Operating Systems

Chapter 3 Process Description and Control

Agenda

- 3.1 What is a Process
- 3.2 Process States
- 3.3 Process Description
- 3.4 Process Control
- 3.5 Summary

Process

1. A program in execution
2. An instance of a program running on a computer
3. The entity that can be assigned to and executed on a processor
4. A unit of activity characterized by：
   ① the execution of a sequence of instructions
   ② a current state
   ③ an associated set of system resources

Process Control Block(进程控制块)

- Contains the process elements
- Created and manage by the operating system
- Allows support for multiple processes

Process Elements

- Identifier
- State
- Priority
- Program counter
- Memory pointers
- Context data
- I/O status information
- Accounting information

Agenda

- 3.1 What is a Process
- 3.2 Process States
- 3.3 Process Description
- 3.4 Process Control
- 3.5 Summary

3.2 Process States

- 3.2.1 Trace of the Process
- 3.2.2 A Two-State Process Model
- 3.2.3 The Creation and Termination of Processes
- 3.2.4 A Five-State Model
- 3.2.5 Suspended Process

Trace of Process(进程轨迹)

- Sequence of instruction that execute for a process
- Dispatcher(调度器) switches the processor from one process to another

Example Execution

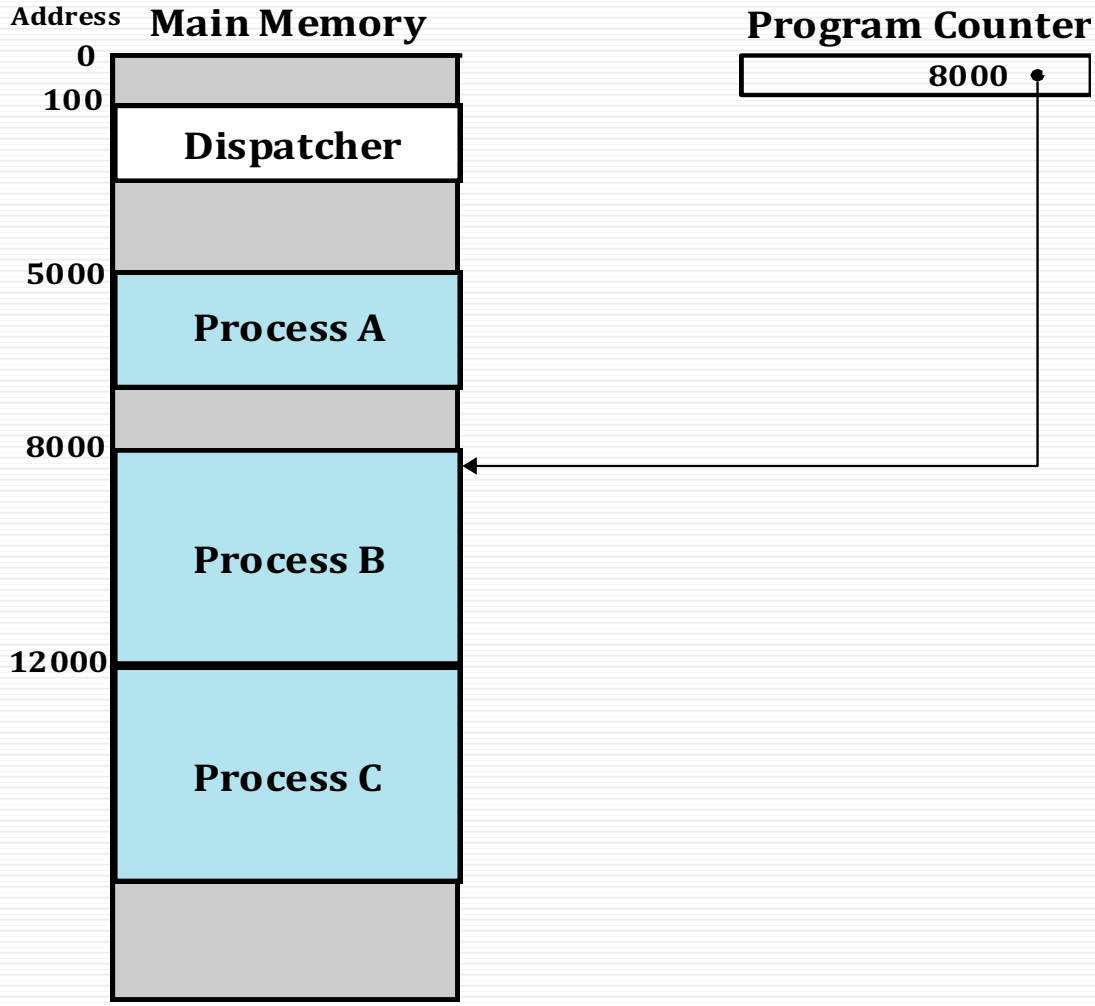


Figure 3.2 Snapshot of Example Execution (Figure 3 at Instruction Cycle 13

Trace of Processes

(a) Trace of Process A	(b) Trace of Process B	(c) Trace of Process C
5000	8000	12000
5001	8001	12001
5002	8002	12002
5003	8003	12003
5004		12004
5005		12005
5006		12006
5007		12007
5008		12008
5009		12009
5010		12010
5011		12011

5000 = Starting address of program of Process A
8000 = Starting address of program of Process B
12000 = Starting address of program of Process C

Figure 3.3 Traces of Processes of Figure 3.2

Trace of Processes

1	5000	
2	5001	
3	5002	
4	5003	
5	5004	
6	5005	
----------		Time out
7	100	
8	101	
9	102	
10	103	
11	104	
12	105	
13	8000	
14	8001	
15	8002	
16	8003	
----------		I/O request
17	100	
18	101	
19	102	
20	103	
21	104	
22	105	
23	12000	
24	12001	
25	12002	
26	12003	
27	12004	
28	12005	
----------		Time out
29	100	
30	101	
31	102	
32	103	
33	104	
34	105	
35	5006	
36	5007	
37	5008	
38	5009	
39	5010	
40	5011	
----------		Time out
41	100	
42	101	
43	102	
44	103	
45	104	
46	105	
47	12006	
48	12007	
49	12008	
50	12009	
51	12010	
52	12011	
----------		Time out

100 = Starting address of dispatcher program

shaded areas indicate execution of dispatcher process;
first and third columns count instruction cycles;
second and fourth columns show address of instruction being executed

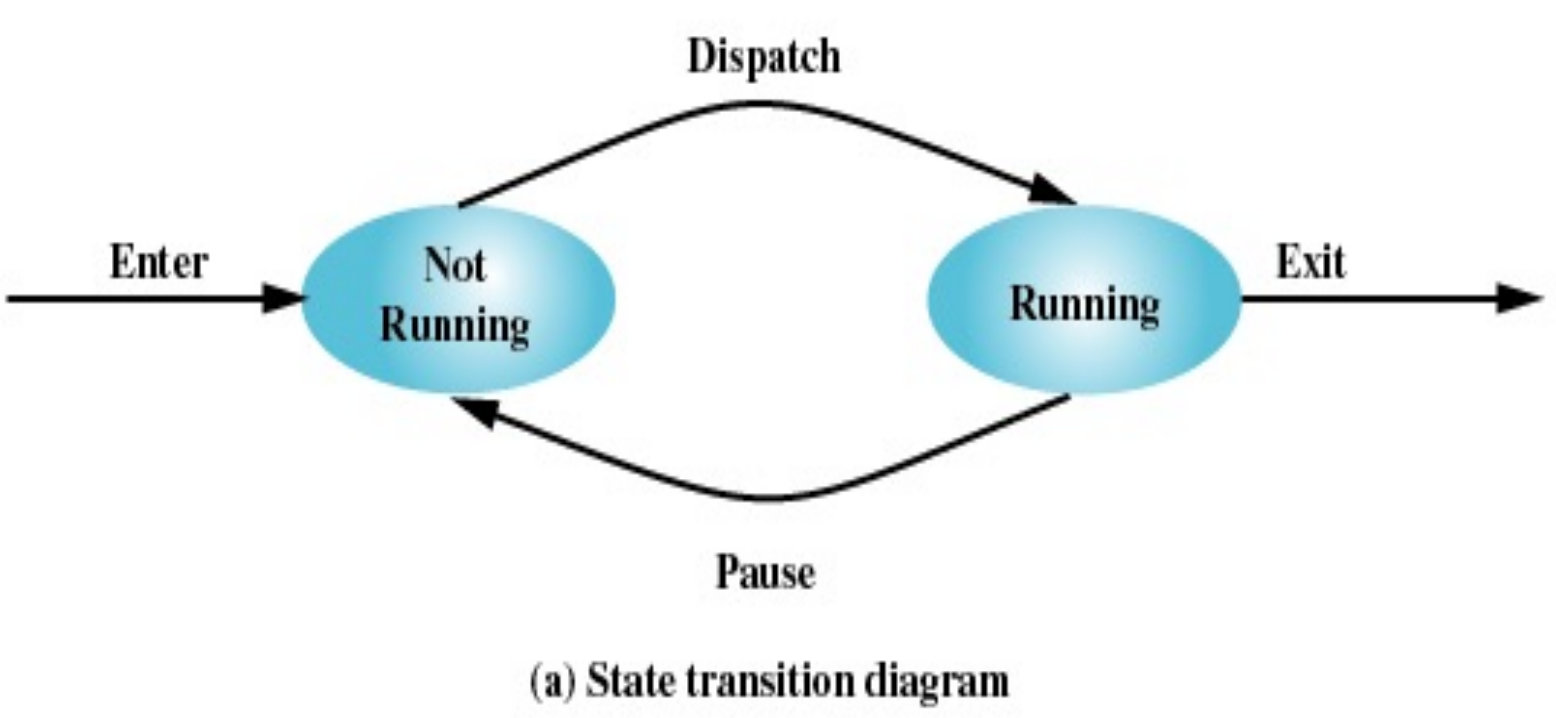
Figure 3.4 Combined Trace of Processes of Figure 3.2

3.2 Process States

- 3.2.1 Trace of the Process
- 3.2.2 A Two-State Process Model
- 3.2.3 The Creation and Termination of Processes
- 3.2.4 A Five-State Model
- 3.2.5 Suspended Process

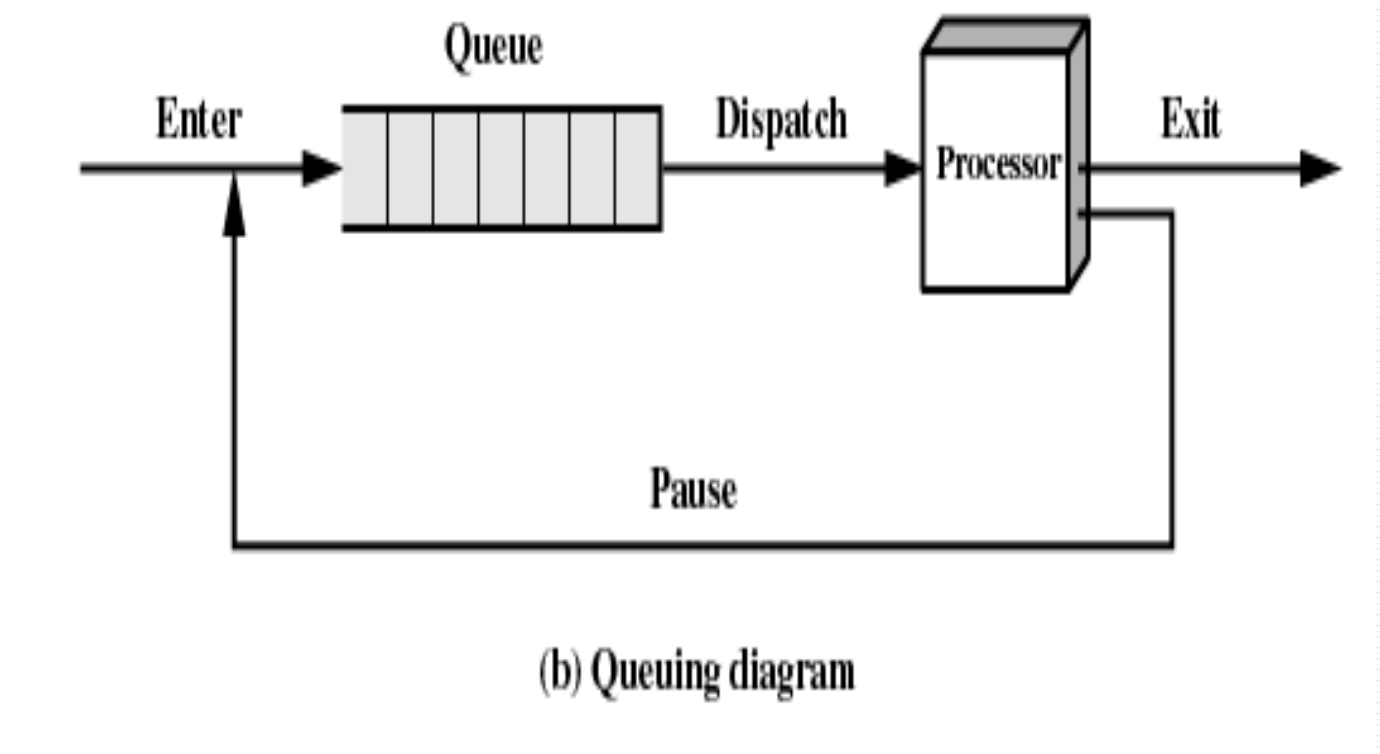
Two-State Process Model

- Process may be in one of two states
  - Running
  - Not-running



(a) State transition diagram

Not-Running Process in a Queue



(b) Queuing diagram

3.2 Process States

- 3.2.1 Trace of the Process
- 3.2.2 A Two-State Process Model
- 3.2.3 The Creation and Termination of Processes
- 3.2.4 A Five-State Model
- 3.2.5 Suspended Process

Process Creation

Table 3.1 Reasons for Process Creation

New batch job	The operating system is provided with a batch job control stream, usually on tape or disk. When the operating system is prepared to take on new work, it will read the next sequence of job control commands.
Interactive logon	A user at a terminal logs on to the system.
Created by OS to provide a service	The operating system can create a process to perform a function on behalf of a user program, without the user having to wait (e.g., a process to control printing).
Spawned by existing process	For purposes of modularity or to exploit parallelism, a user program can dictate the creation of a number of processes.

Process Termination

Table 3.2 Reasons for Process Termination

Normal completion	The process executes an OS service call to indicate that it has completed running.
Time limit exceeded	The process has run longer than the specified total time limit. There are a number of possibilities for the type of time that is measured. These include total elapsed time ("wall clock time"), amount of time spent executing, and, in the case of an interactive process, the amount of time since the user last provided any input.
Memory unavailable	The process requires more memory than the system can provide.
Bounds violation	The process tries to access a memory location that it is not allowed to access.
Protection error	The process attempts to use a resource such as a file that it is not allowed to use, or it tries to use it in an improper fashion, such as writing to a read-only file.
Arithmetic error	The process tries a prohibited computation, such as division by zero, or tries to store numbers larger than the hardware can accommodate.

Process Termination

Table 3.2 Reasons for Process Termination

Time overrun	The process has waited longer than a specified maximum for a certain event to occur.
I/O failure	An error occurs during input or output, such as inability to find a file, failure to read or write after a specified maximum number of tries (when, for example, a defective area is encountered on a tape), or invalid operation (such as reading from the line printer).
Invalid instruction	The process attempts to execute a nonexistent instruction (often a result of branching into a data area and attempting to execute the data).
Privileged instruction	The process attempts to use an instruction reserved for the operating system.
Data misuse	A piece of data is of the wrong type or is not initialized.
Operator or OS intervention	For some reason, the operator or the operating system has terminated the process (for example, if a deadlock exists).
Parent termination	When a parent terminates, the operating system may automatically terminate all of the offspring of that parent.
Parent request	A parent process typically has the authority to terminate any of its offspring.

3.2 Process States

- 3.2.1 Trace of the Process
- 3.2.2 A Two-State Process Model
- 3.2.3 The Creation and Termination of Processes
- 3.2.4 A Five-State Model
- 3.2.5 Suspended Process

Limit of Two-State Process Model

- Not-running
  - ready to execute
  - waiting for I/O (blocked)
- Dispatcher cannot just select the process that has been in the queue the longest because it may be blocked

A Five-State Model

- Running(运行态)
- Ready(就绪态)
- Blocked(阻塞态)
- New(新建态)
- Exit(退出态)

Five-State Process Model

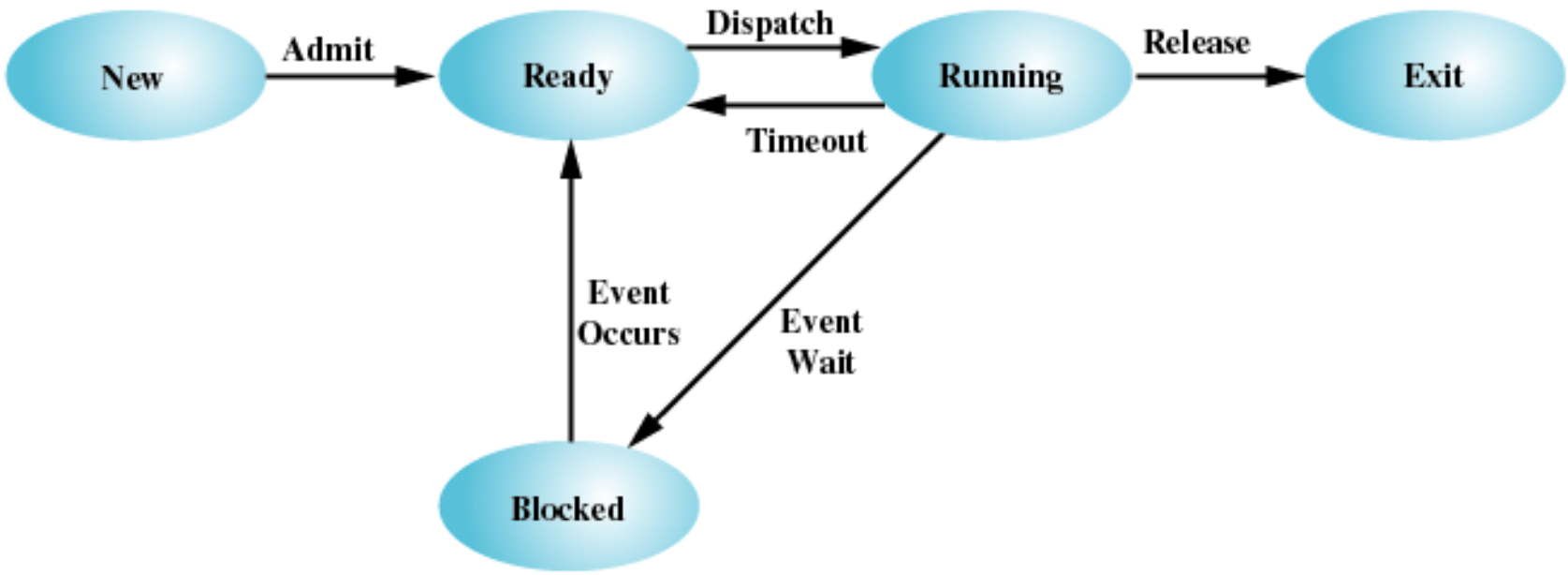


Figure 3.6 Five-State Process Model

Process States

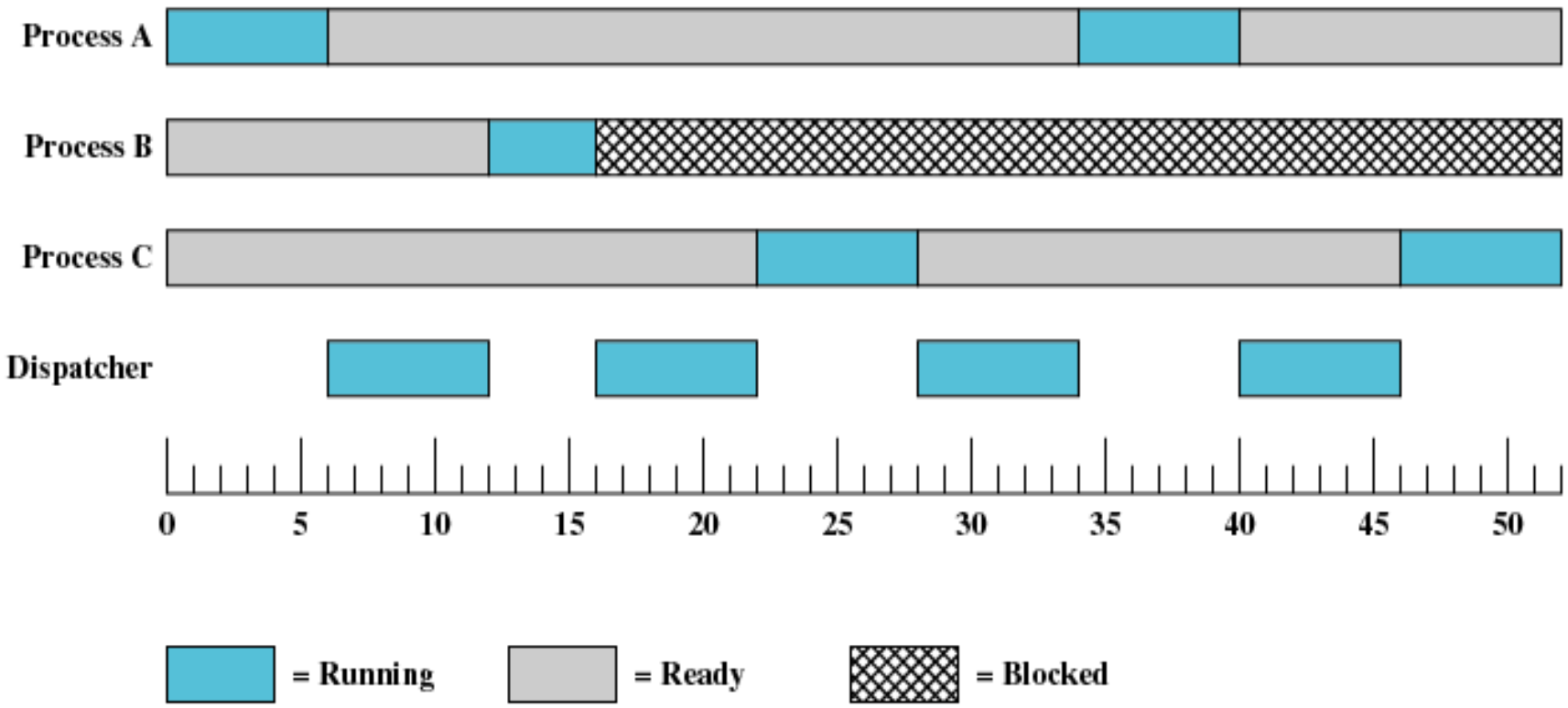
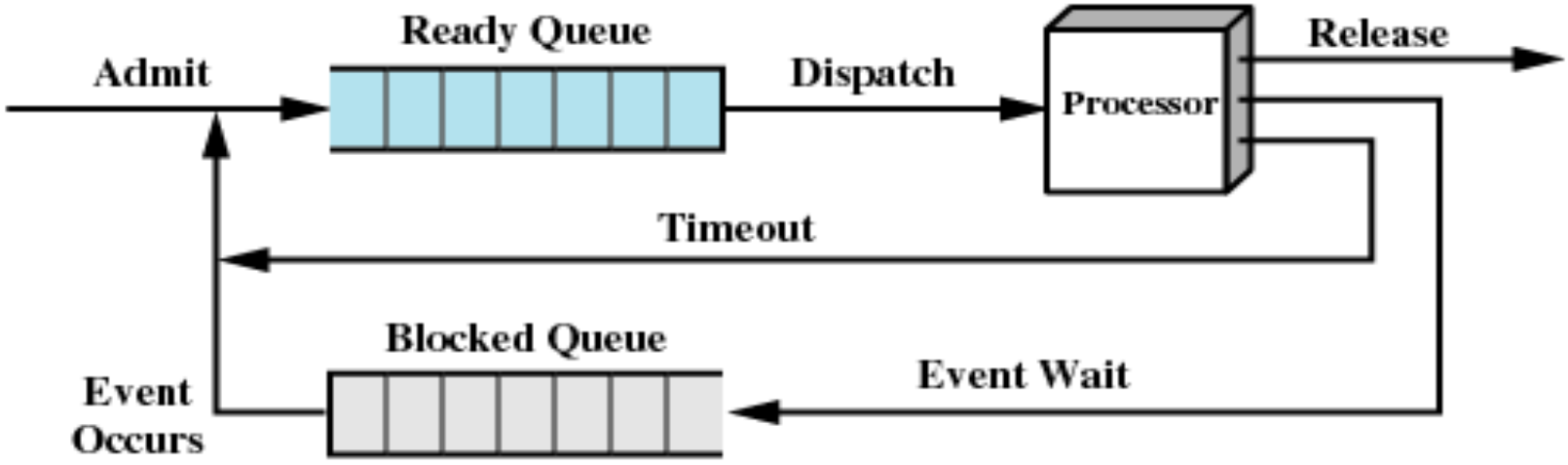


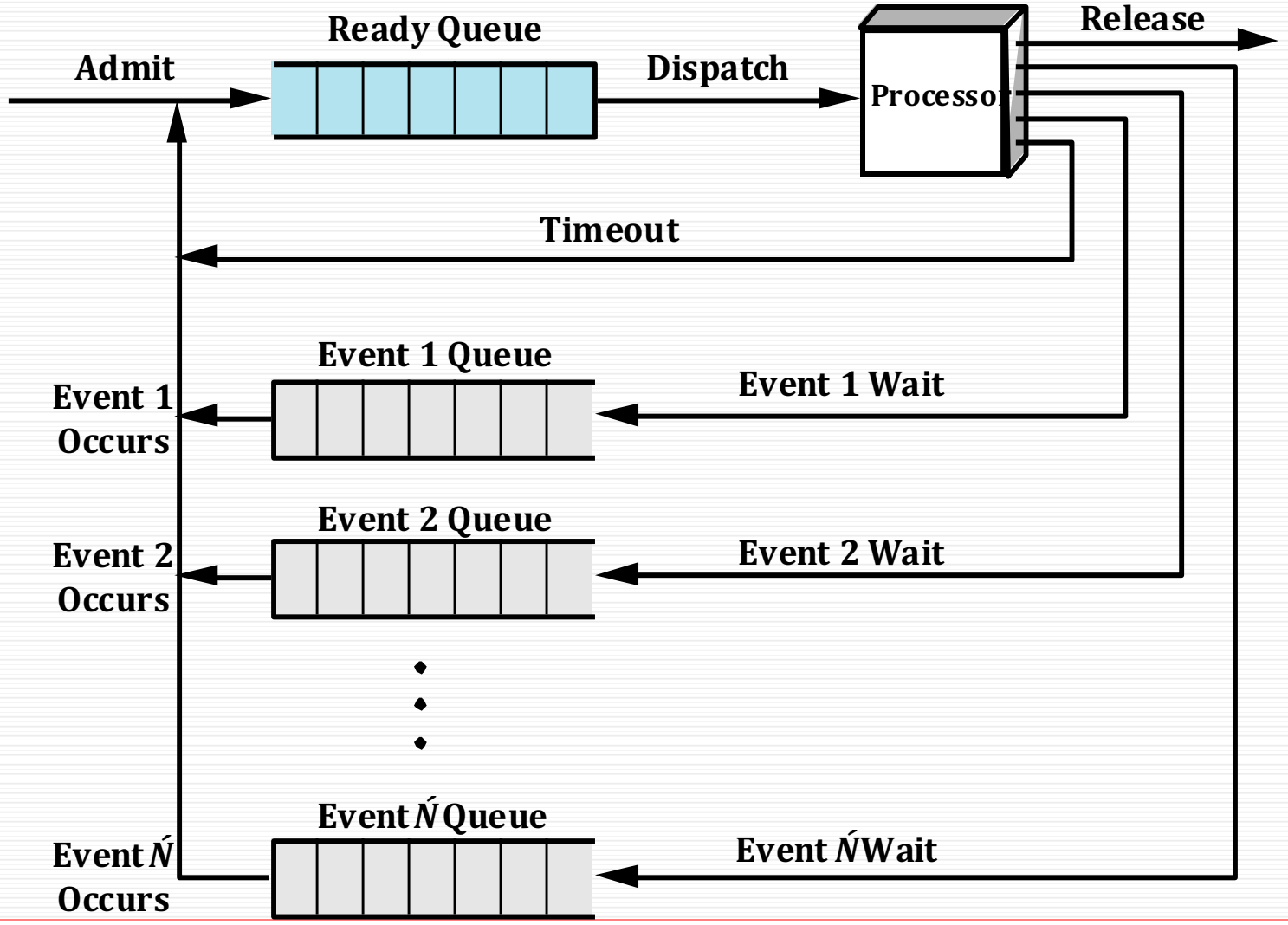
Figure 3.7 Process States for Trace of Figure 3.4

Using Two Queues



(a) Single blocked queue

Multiple Blocked Queues



(b) Multiple blocked queues

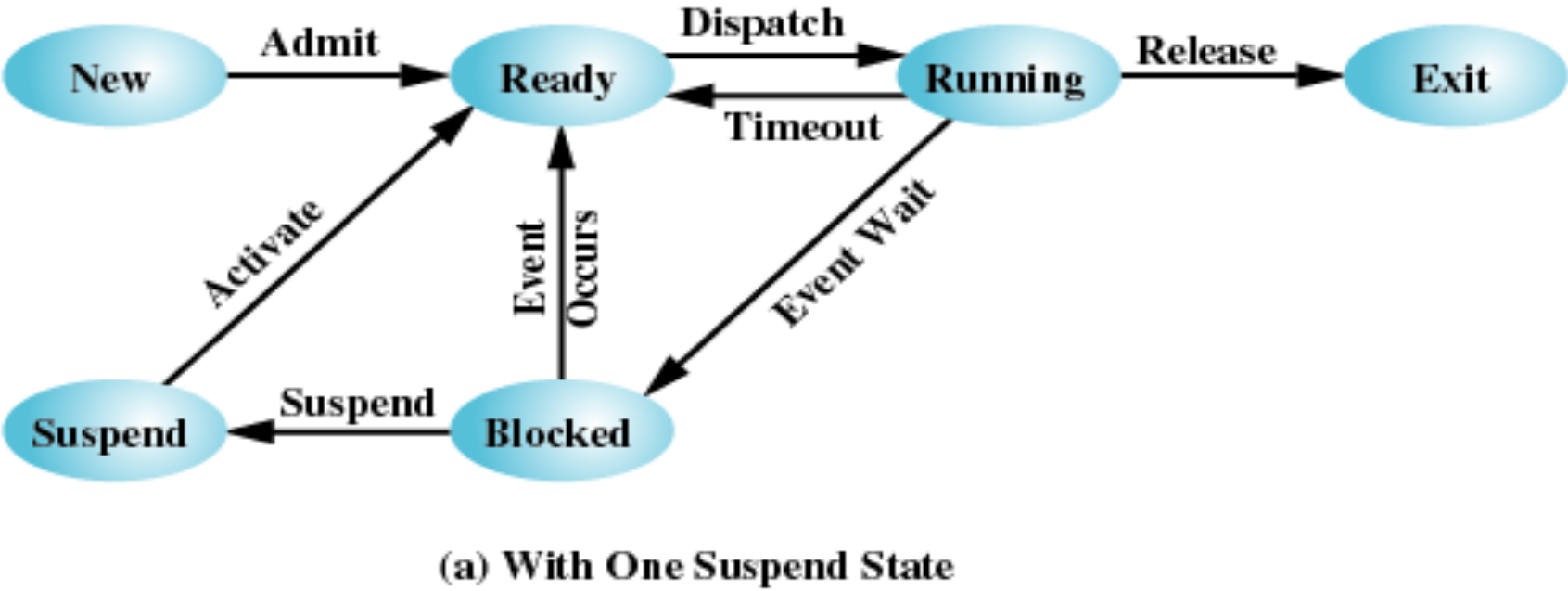
3.2 Process States

- 3.2.1 Trace of the Process
- 3.2.2 A Two-State Process Model
- 3.2.3 The Creation and Termination of Processes
- 3.2.4 A Five-State Model
- 3.2.5 Suspended Process

Suspended Processes(被挂起的进程)

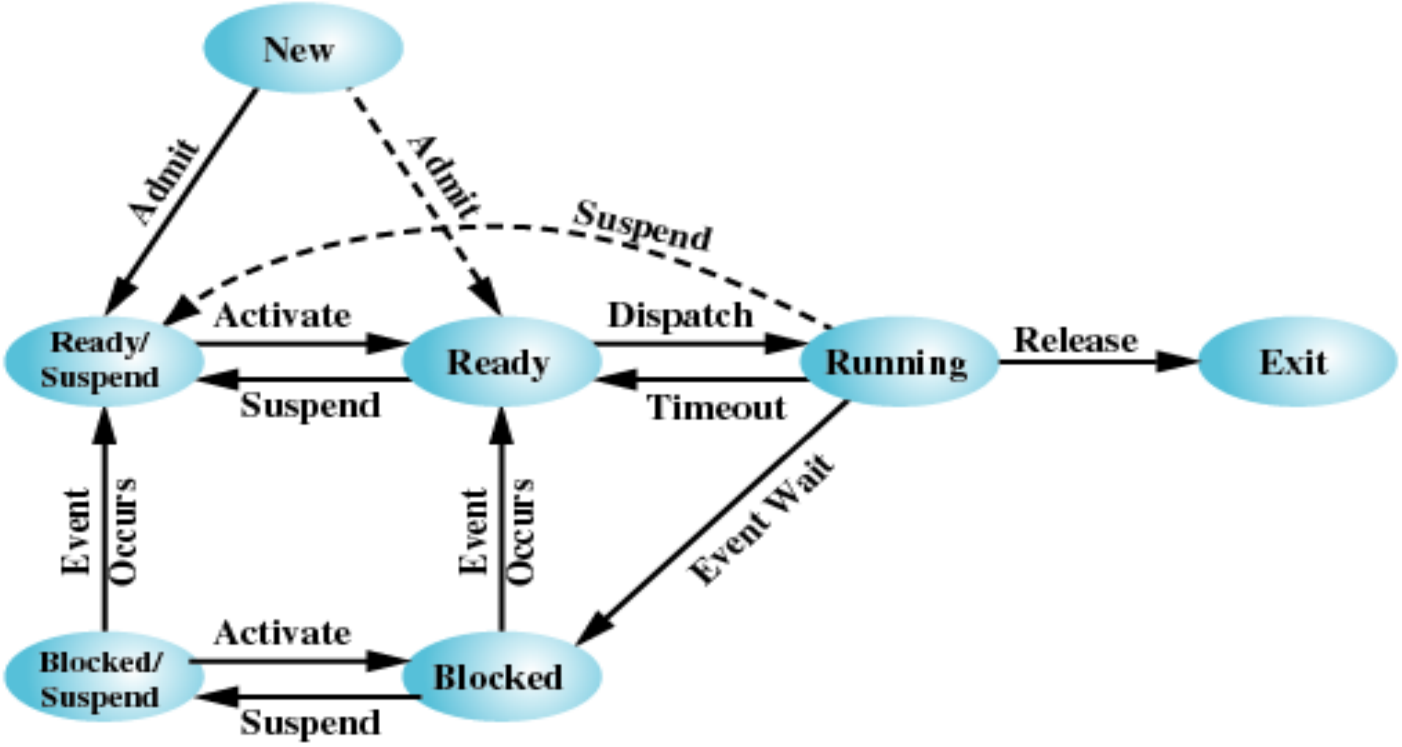
- Processor is faster than I/O so all processes could be waiting for I/O
- Swap these processes to disk to free up more memory
- Blocked state becomes suspend state when swapped to disk
- Two new states
  - Blocked/Suspend
  - Ready/Suspend

One Suspend State



(a) With One Suspend State

Two Suspend States



(b) With Two Suspend States

Figure 3.9 Process State Transition Diagram with Suspend States

Reasons for Process Suspension

Table 3.3 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.

Agenda

- 3.1 What is a Process
- 3.2 Process States
- 3.3 Process Description
- 3.4 Process Control
- 3.5 Summary

3.3 Process Description

- 3.3.1 Processes and Resources
- 3.3.2 Operating System Control Structures
- 3.3.3 Process Control Structures

Processes and Resources

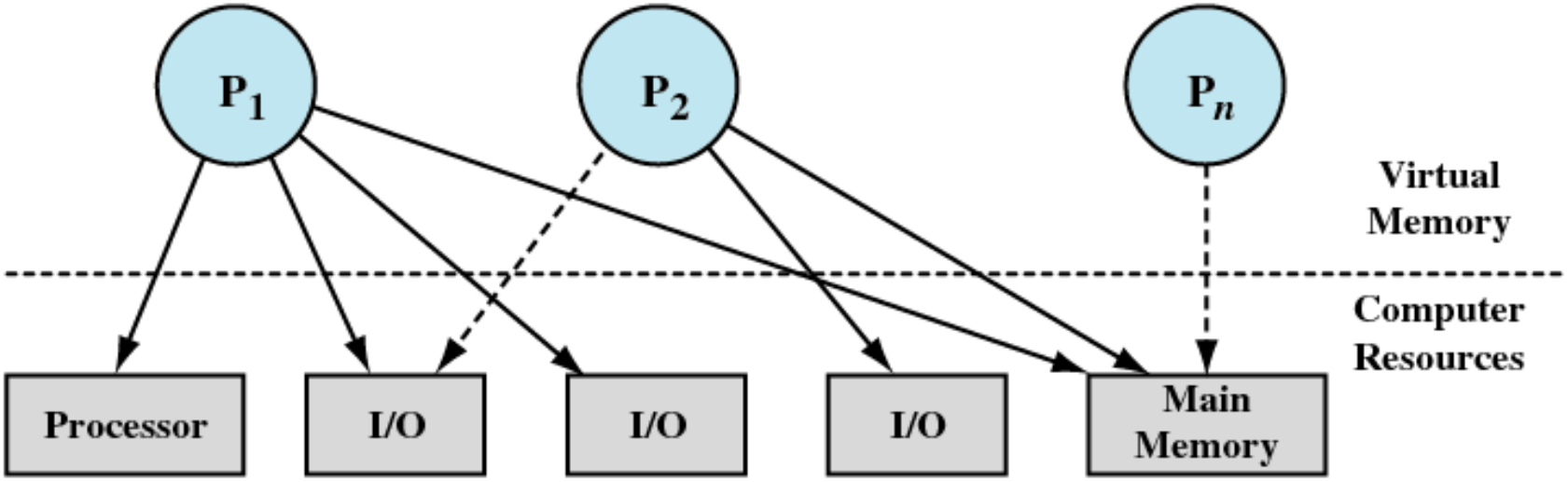


Figure 3.10 Processes and Resources (resource allocation at one snapshot in time)

3.3 Process Description

- 3.3.1 Processes and Resources
- 3.3.2 Operating System Control Structures
- 3.3.3 Process Control Structures

Operating System Control Structures

- Information about the current status of each process and resource (每个进程和资源的当前状态)
- Tables are constructed for each entity the operating system manages (操作系统构造并维护他所管理的所有实体的信息表)

Structure of OS Control Tables

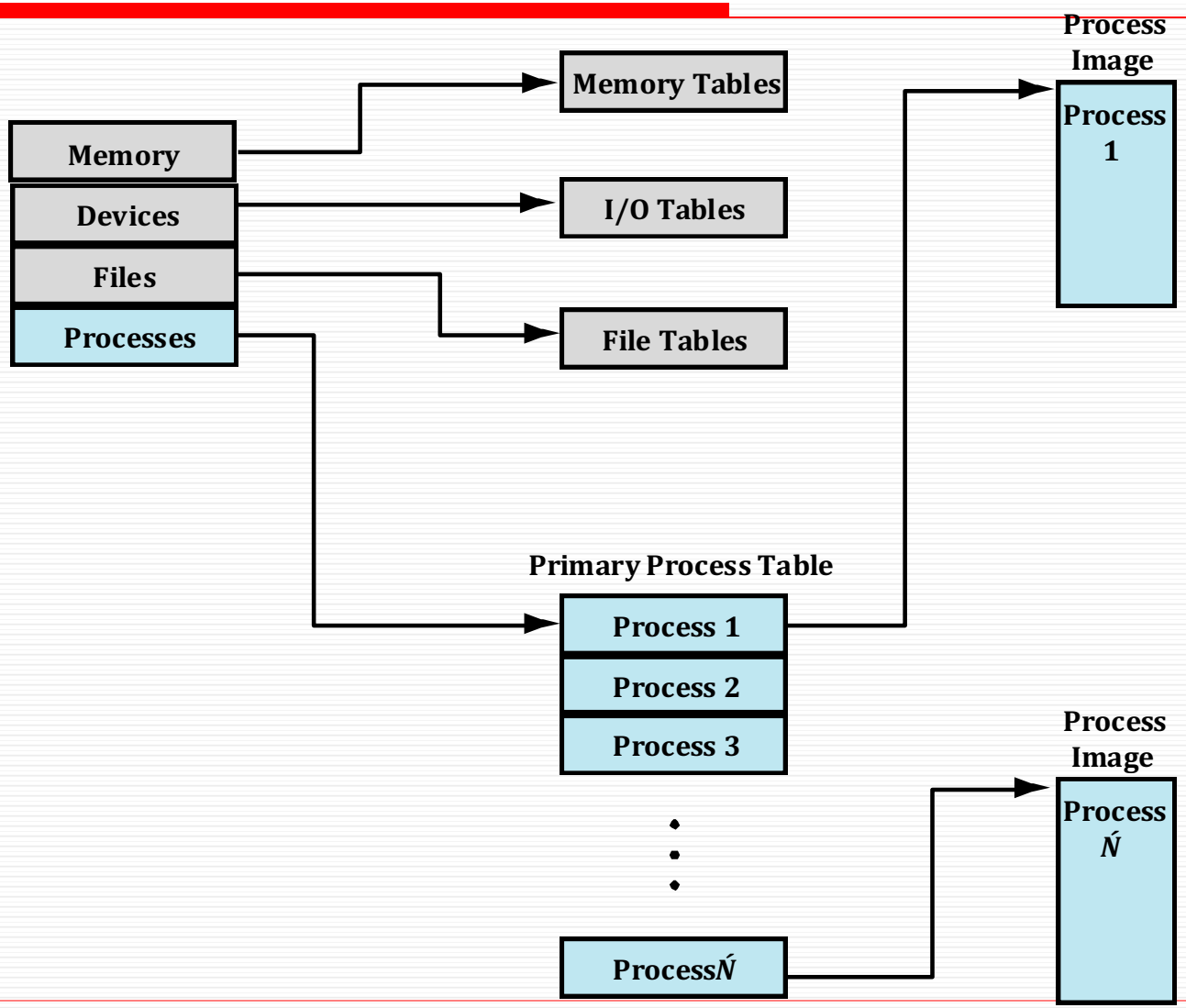


Figure 3.11 General Structure of Operating System Control Tab

Memory Tables

- Allocation of main memory to processes (分配给进程的主存)
- Allocation of secondary memory to processes (分配给进程的辅存)
- Protection attributes for access to shared memory regions(共享内存区域的保护属性)
- Information needed to manage virtual memory(虚拟内存的管理信息)

I/O Tables

- I/O device is available or assigned(分配状态)
- Status of I/O operation
- Location in main memory being used as the source or destination of the I/O transfer (数据传送的源和目的地址)

File Tables

- Existence of files
- Location on secondary memory
- Current Status
- Attributes

Process Table

- Every entry is a description of a process image

Process Image

- Process Image
  - Collection of program, data, stack, attributes

Table 3.4 Typical Elements of a Process Image

User Data
The modifiable part of the user space. May include program data, a user stack area, and programs that may be modified.

User Program
The program to be executed.

System Stack
Each process has one or more last-in-first-out (LIFO) system stacks associated with it. A stack is used to store parameters and calling addresses for procedure and system calls.

Process Control Block
Data needed by the operating system to control the process (see Table 3.5).

3.3 Process Description

- 3.3.1 Processes and Resources
- 3.3.2 Operating System Control Structures
- 3.3.3 Process Control Structures

Process Control Block

- Process Identification
- Processor State Information
  - User-Visible Registers
  - Control and Status Registers
  - Stack Pointers
- Process Control information
  - Scheduling and State Information
  - Data Structuring (link information)
  - Interprocess Communication
  - Process Privileges
  - Memory Management
  - Resource Ownership and Utilization

Agenda

- 3.1 What is a Process
- 3.2 Process States
- 3.3 Process Description
- 3.4 Process Control
- 3.5 Summary

3.4 Process Control

- 3.4.1 Modes of Execution
- 3.4.2 Process Creation
- 3.4.3 Process Switching
- 3.4.4 Execution of the Operating System

Modes of Execution

- User mode
  - Less-privileged mode
  - User programs typically execute in this mode
- System mode, control mode, or kernel mode
  - More-privileged mode
  - Kernel of the operating system
- Translation between the two model
  - Processor status register (psr) and current privileged level (cpl)
  - User mode->System mode
  - System mode->User mode

3.4 Process Control

- 3.4.1 Modes of Execution
- 3.4.2 Process Creation
- 3.4.3 Process Switching
- 3.4.4 Execution of the Operating System

Process Creation

1. Assign a unique process identifier
2. Allocate space for the process
3. Initialize process control block
4. Set up appropriate linkages
   - Ex: add new process to linked list used for scheduling queue
5. Create of expand other data structures
   - Ex: maintain an accounting file

3.4 Process Control

- 3.4.1 Modes of Execution
- 3.4.2 Process Creation
- 3.4.3 Process Switching
- 3.4.4 Execution of the Operating System

When to Switch a Process

A process switch may occur any time that the OS has gained control from the currently running process. The possible events that may give control to the OS include:

1. Interrupt
   - Clock interrupt
     - ✓ process has executed for the maximum allowable time slice
   - I/O interrupt
   - Memory fault
     - ✓ Referenced virtual address is not in main memory, so it must be brought in.

When to Switch a Process

2. Trap
   - error or exception occurred
   - may cause process to be moved to Exit state
3. Supervisor call (System Call)
   - such as file open

Process Switching

1. Save context of processor including program counter and other registers
2. Update the process control block of the process and change the process's state that is currently in the Running state
3. Move process control block to appropriate queue – ready; blocked; ready/suspend
4. Select another process for execution

Process Switching

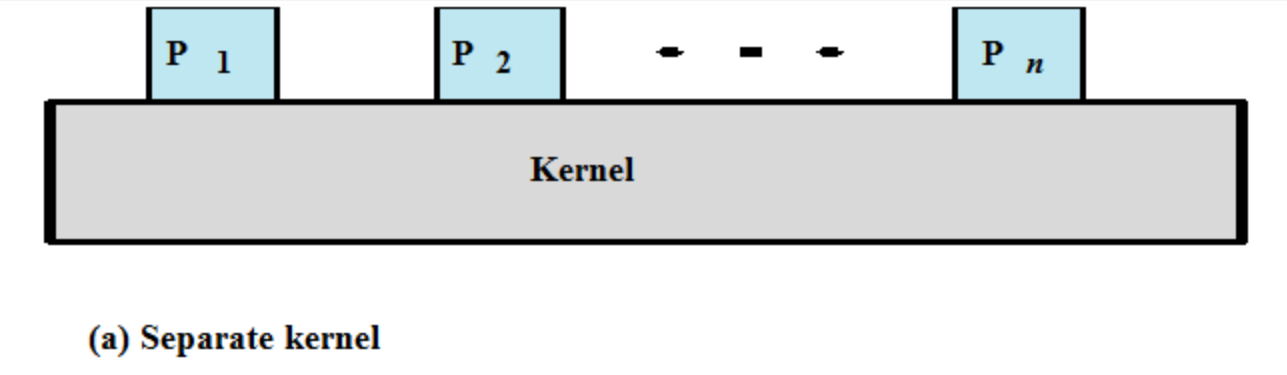
5. Update the process control block of the process selected and change it state
6. Update memory-management data structures
7. Restore context of the selected process

3.4 Process Control

- 3.4.1 Modes of Execution
- 3.4.2 Process Creation
- 3.4.3 Process Switching
- 3.4.4 Execution of the Operating System

Execution of the Operating System

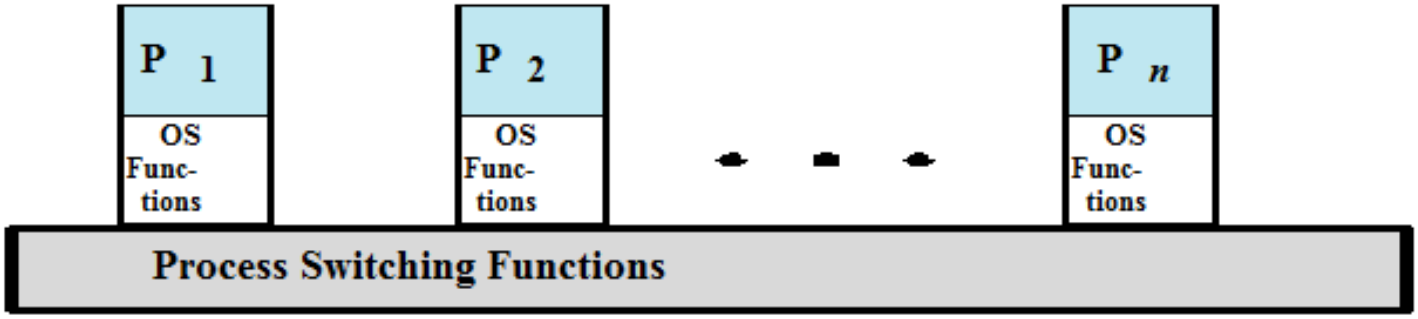
1. Non-process Kernel（无进程内核）
   - Execute kernel outside of any process
   - Operating system code is executed as a separate entity that operates in privileged mode



(a) Separate kernel

Execution of the Operating System

2. Execution Within User Processes(在用户进程中执行)
   - Operating system software within context of a user process
   - Process executes in privileged mode when executing operating system code



(b) OS functions execute within user processes

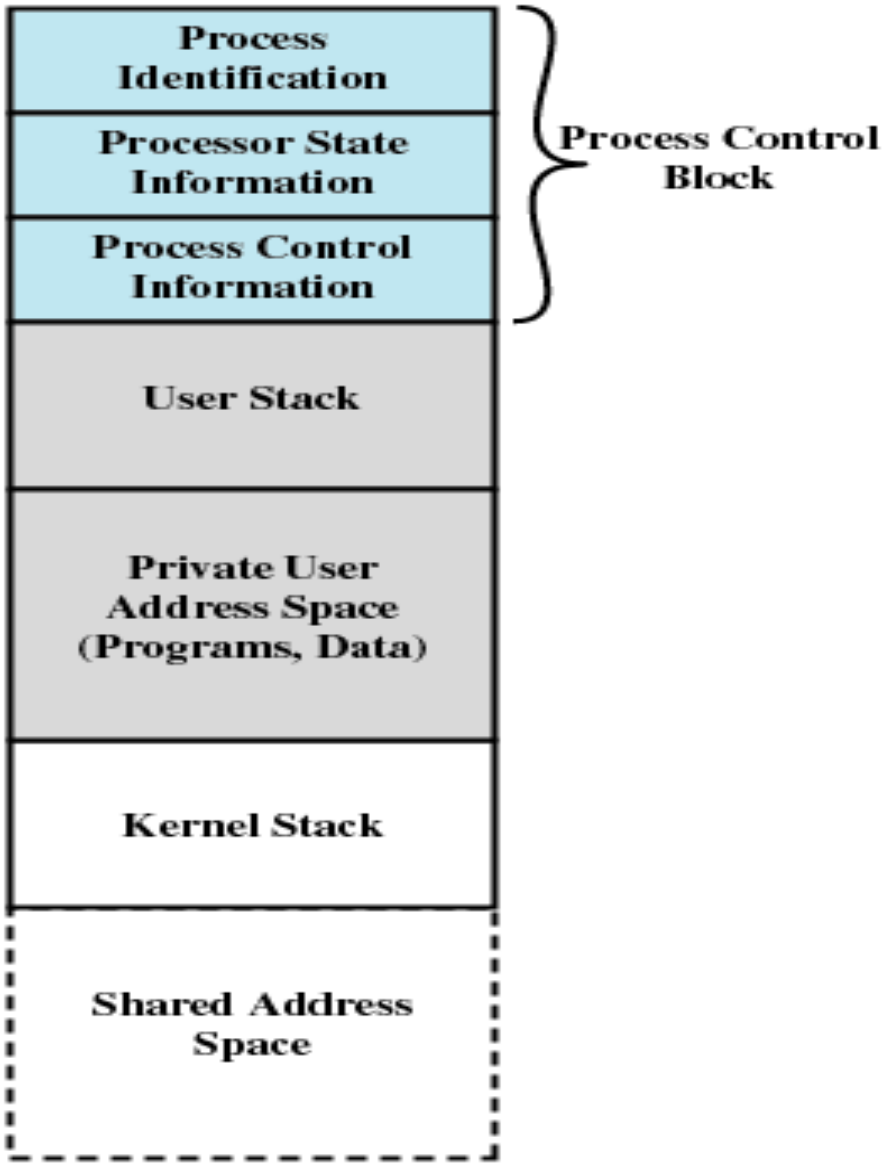
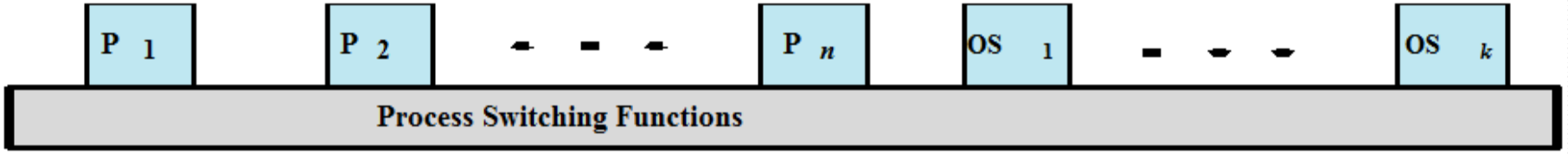


Figure 3.16 Process Image: Operating System Executes Within User Space

Execution of the Operating System

3. Process-Based Operating System（基于进程的OS）
   - Implement operating system as a collection of system processes
   - Useful in multi-processor or multi-computer environment



(c) OS functions execute as separate processes

Agenda

- 3.1 What is a Process
- 3.2 Process States
- 3.3 Process Description
- 3.4 Process Control
- 3.5 Summary