** of processes
  - **Ready queue** – set of all processes residing in main memory, ready and waiting to execute.
  - **Wait queues** – set of processes waiting for an event (i.e., I/O)
  - Processes migrate among the various queues.

# Ready and Wait Queues



# Representation of Process Scheduling