第8-9讲 进程状态转换

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？

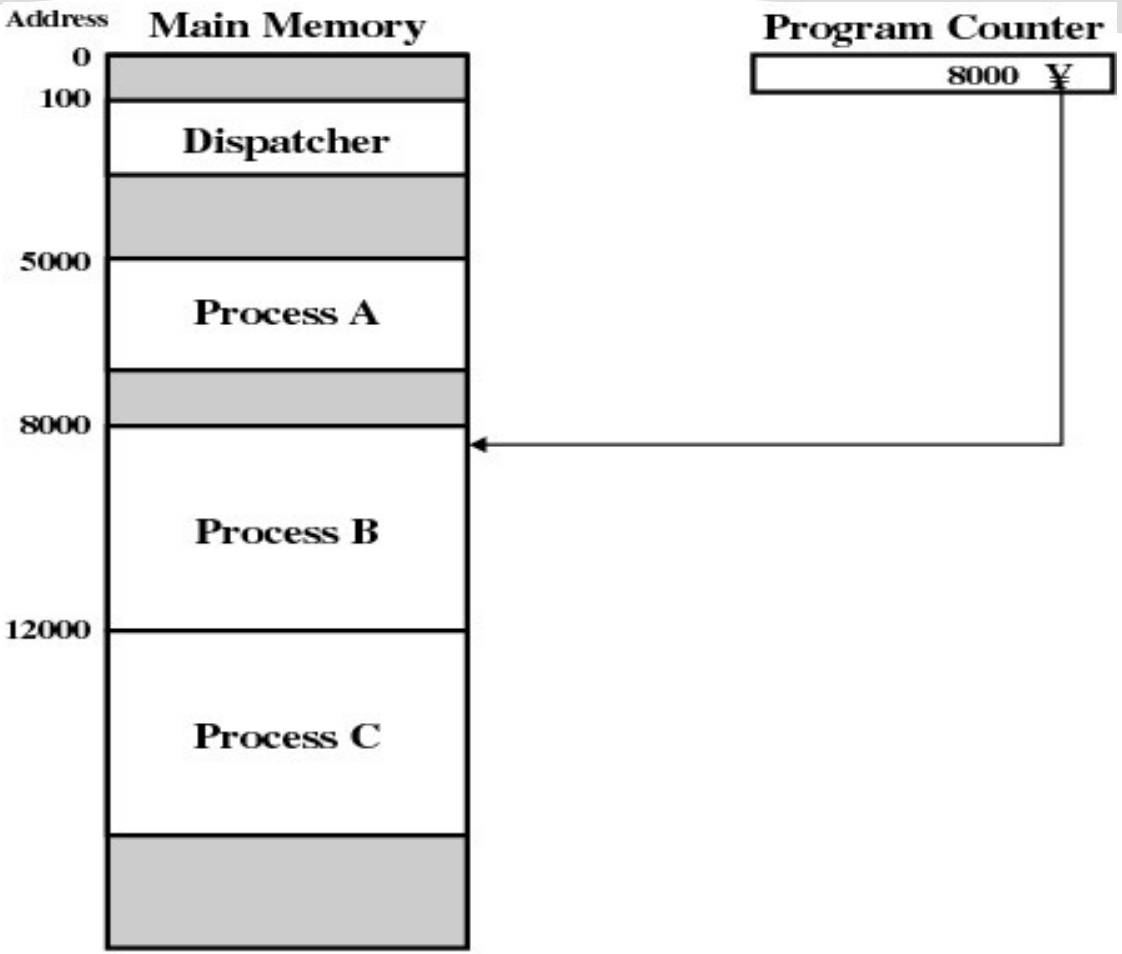


Figure 3.1 Snapshot of Example Execution (Figure 3.3) at Instruction Cycle 13

1	5000
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

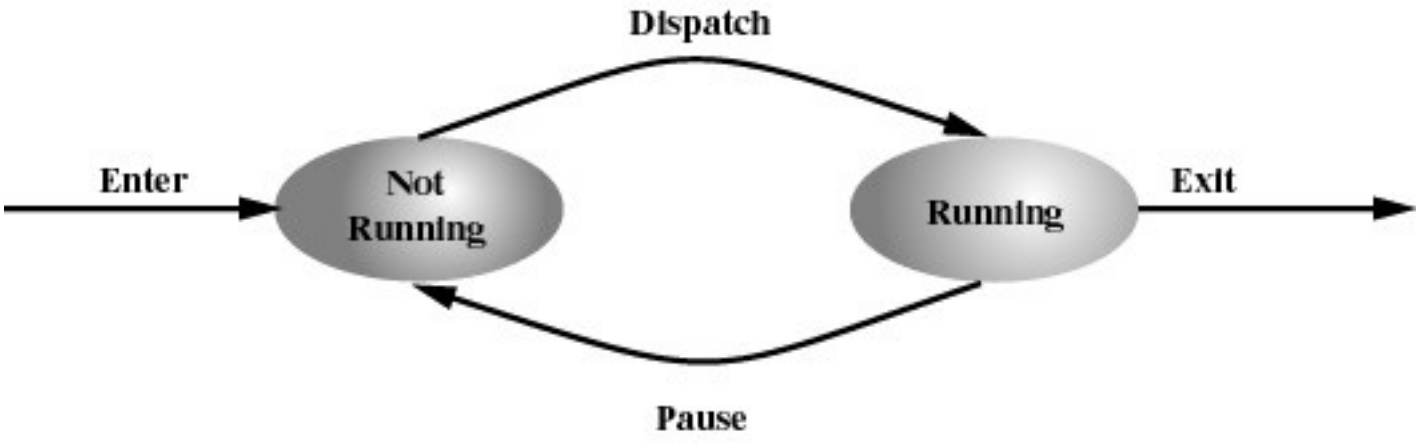
100 = Starting address of dispatcher program

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 3.3 Combined Trace of Processes of Figure 3.1

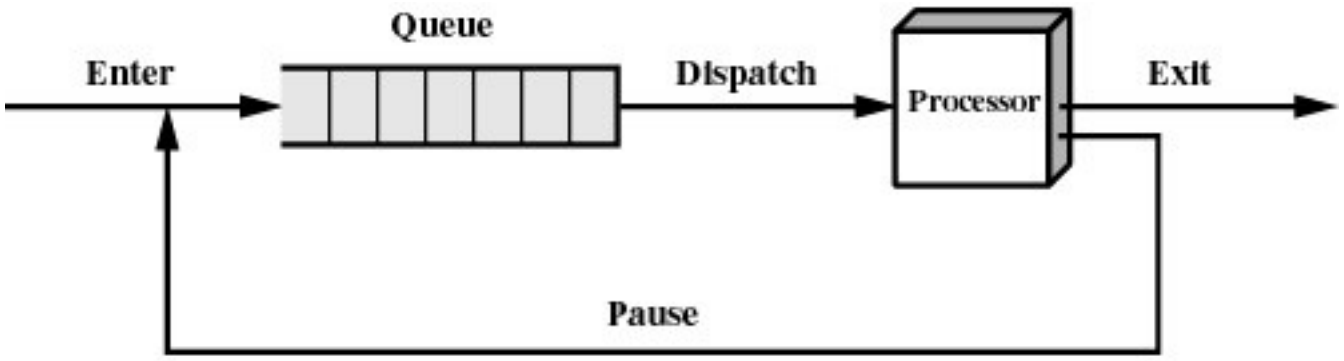
Two-State Process Model

- **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

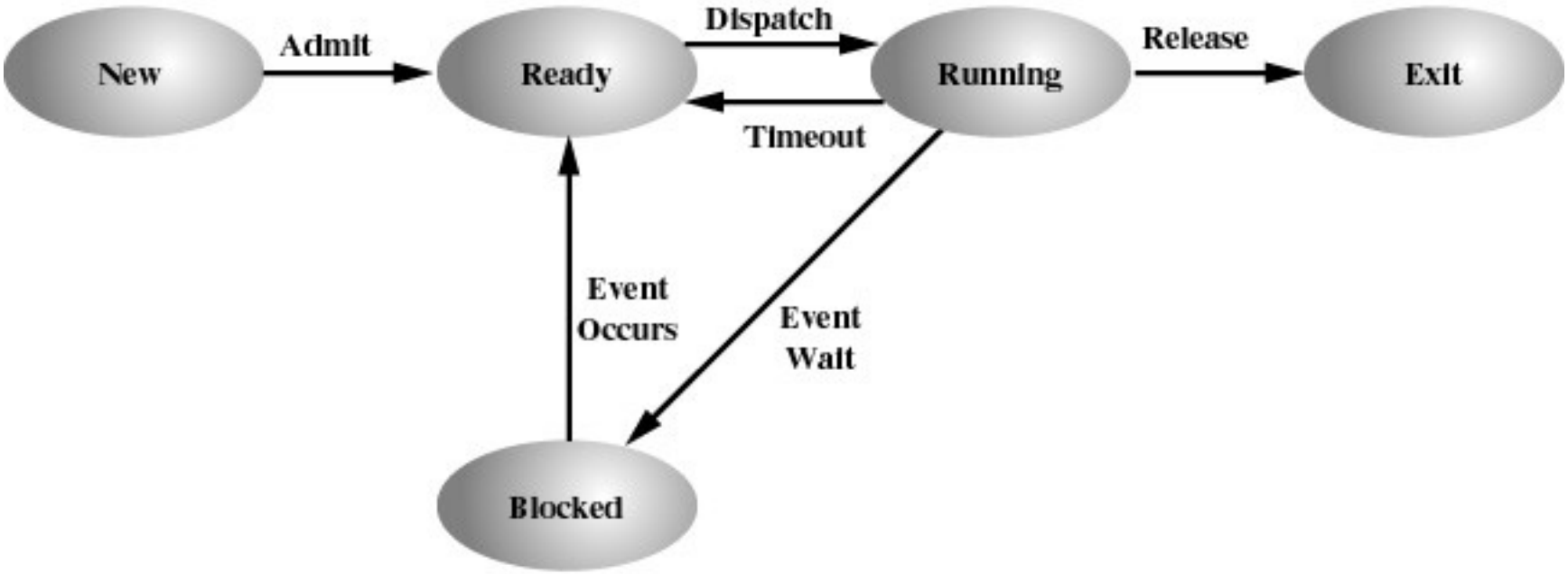


Figure 3.5 Five-State Process Model

- 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

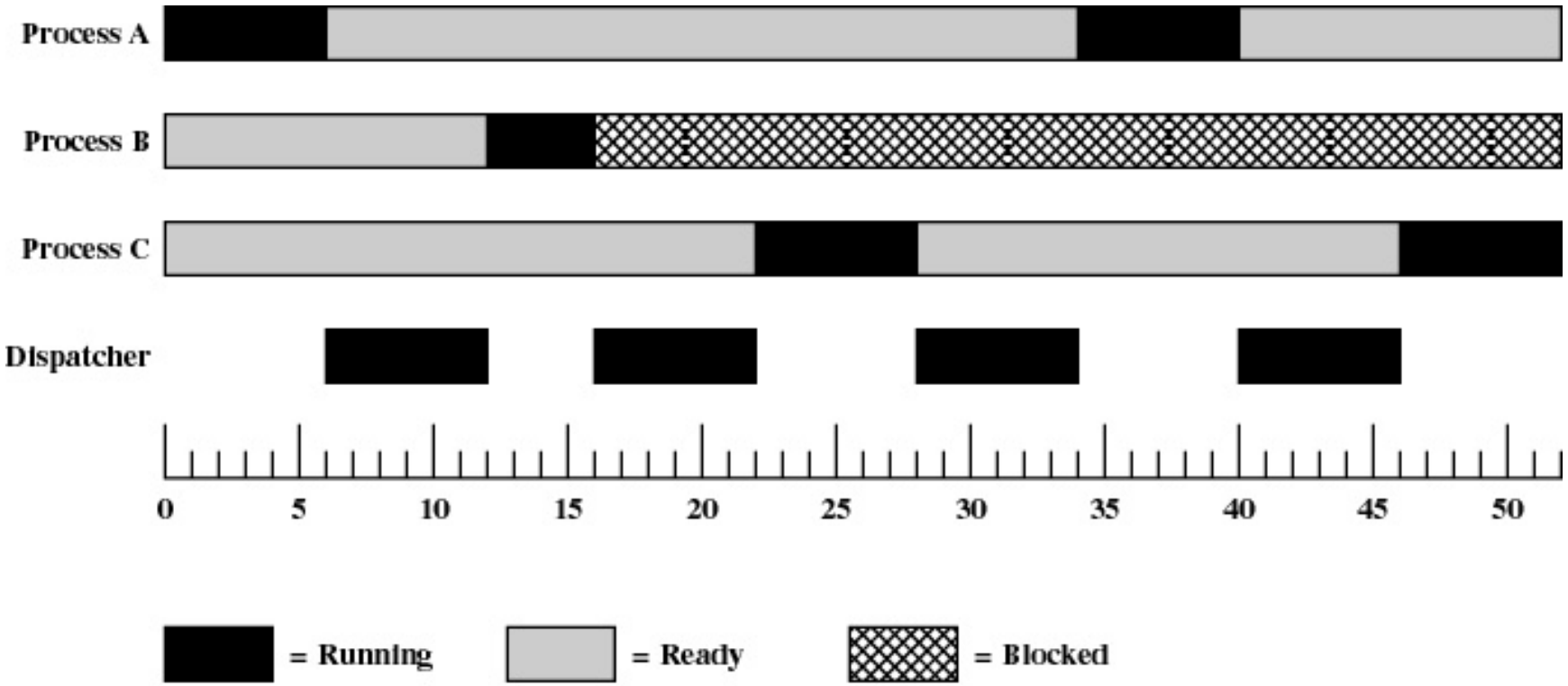
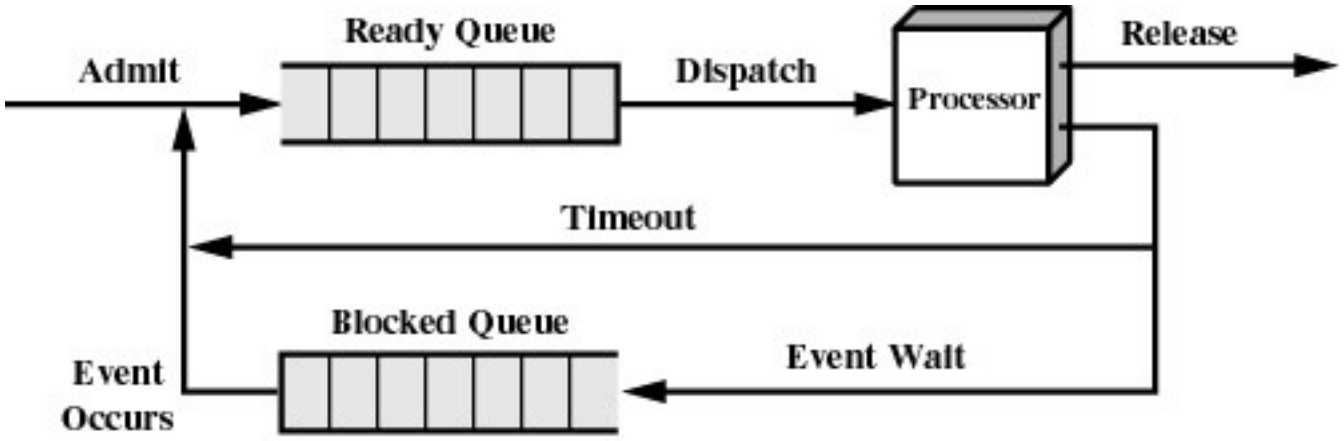
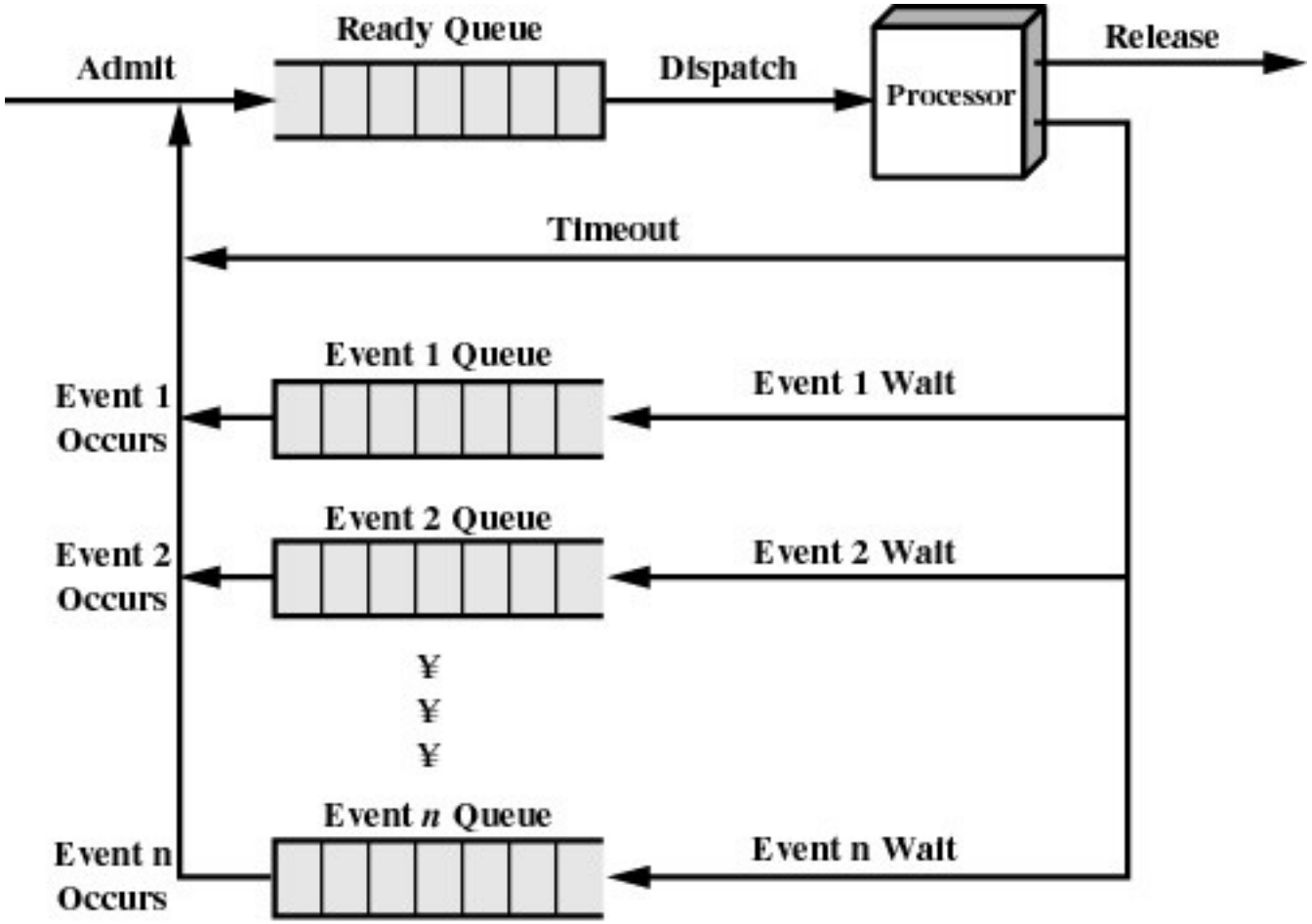


Figure 3.6 Process States for Trace of Figure 3.3

Using Two Queues



(a) Single blocked queue



(b) Multiple blocked queues

Swapping（对换技术，交换技术）

将内存中暂时不能运行的进程，或暂时不用的数据和程序， Swapping-out 到外存，以腾出足够的内存空间，把已具备运行条件的进程，或进程所需要的数据和程序， Swapping-in 内存。

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

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.

被挂起进程的特征

- 不能立即执行
- 可能是等待某事件发生。若是，则阻塞条件独立于挂起条件，即使阻塞事件发生，该进程也不能执行
- 使之挂起的进程为：自身、其父进程、 OS
- 只有挂起它的进程才能使之由挂起状态转换为其他状态

Suspend *vs.* Blocked (挂起与阻塞)

问题

1. 是否只能挂起阻塞进程?

2. 如何激活一个挂起进程?

Suspend *vs.* Blocked

- 区分两个概念：
  - ？ 进程是否等待事件，阻塞与否
  - ？ 进程是否被换出内存，挂起与否
- ４种状态组合：

Ready：进程在内存，准备执行

Blocked：进程在内存，等待事件

Ready，Suspend：进程在外存，只要调入内存即可执行

Blocked,Suspend：进程在外存，等待事件

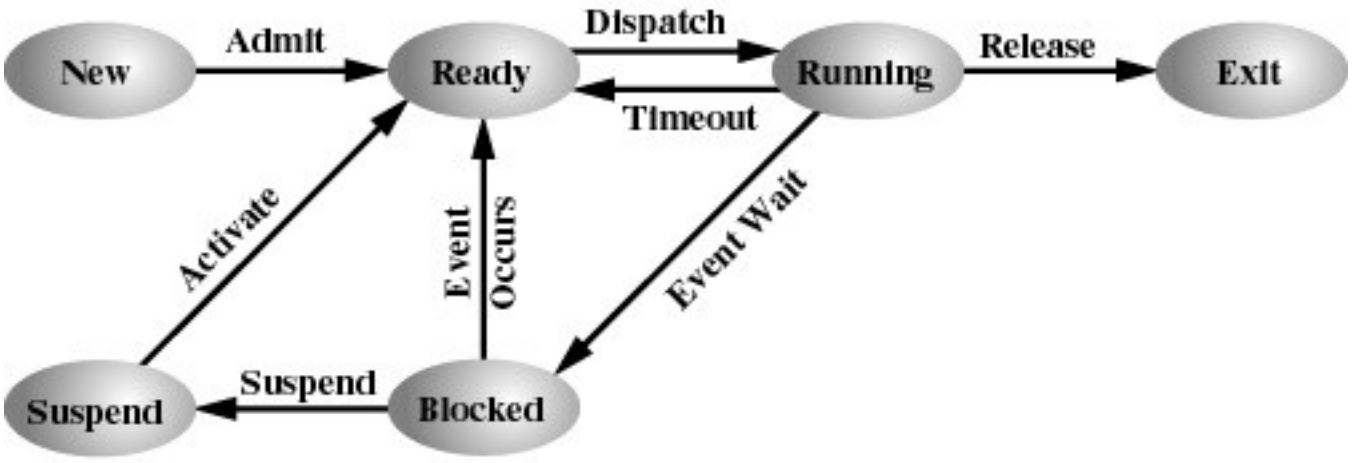
注：

处理机可调度执行的进程有两种：

- 新创建的进程
- 或换入一个以前挂起的进程

通常为避免增加系统负载，系统会换入一个以前挂起的进程执行。

One Suspend State



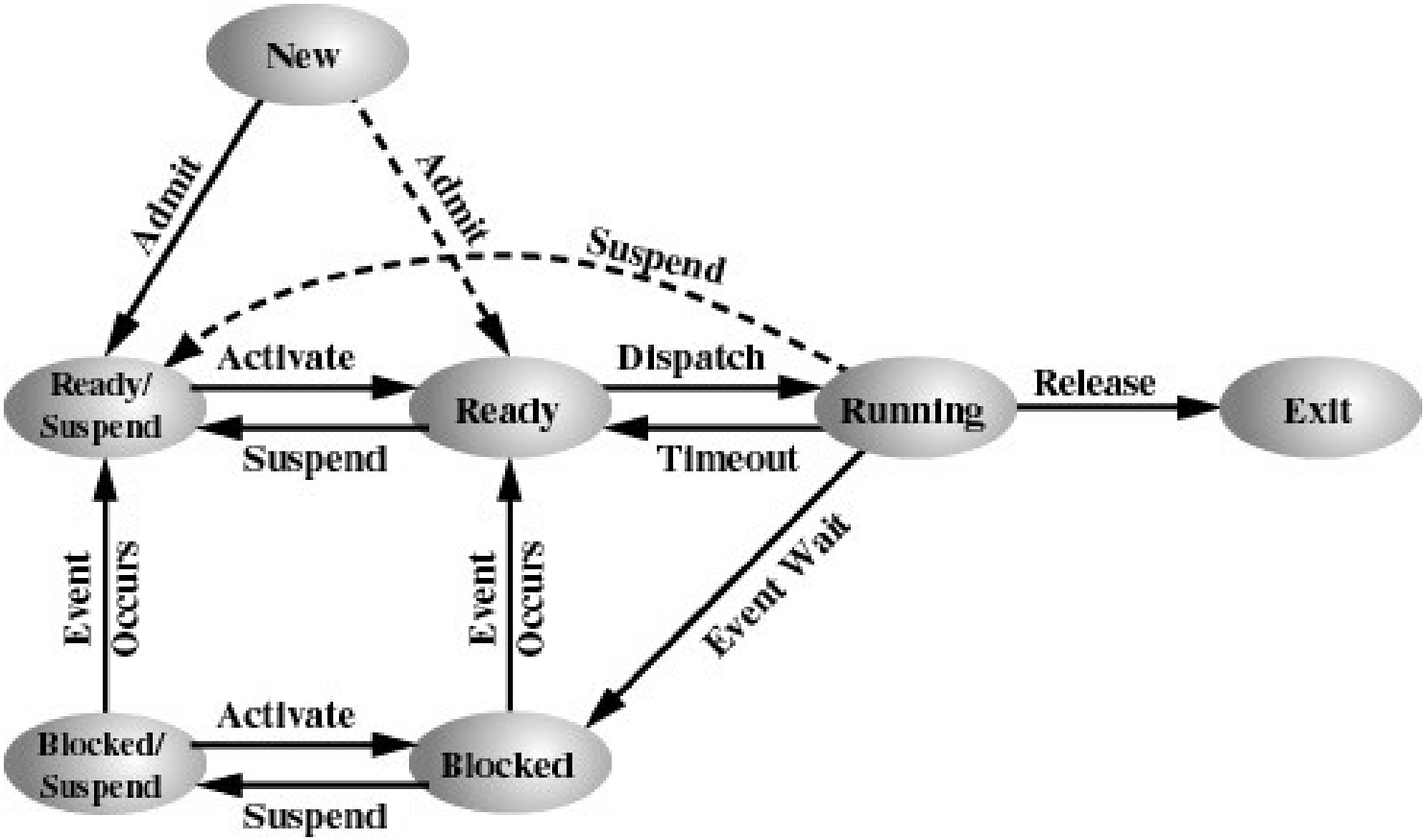
(a) With One Suspend State

此方案存在问题！

Two new states

- Blocked, suspend
- Ready, suspend

Two Suspend States



(b) With Two Suspend States

具有挂起状态的进程状态转换

- Blocked → Blocked, Suspend ： OS 通常将阻塞进程换出，以腾出内存空间
- Blocked, Suspend → Ready, Suspend: 当 Blocked, Suspend 进程等待的事件发生时，可以将其转换为 Ready, Suspend
- Ready, Suspend → Ready ： OS 需要调入一个进程执行时 →
- Ready → Ready, Suspend ：一般， OS 挂起阻塞进程。但有时也会挂起就绪进程，释放足够的内存空间
- New → Ready, Suspend （ New → Ready ）：新进程创建后，可以插入到 Ready 队列或 Ready, Suspend 队列。若无足够的内存分配给新进程，则需要 New

具有挂起状态的进程状态转换（续）

- Blocked, Suspend → Blocked ：当 Blocked,Suspend 队列中有一个进程的阻塞事件可能会很快发生，则可将一个 Blocked, Suspend 进程换入内存，变为 Blocked
- Running → Ready, Suspend ：当执行进程的时间片用完时，会转换为 Ready 。或，一个高优先级的 Blocked, Suspend 进程正好变为非阻塞状态， OS 可以将执行进程转换为 Ready,Suspend 状态
- All → Exit ：通常， Running Exit 。但某些 OS 中，父进程可以终止其子进程，使任何状态的进程都可转换为退出状态