

Processes and Scheduling

Part 2

进程与调度

Abstract

基础：进程描述及控制

实现：互斥与同步

避免：死锁与饥饿

解决：几个经典问题

关于：进程通信

策略：进程调度

§ 2. 1 Process Description And Control

Learning objectives

By the end of this lecture you should be able to:

- **Explain what's Process, Swapping, Thread**
- **掌握分析进程的结构, PCB, Process image(进程映像)**
- **描述进程的基本状态及转换规则与原因**
- **区别进程的挂起与阻塞状态**
- **理解OS内核的主要功能**
- **理解Process Control Primitives（原语）**
- **区别Process Switching vs. Mode Switching**
- **区别Process vs. Thread**

Major Requirements of an Operating System

- **Interleave the execution of several processes to maximize processor utilization while providing reasonable response time**
- **Allocate resources to processes**
- **Support interprocess communication and user creation of processes**

程序的执行顺序

- 程序顺序执行
 - 程序顺序执行时的特征：顺序性、封闭性、可再现性
- 程序并发执行
 - 程序并发执行时的特征：间断性、非封闭性、不可再现性
- 程序并发执行条件（**Bernstein条件**）

$$R(P1) \cap W(P2) \cup W(P1) \cap R(P2) \cup W(P1) \cap W(P2) = \{ \}$$

Process

- **Also called a task**
- **Execution of an individual program**
 - ✓ 进程是程序在一个数据集合上的运行过程，是系统进行资源分配和调度的一个独立单位
 - ✓ 进程是可并发执行的程序在一个数据集合上的运行过程
- **Can be traced**
 - **list the sequence of instructions that execute**

Characteristics of Process

- **Dynamic(动态性)**
- **Concurrency(并发性)**
- **Independent(独立性)**
- **Asynchronous(异步性)**

Process Structure

- **Programs**
- **Datas**
- **PCB (Process Control Block)**

Process States

进程的并发执行

进程的2状态

进程的5状态

进程状态转换图

例

假设内存中有3个进程A、B、C，他们的程序代码已全部装入内存。若A、C两进程需要执行12条指令，B进程需要执行4条指令，且B进程执行到第4条指令处必须等待I/O。如何跟踪他们的执行过程？

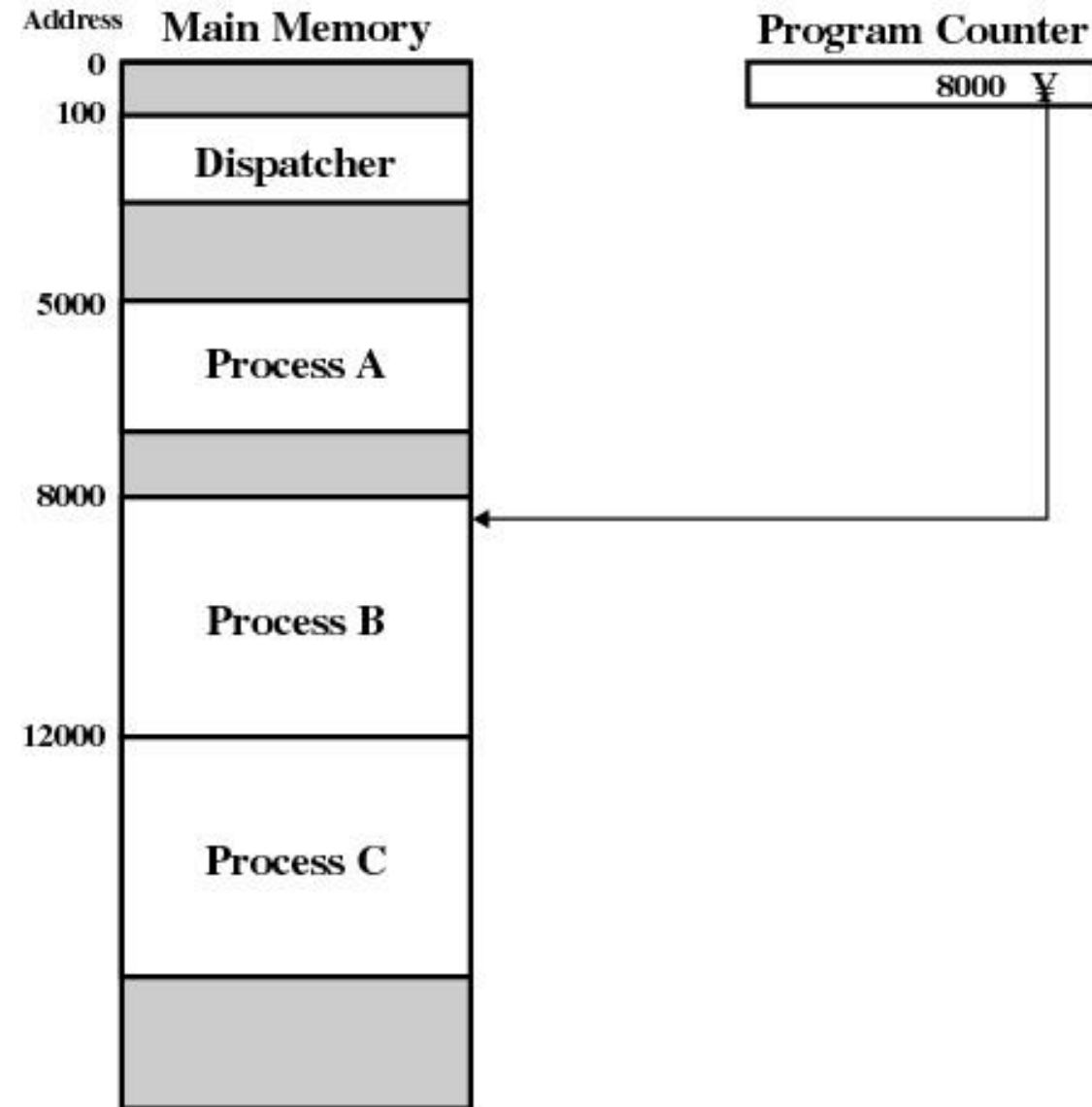


Figure 3.1 Snapshot of Example Execution (Figure 3.3)
at Instruction Cycle 13

1	5000			27	12004
2	5001			28	12005
3	5002				-----Time out
4	5003			29	100
5	5004			30	101
6	5005			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;

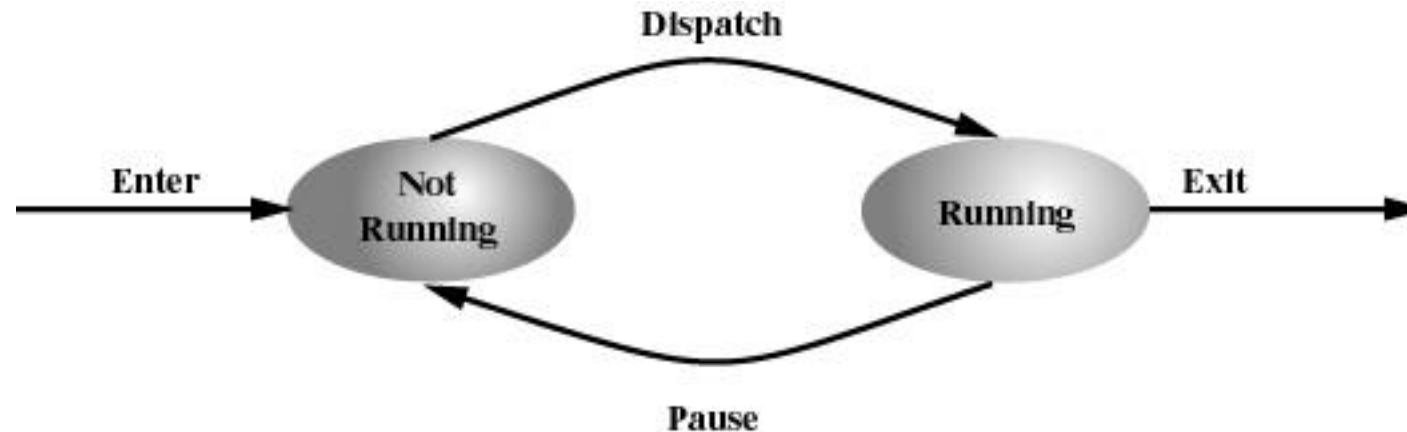
first and third columns count instruction cycles;

second and fourth columns show address of instruction being executed

Figure 3.3 Combined Trace of Processes of Figure 3.1

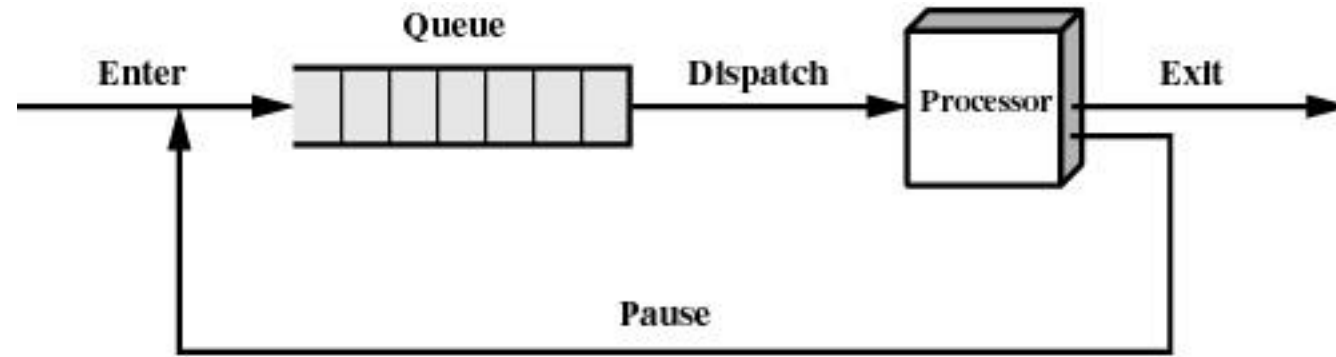
Two-State Process Model

- Process may be in one of two states
 - Running(执行)
 - Not-running (非执行)



(a) State transition diagram

Not-Running Process in a Queue



(b) Queuing diagram

注:

- 并非所有进程只要Not-running就处于ready(就绪)，有的需要blocked(阻塞)等待I/O完成
- Not-running又可分为ready和blocked两种状态

A Five-State Model

- Running (执行)
- Ready (就绪)
- Blocked (阻塞)
- New (新状态)
- Exit (退出)

- **Running:** 占用处理机（单处理机环境中，某一时刻仅一个进程占用处理机）
- **Ready:** 准备执行
- **Blocked:** 等待某事件发生才能执行，如等待I/O完成等
- **New:** 进程已经创建，但未被OS接纳为可执行进程
- **Exit:** 因停止或取消，被OS从执行状态释放

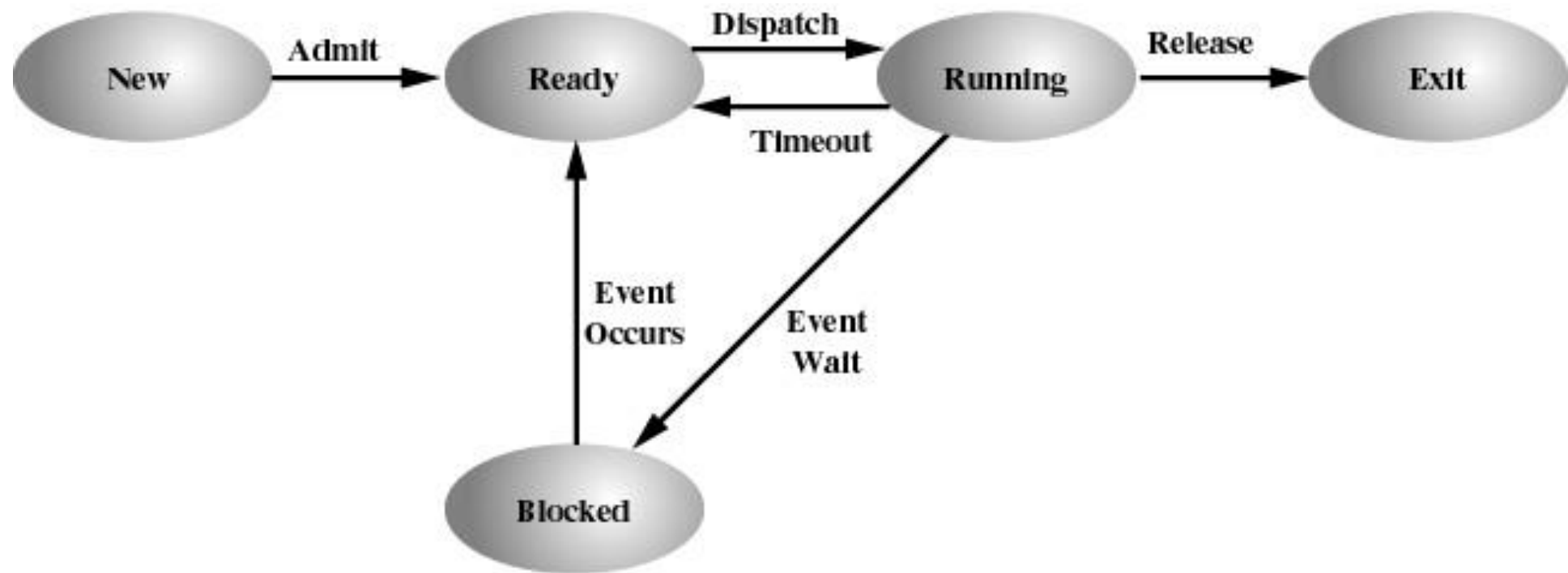


Figure 3.5 Five-State Process Model

- **Null→New**: 新创建进程首先处于新状态
- **New→Ready**: OS接纳新状态进程为就绪进程
- **Ready→Running**: OS只能从就绪进程中选一个进程执行
- **Running→Exit**: 执行状态的进程执行完毕, 或被取消, 则转换为退出状态
- **Running→Ready**: 分时系统中, 时间片用完, 或优先级高的进程到来, 将终止优先级低的进程的执行
- **Running→Blocked**: 执行进程需要等待某事件发生。通常因进程需要的系统调用不能立即完成, 而阻塞
- **Blocked→Ready**: 当阻塞进程等待的事件发生, 就转换为就绪状态
- **Ready→Exit**: 某些系统允许父进程在任何情况下终止其子进程。若一个父进程终止, 其子孙进程都必须终止。
- **Blocked→Exit**: 同前

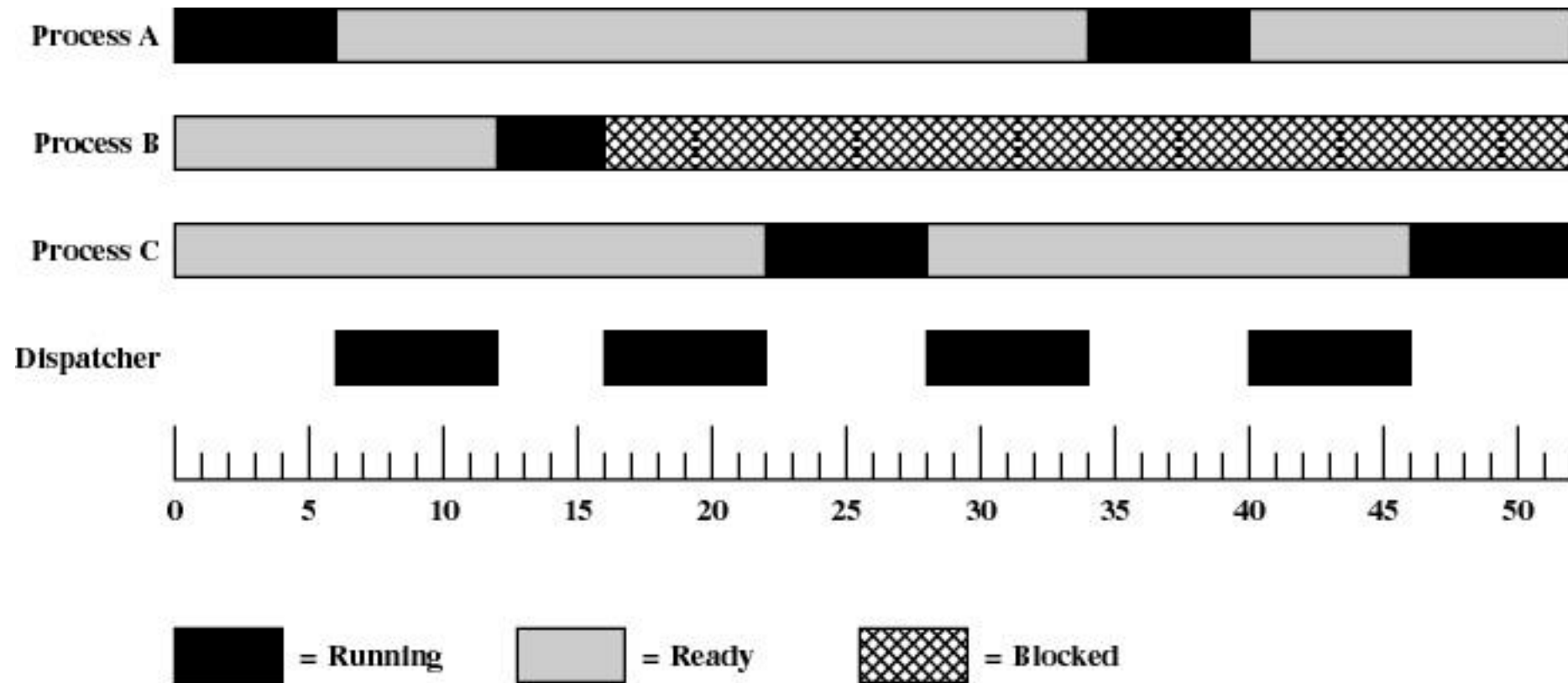
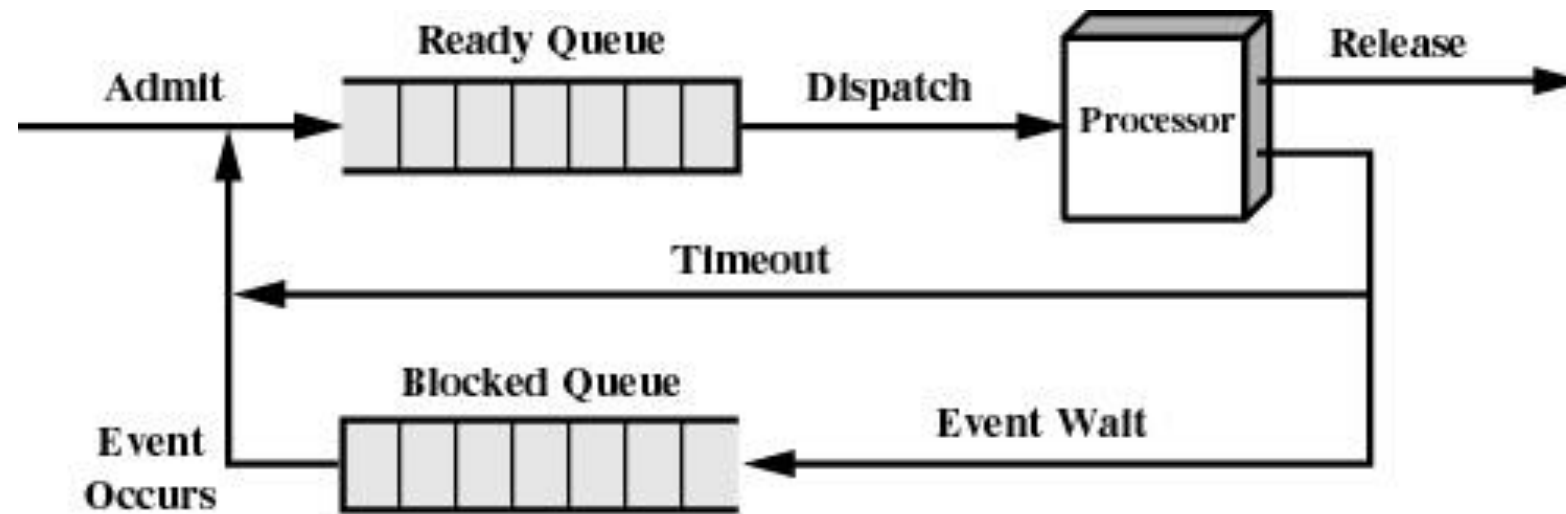
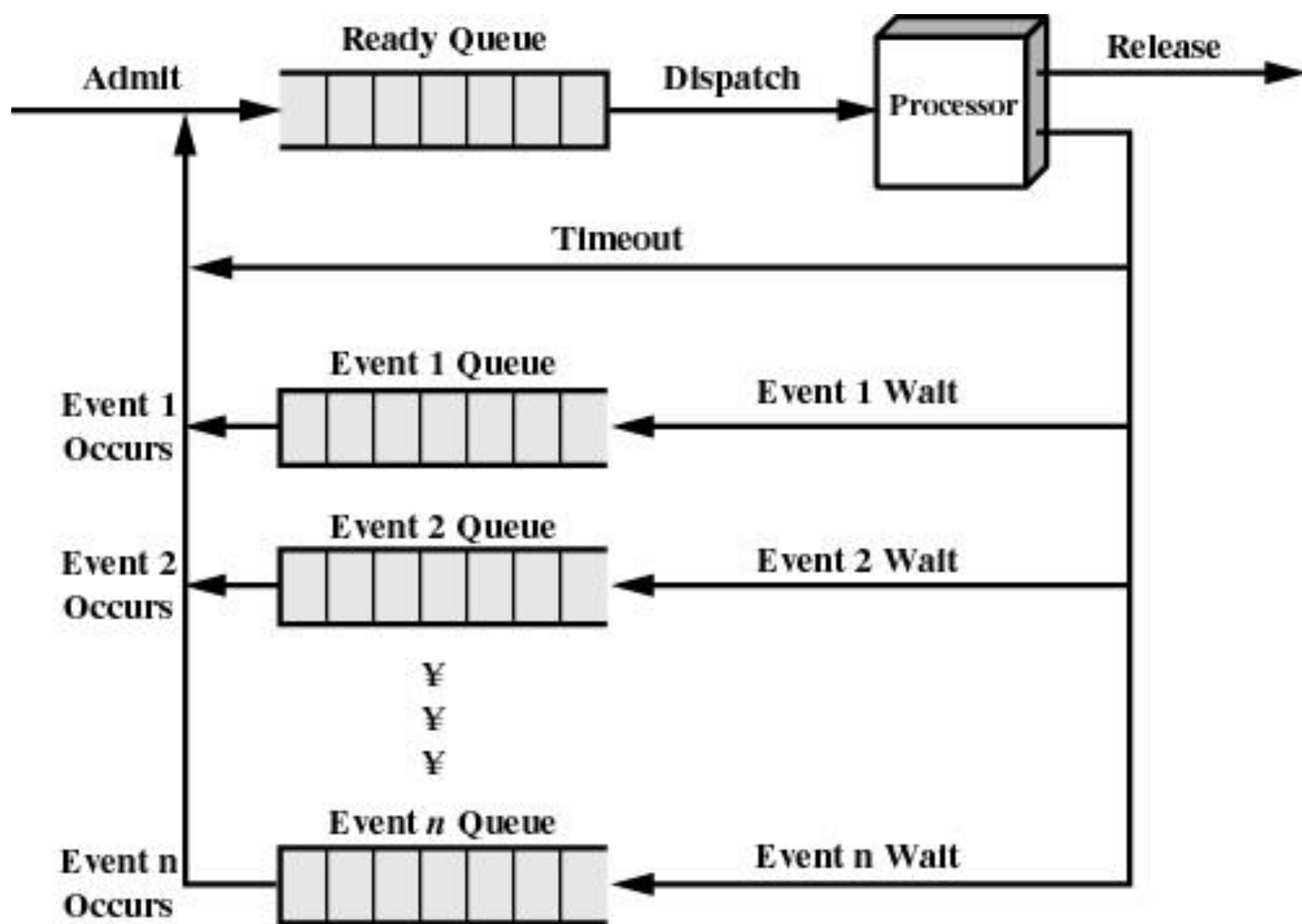


Figure 3.6 Process States for Trace of Figure 3.3

Using Two Queues



(a) Single blocked queue



(b) Multiple blocked queues