A Five-State Model

- Running
- Ready
- Blocked
- New
- Exit

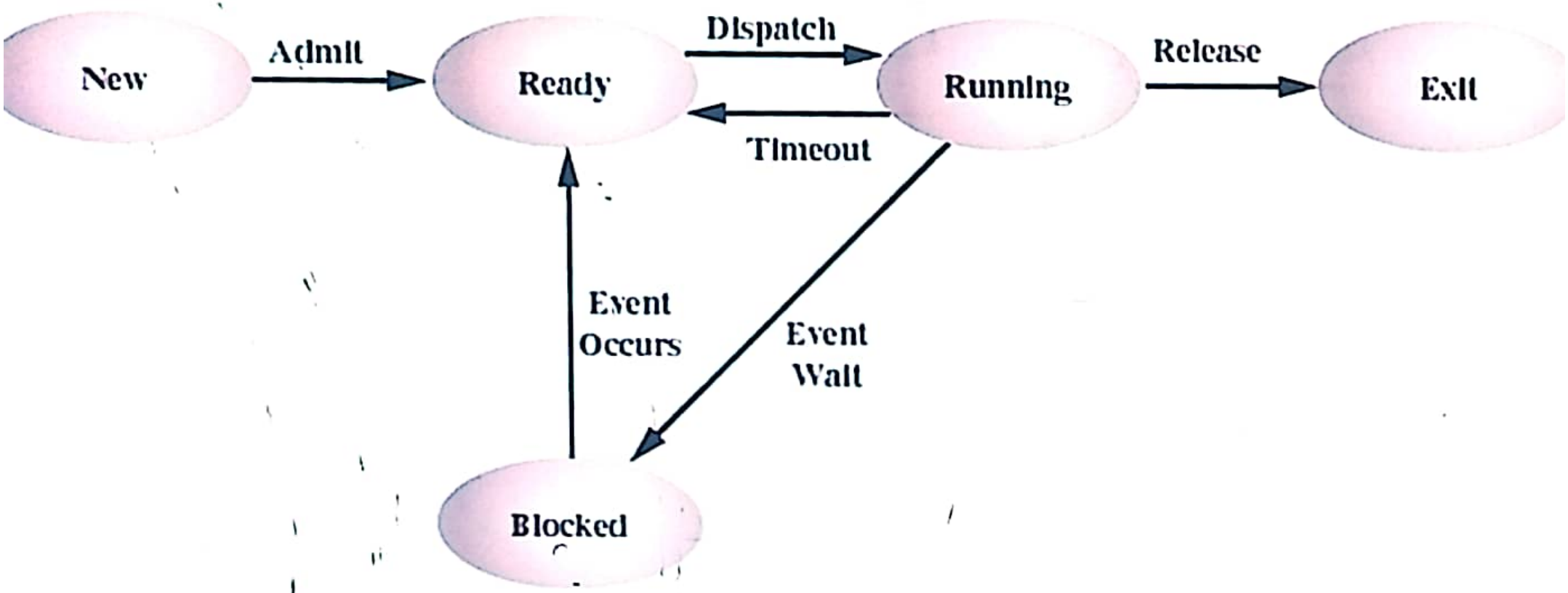


Figure 3.5 Five-State Process Model

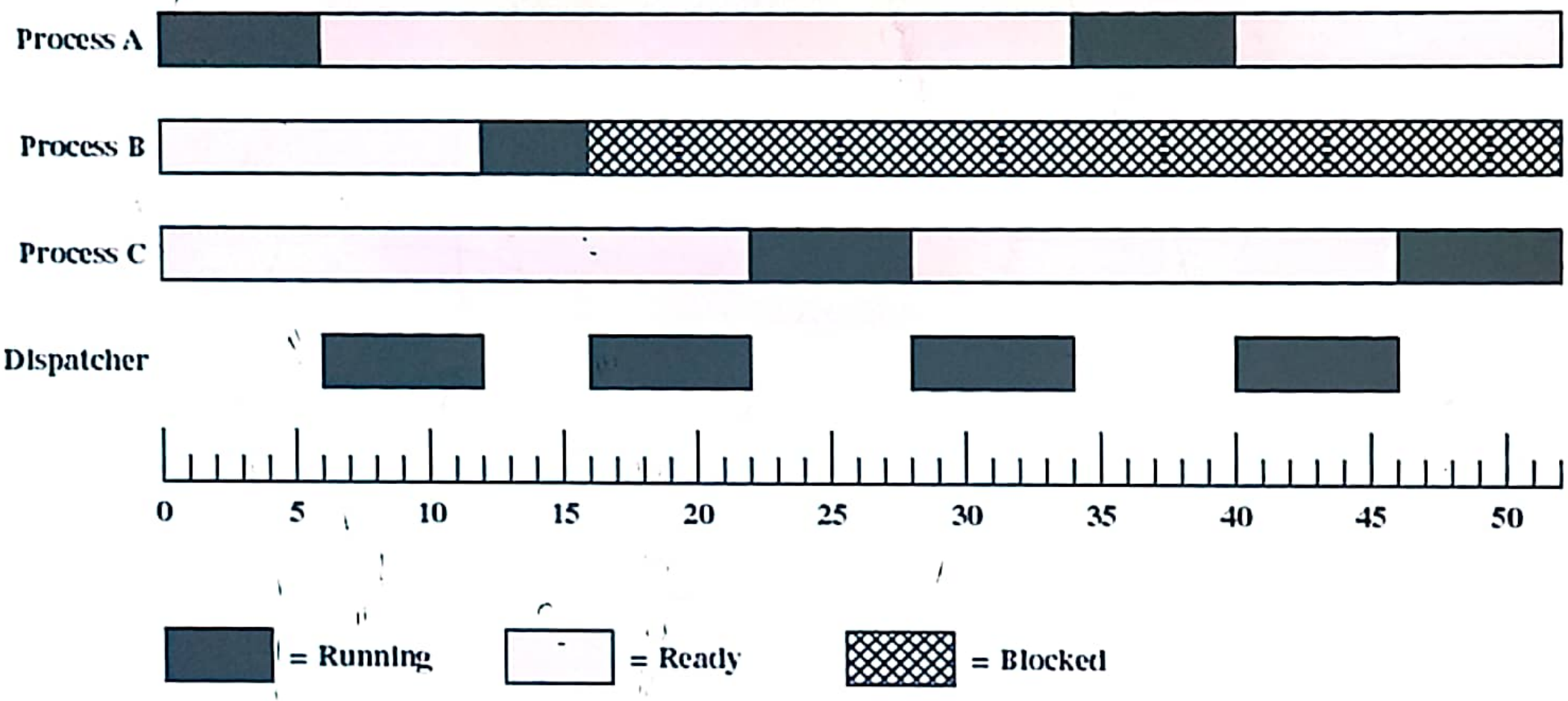
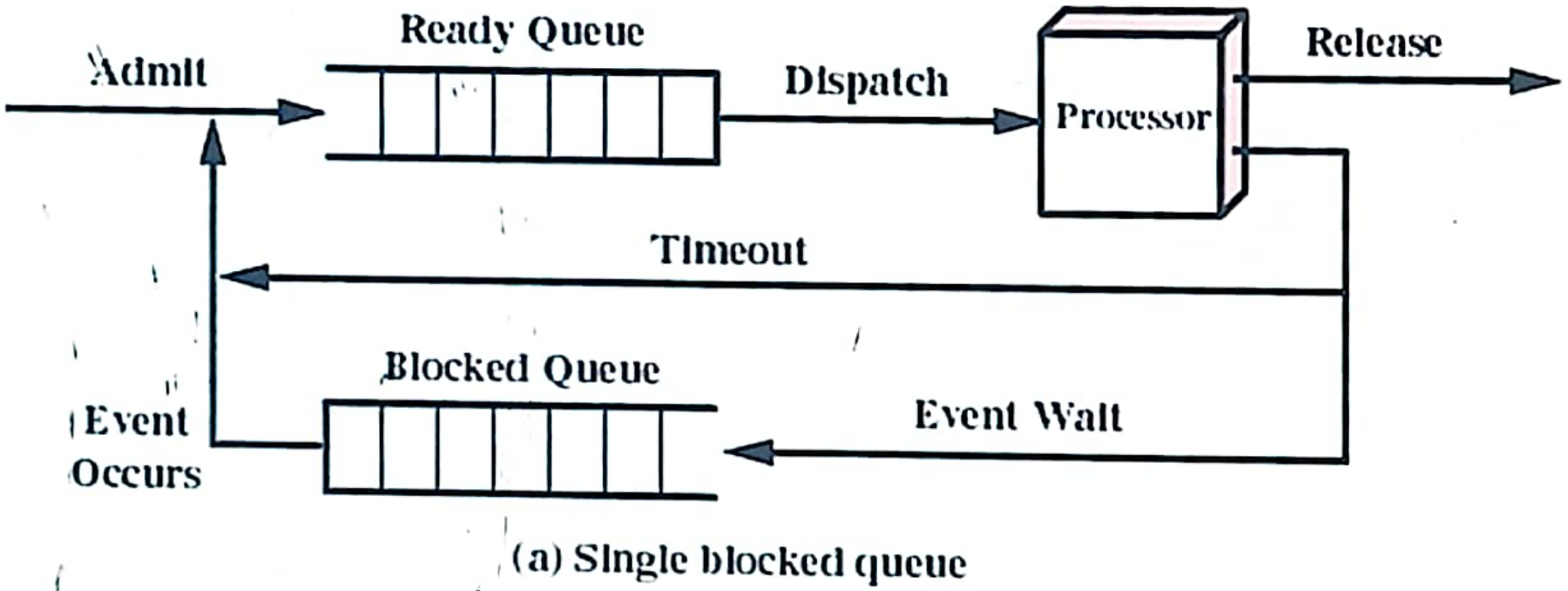
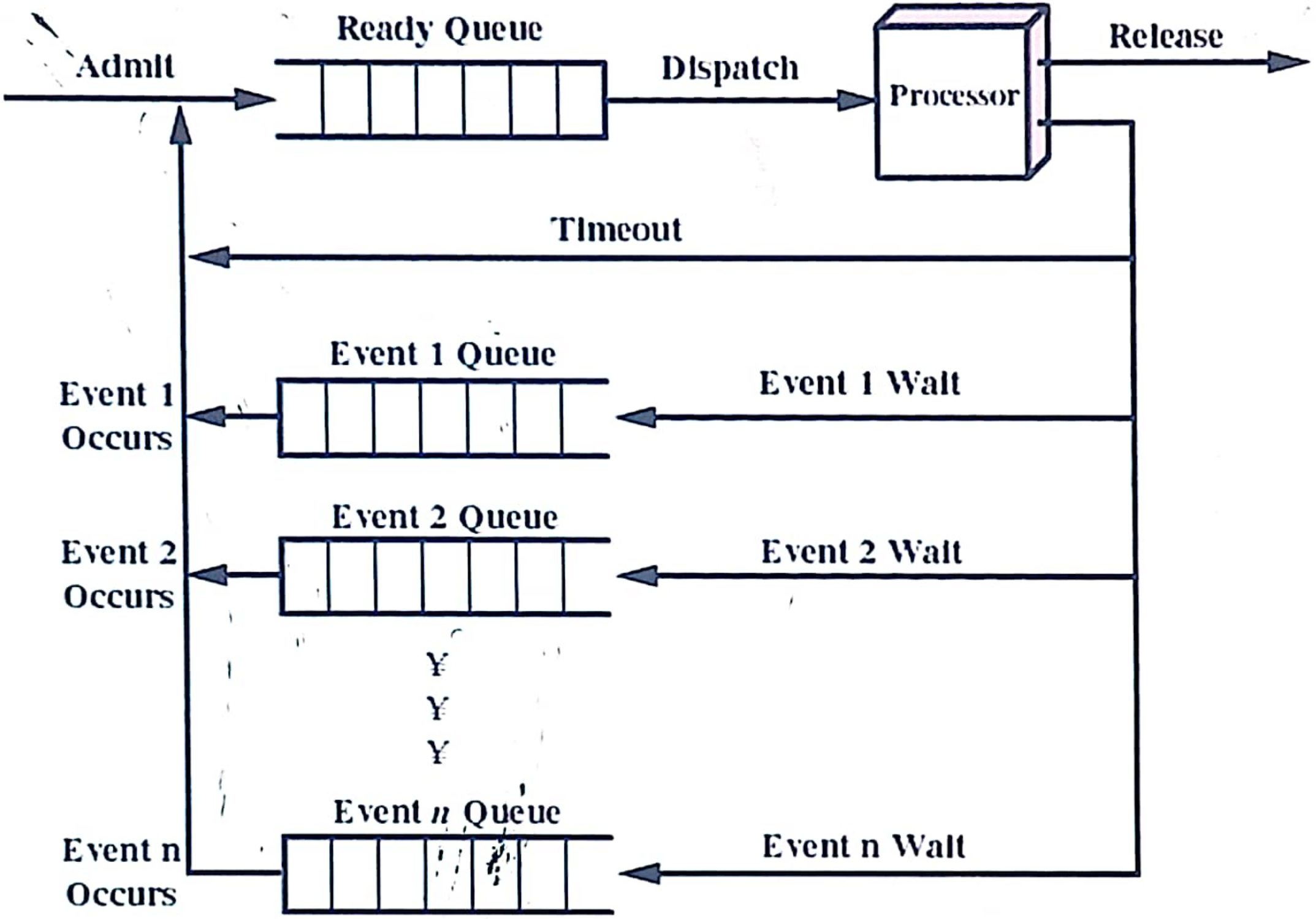


Figure 3.6 Process States for Trace of Figure 3.3

Using Two Queues



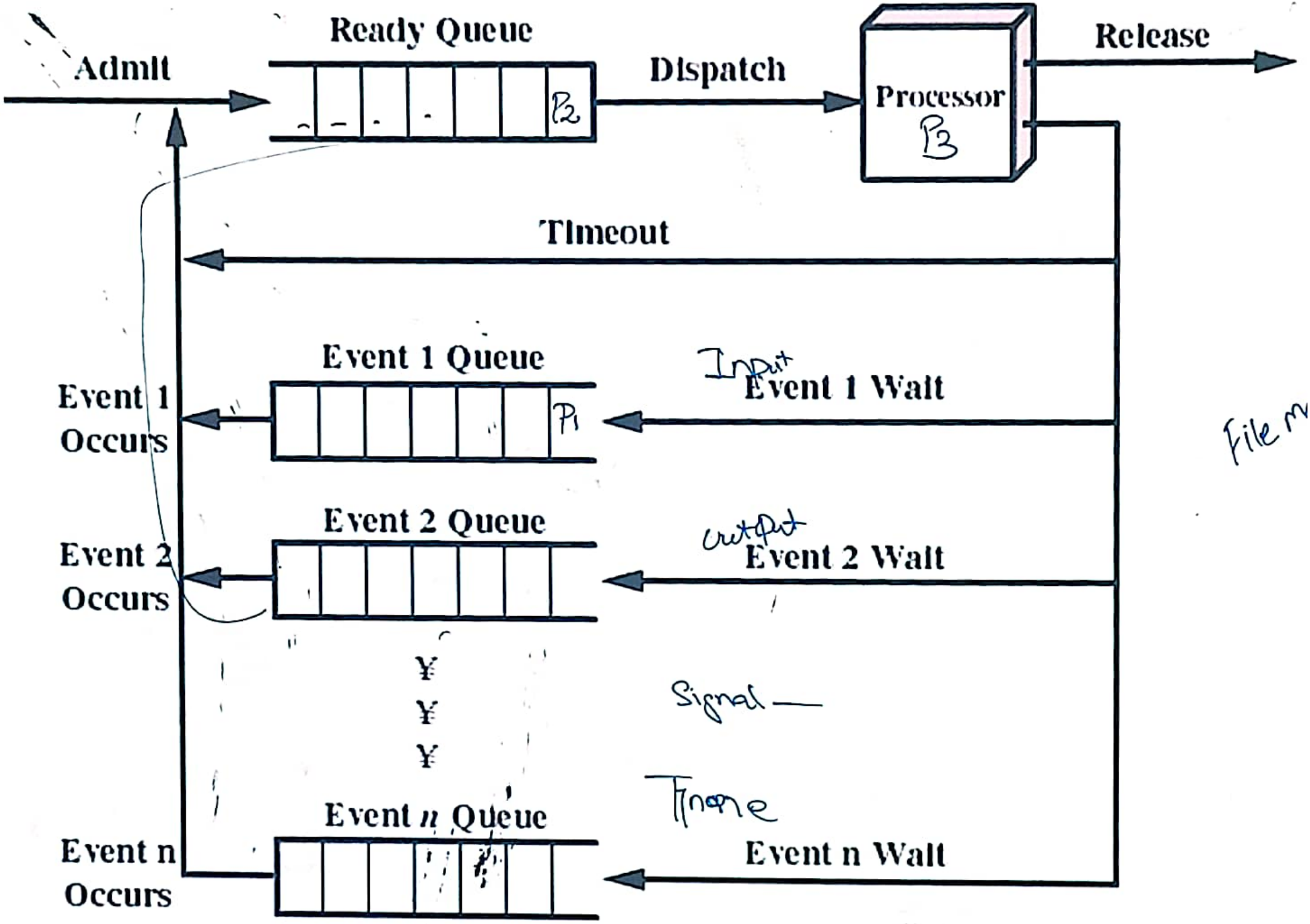
(a) Single blocked queue



(b) Multiple blocked queues

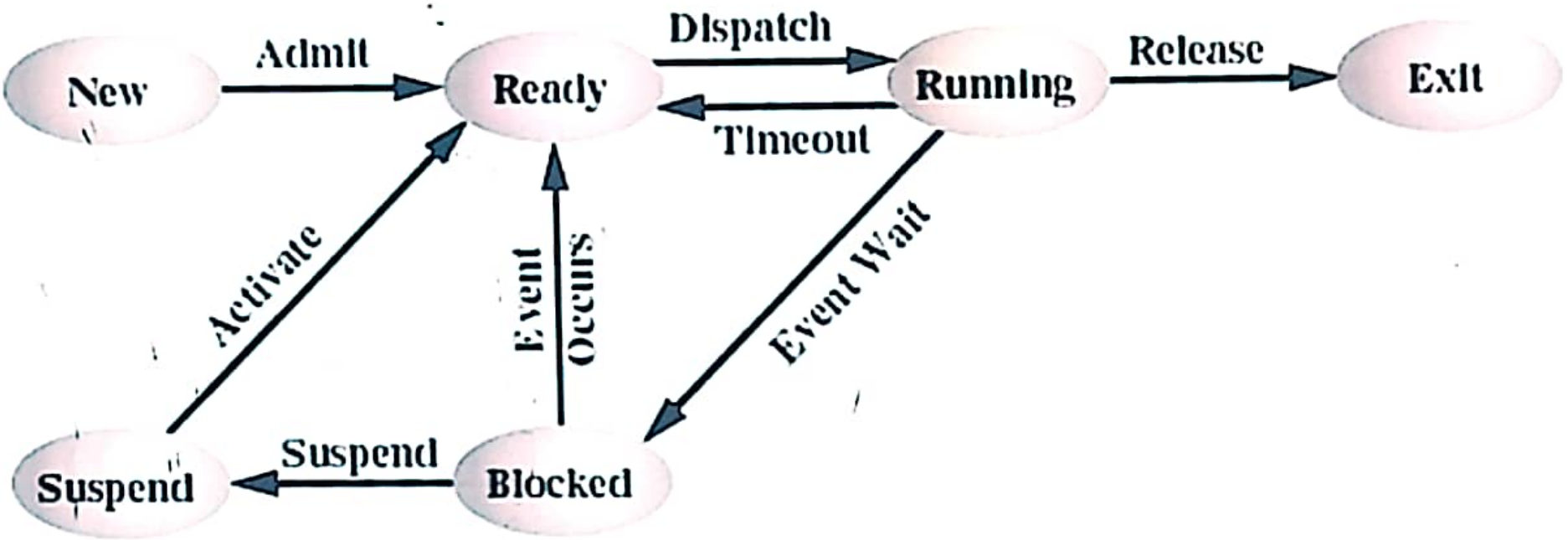
Suspended Processes

- Processor is faster than I/O so all processes could be waiting for I/O
- Swap these processes to disk to free up more memory
- Blocked state becomes suspend state when swapped to disk
- Two new states
  - Blocked, suspend
  - Ready, suspend



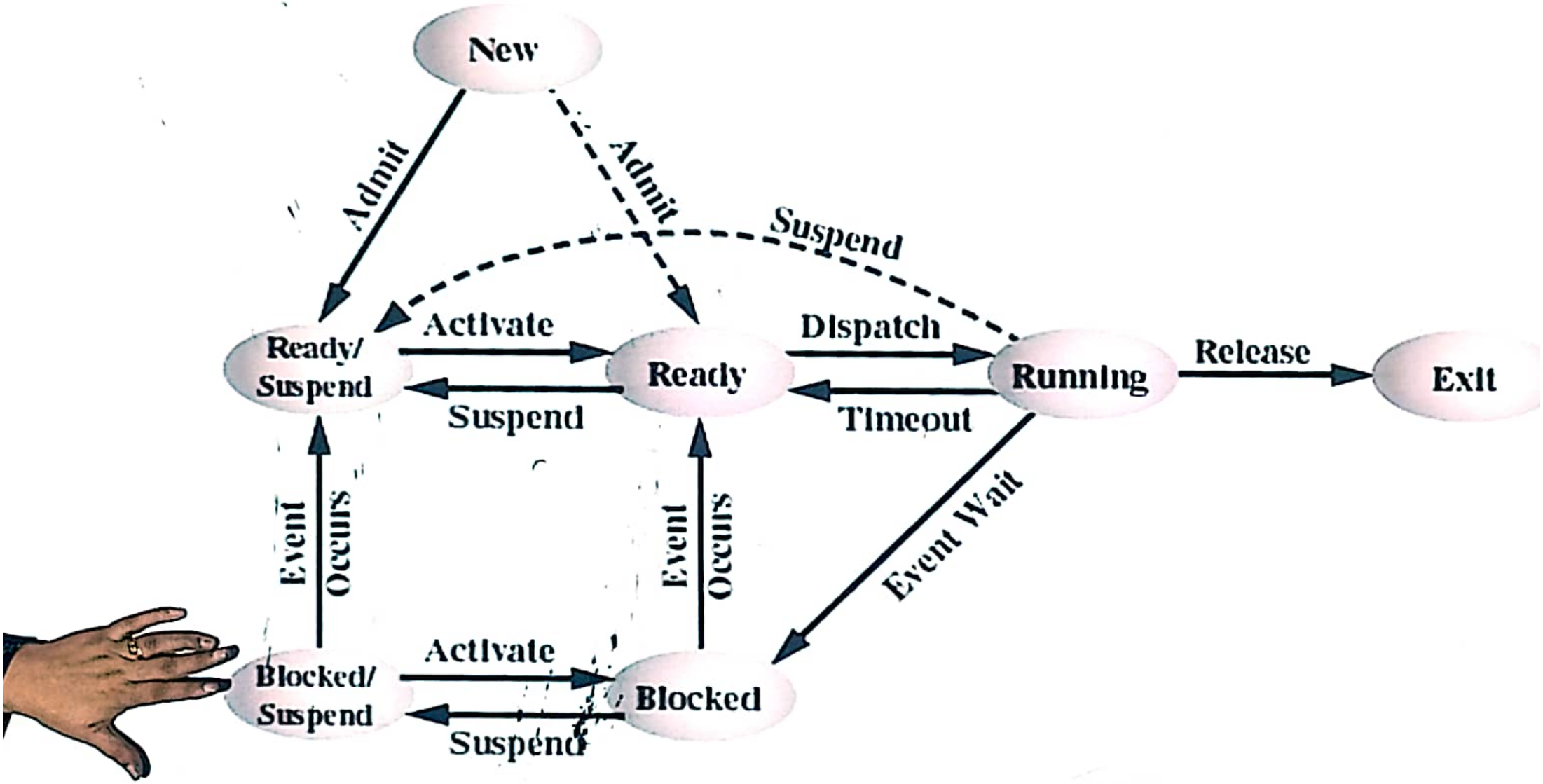
(b) Multiple blocked queues

One Suspend State



(a) With One Suspend State

Two Suspend States



(b) With Two Suspend States

Reasons for Process Suspension

Swapping	The operating system needs to release sufficient main memory to bring in a process that is ready to execute.
Other OS reason	The operating system may suspend a background or utility process or a process that is suspected of causing a problem.
Interactive user request	A user may wish to suspend execution of a program for purposes of debugging or in connection with the use of a resource.
Timing	A process may be executed periodically (e.g., an accounting or system monitoring process) and may be suspended while waiting for the next time interval.
Parent process request	A parent process may wish to suspend execution of a descendent to examine or modify the suspended process, or to coordinate the activity of various descendents.

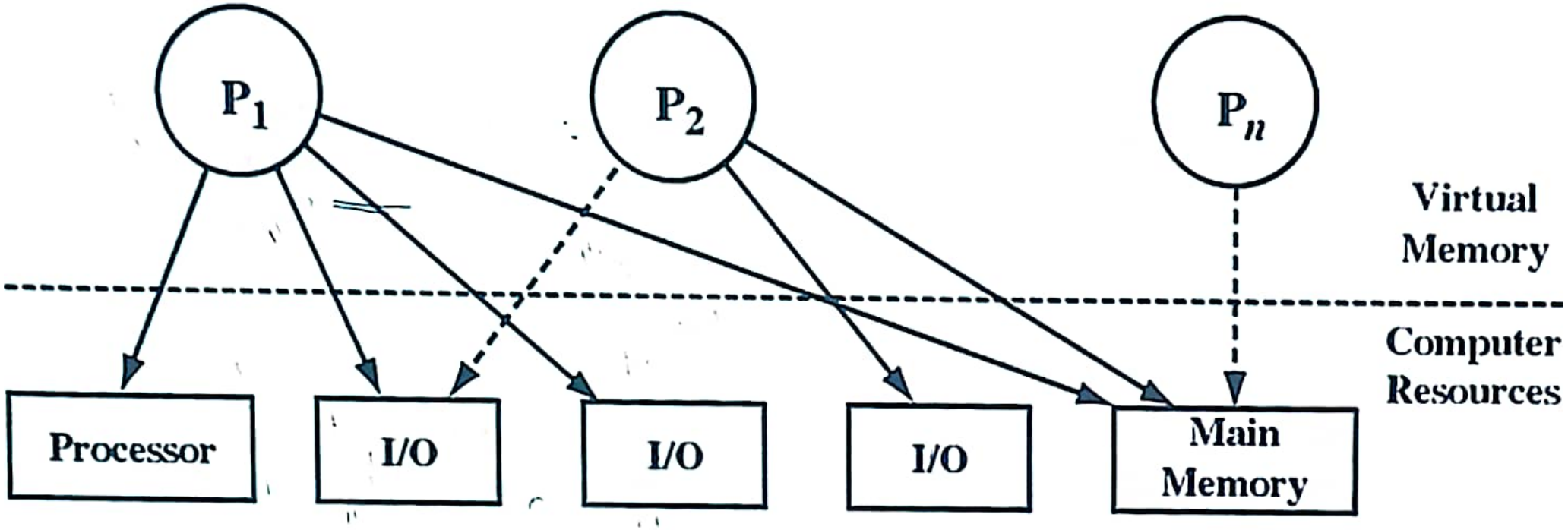


Figure 3.9 Processes and Resources (resource allocation at one snapshot in time)

Operating System Control Structures

- Information about the current status of each process and resource
- Tables are constructed for each entity the operating system manages

Memory Tables

- Allocation of main memory to processes
- Allocation of secondary memory to processes
- Protection attributes for access to shared memory regions
- Information needed to manage virtual memory

I/O Tables

- I/O device is available or assigned
- Status of I/O operation
- Location in main memory being used as the source or destination of the I/O transfer

File Tables

- Existence of files
- Location on secondary memory
- Current Status
- Attributes
- Sometimes this information is maintained by a file-management system

Process Table

- Where process is located
- Attributes necessary for its management
  - Process ID
  - Process state
  - Location in memory

Process Location

- Process includes set of programs to be executed
  - Data locations for local and global variables
  - Any defined constants
  - Stack
- Process control block
  - Collection of attributes
- Process image
  - Collection of program, data, stack, and attributes

Process Control Block

- Process identification
  - Identifiers
    - Numeric identifiers that may be stored with the process control block include
    - Identifier of this process
    - Identifier of the process that created this process (parent process)
    - User identifier

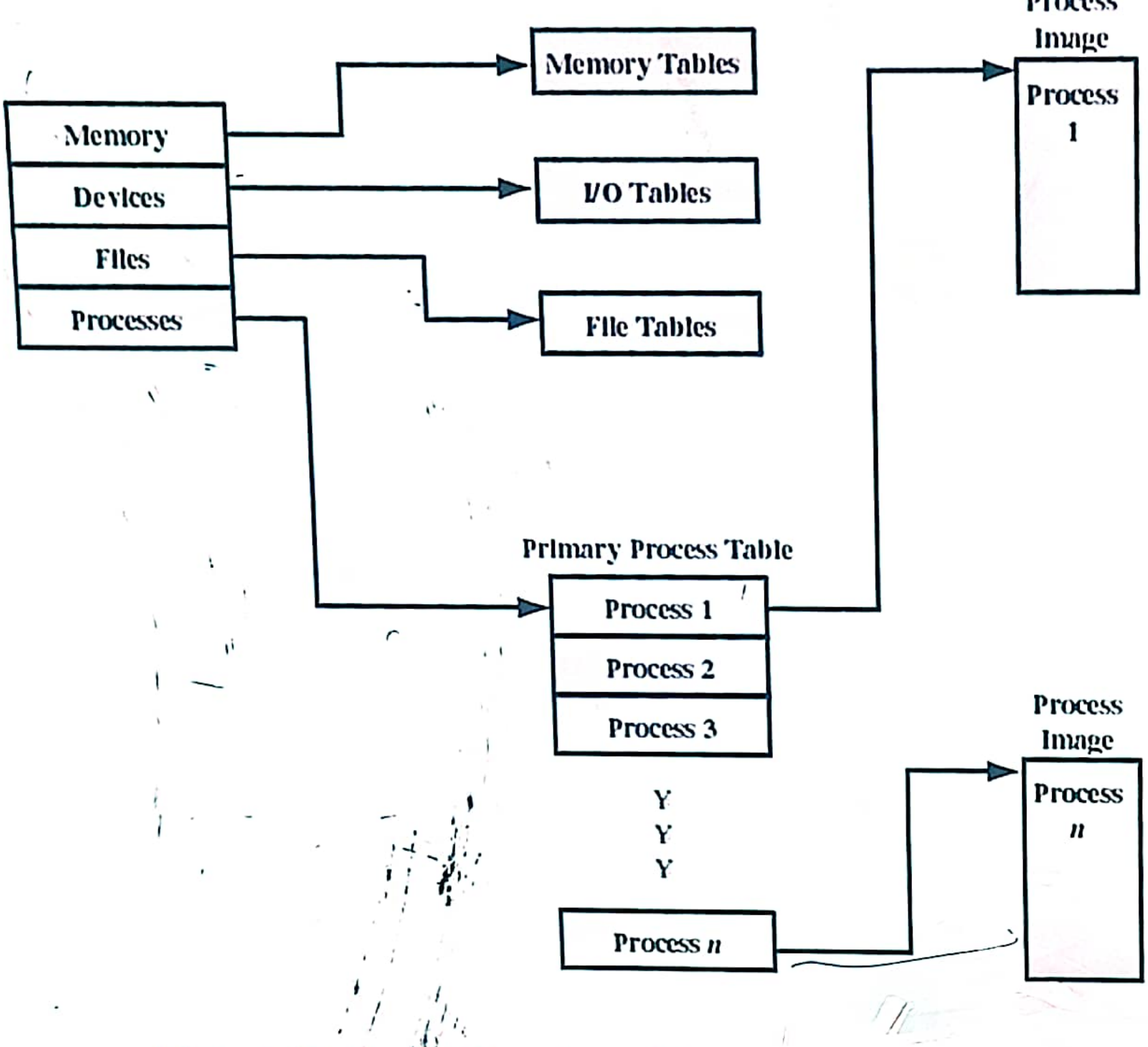


Figure 3.10 General Structure of Operating System Control Tables

Process Control Block

- Processor State Information
  - User-Visible Registers
    - A user-visible register is one that may be referenced by means of the machine language that the processor executes. Typically, there are from 8 to 32 of these registers, although some RISC implementations have over 100.

Process Control Block

- Processor State Information
  - Control and Status Registers

    These are a variety of processor registers that are employed to control the operation of the processor. These include

    - •*Program counter:* Contains the address of the next instruction to be fetched
    - •*Condition codes:* Result of the most recent arithmetic or logical operation (e.g., sign, zero, carry, equal, overflow)

    •*Status information:* Includes interrupt enabled/disabled flags, execution mode

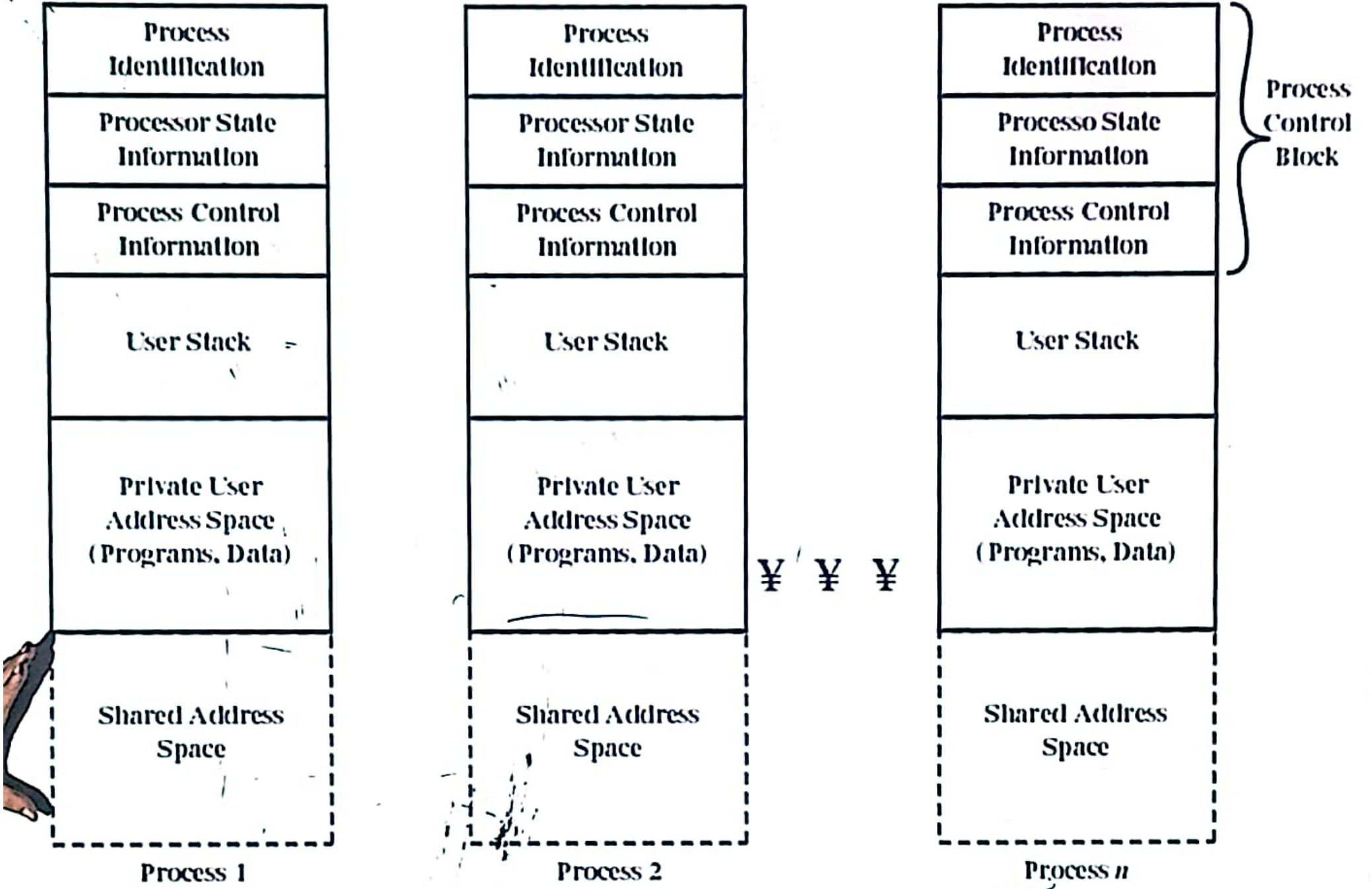


Figure 3.12 User Processes in Virtual Memory

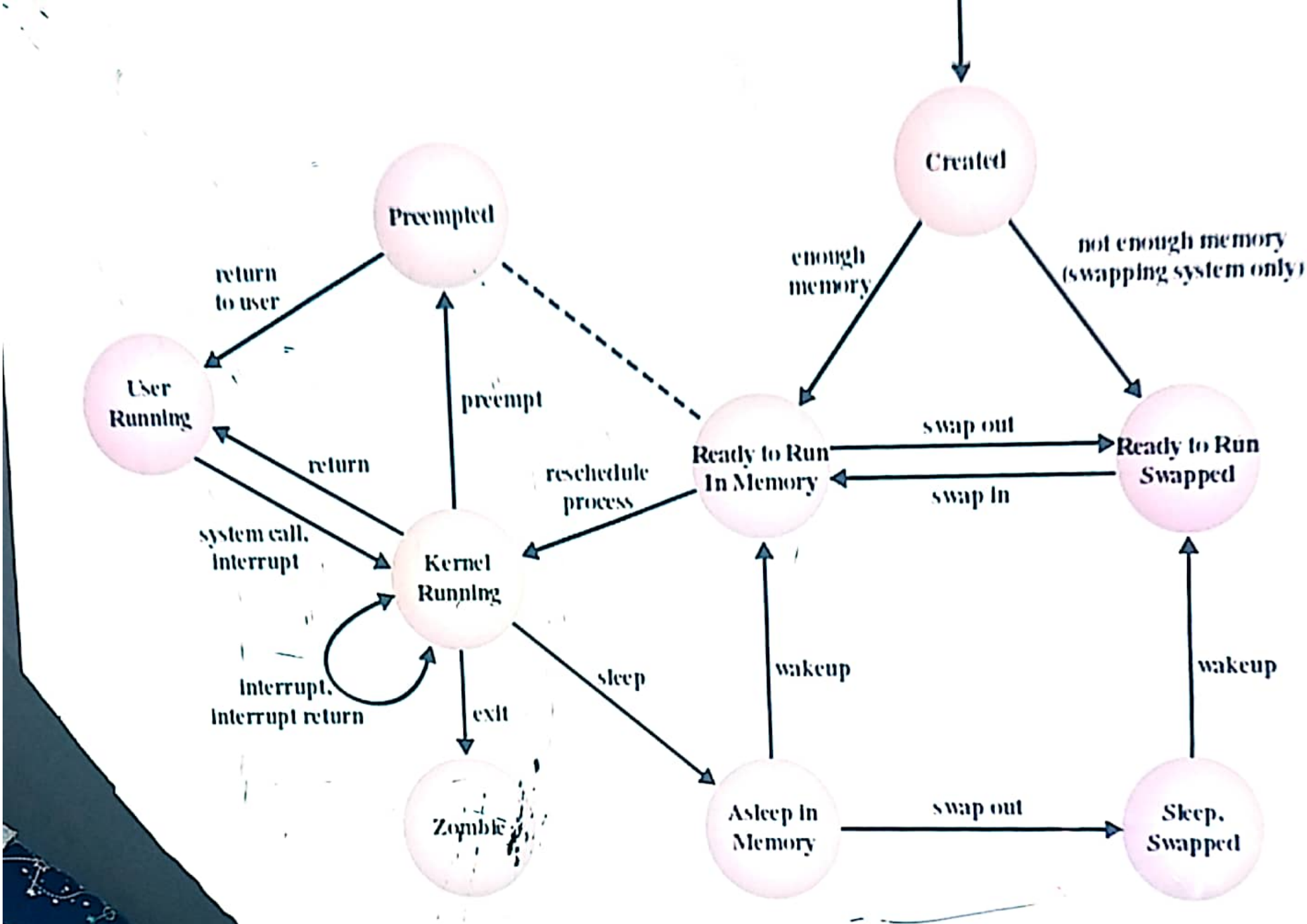


Figure 3.16 UNIX Process State Transition Diagram