

第二章 进程的描述与控制

1010010101001111010000100101110100101101010101110100004100001010010100
0041000010100101001001010000101101001010140000111101001010100111101000010010111010010
110101010101110100004100001010010100100101000010110100101014000011110100101

第二章 进程的描述与控制

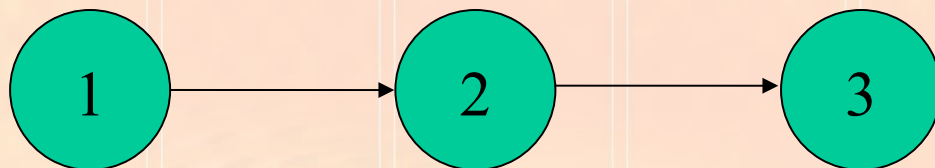
- 为了描述程序在并发执行时对系统资源的共享，我们需要一个描述程序执行时动态特征的概念，这就是进程或线程。

- 进程的基本概念
- 进程控制
- 进程同步
- 经典进程的同步问题
- 进程通信
- 线程

2.1 程序的执行特征

- 程序的执行有两种方式：**顺序执行**和**并发执行**。

1) 顺序执行是单道批处理系统的执行方式。



- **特征：**

- **顺序性**：按照程序结构所指定的次序执行（可能有分支或循环）
- **封闭性**：独占全部资源，计算机的状态只由于该程序的控制逻辑所决定，结果不受外界因素的影响
- **可再现性**：初始条件相同则结果相同。

2.1.1 程序的执行特征

2) 并发执行是为了提高资源利用率

- **特征**（在不加入任何系统控制情况下）：
 - **间断性**：表现为“走走停停”，一个程序可能走到中途停下来，失去原有的时序关系；
 - **失去封闭性**：共享资源，受其他程序的控制逻辑的影响。如：一个程序写到存储器中的数据可能被另一个程序修改，失去原有的不变特征。
 - **失去可再现性**：失去封闭性 - > 失去可再现性；外界环境在程序的两次执行期间发生变化，失去原有的可重复特征。
- **并发执行的条件**：达到封闭性和可再现性（正确性的要求）。

2.1.1 程序的执行特征

并发程序失去可再现性例子

例：讨论共享公共变量的两个程序，
它们执行时可能产生的不同结果。

n:=0

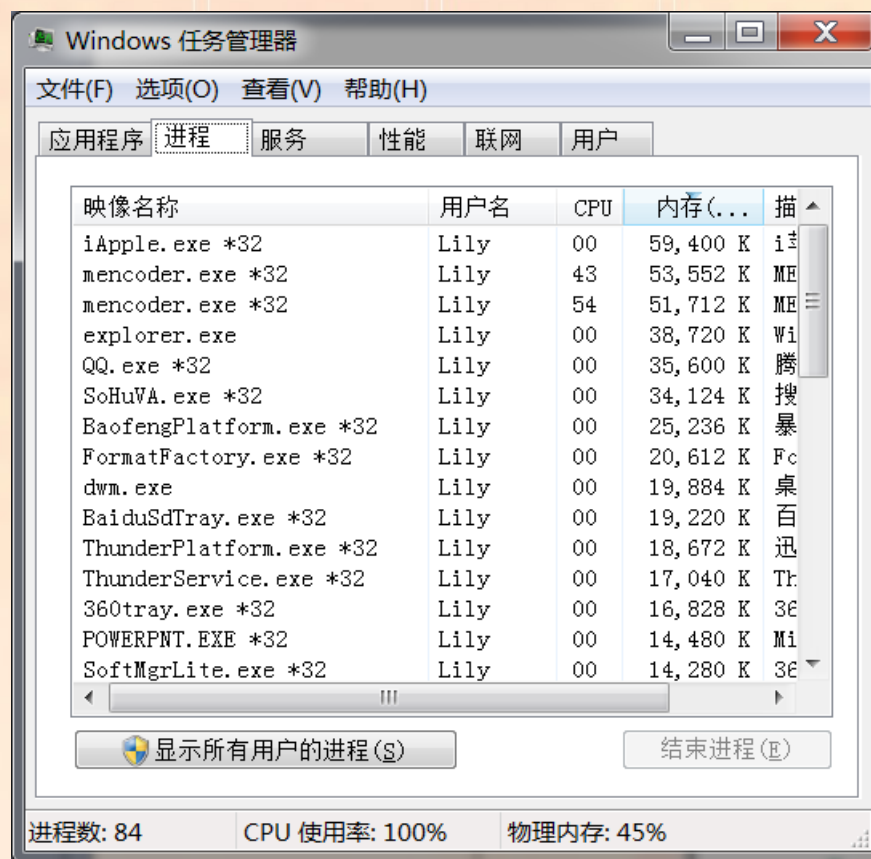
程序 A

```
...  
n := n+1;  
...
```

程序 B

```
...  
print(n);  
n := 0;  
...
```

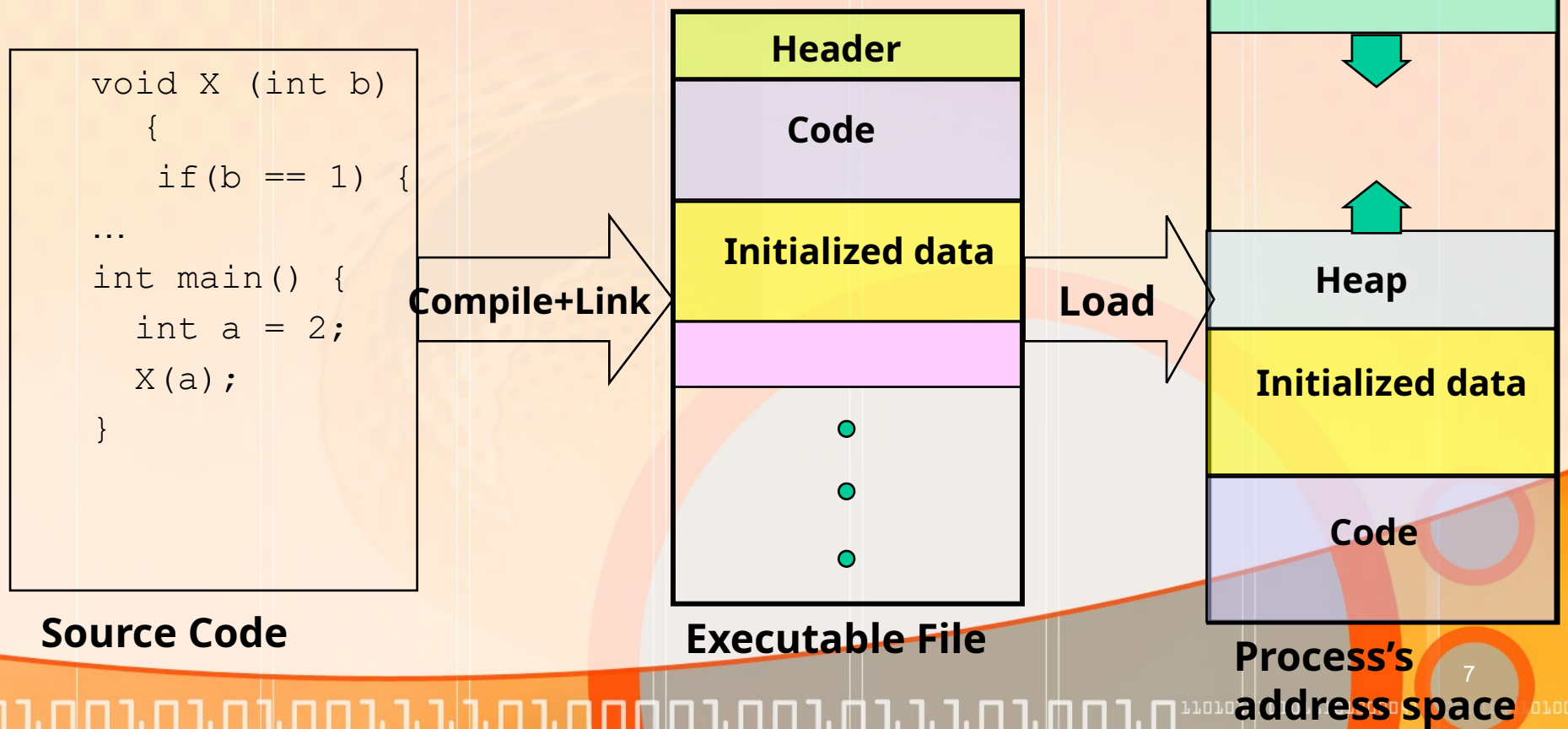
2.2 进程的描述



2.2.1 进程的定义与特征

• 进程：

- 一个具有一定独立功能的程序在一个数据集合上的一次动态执行过程。
- 处理机、存储器和外设等资源的分配和回收的基本单位。



2.2.1 进程的定义与特征

一个进程应该包括：

- 程序的代码；
- 程序处理的数据，堆、栈；
- 程序计数器中的值，指示下一条将运行的指令；
- 一组通用的寄存器的当前值；
- 一组系统资源（如打开的文件）

进程包含了正在运行的一个程序的所有状态信息。

2.2.1 进程的定义与特征

- 进程与程序的关系：
 - 程序是产生进程的基础；进程是程序功能的体现。
 - 程序是**静态**实体；进程是**动态**过程。
 - 程序是指令、数据及其组织形式的描述；进程是程序（那些指令和数据）的真正运行实例。
 - 进程是暂时的，是一个状态变化的过程；程序是永久的，可长久保存。
 - 通过多次执行，**一个程序可产生多个进程**；通过调用关系，**一个进程可包括多个程序**。
 - 进程需要一些资源才能完成工作，如 CPU 使用时间、存储器、文件以及 I/O 设备。

2.2.1 进程的定义与特征

• 进程特征：

- ① **动态性**：创建产生，调度执行，受制于资源，撤销消亡。
 - 进程具有动态的地址空间（数量和内容），地址空间上包括：**代码、数据、进程控制块（PCB）**
- ② **并发性**：多个进程同时存在于内存，宏观上同时运行。
- ③ **独立性**：资源分配的单位。
 - 各进程的地址空间相互独立，除非采用进程间通信手段
- ④ **异步性**：进程按各自独立的、不可预知的速度向前推进。
- ⑤ **结构化**：进程实体由代码段，数据段和进程控制块 (PCB) 组成。**进程控制块 (PCB) + 程序 + 数据 = 进程实体。**
 - 进程控制块包含在**核心区**。内存核心段通常存放 OS 核心部分，由各个进程共享，包括各进程的 PCB。
 - 程序文件中通常划分为代码段和数据段

2.2.1 进程的定义与特征

例：单处理机系统中，可并行的是

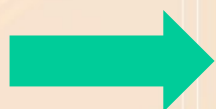
- I 进程与进程 II 处理机与设备**
III 处理机与通道 IV 设备与设备

答： II III IV

进程控制块

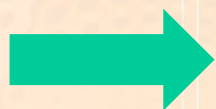
- 描述进程的数据结构：进程控制块 (Process Control Block , PCB)。
- 操作系统为每个进程都维护了一个 PCB ，用来保存与该进程有关的各种状态信息。

PCB
进程控制块



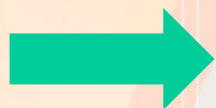
动态特征的集中反映

程序段



描述要完成的功能

数据段



操作对象及工作区



进程控制块

- **PCB:**

- PCB 是 OS 中最重要的记录型结构，记录用于描述进程执行情况与控制进程运行的全部信息。
- PCB 是进程存在的**唯一标志**，是每个进程在 OS 中的登记表项，OS 通过对 PCB 的组织管理来对进程进行控制和管理。
- 进程**创建时生成 PCB**；进程终止时回收 PCB。
- PCB 常驻于由 OS 维护的**内存核心区**，不能由应用程序自身的代码来直接访问，而要通过**系统调用间接访问**。
- OS 专门开辟 PCB 区将所有的 PCB 组织成若干个链表或队列。

PCB 的作用

存放进程管理和控制信息的数据结构称为进程控制块。它是进程管理和控制的**最重要的数据结构**，在创建时，建立 PCB，并伴随进程运行的全过程，直到进程撤消而撤消。

- PCB 就像我们的户口。
- **PCB 是进程存在的唯一标志。**

系统的所有 PCB 组织成链表或队列，常驻内存的 PCB 区。

PCB 中信息：

1. **进程描述信息**：进程标识符，唯一内部标识符；进程名，外部标识符；本进程的产生者标识（父进程标识）；用户标识符，以指示拥有该进程的用户。
2. **处理机状态信息**，保存进程运行现场信息：
 - 通用寄存器 8-32 个，用户程序可使用的数据、地址等寄存器
 - 指令计数器 要访问的下一条指令地址
 - 程序状态字 PSW 条件码、执行方式、中断屏蔽标志
 - 用户栈指针 用户进程拥有的系统栈，存放过程和系统调用参数及调用地址

进程控制块

PCB 中信息（续）：

3. 进程调度信息，用于操作系统调度进程并占用处理机：

- 进程的当前状态；
- 优先级 (priority)；
- 运行统计信息（执行时间、页面调度）；
- 事件：阻塞原因等。

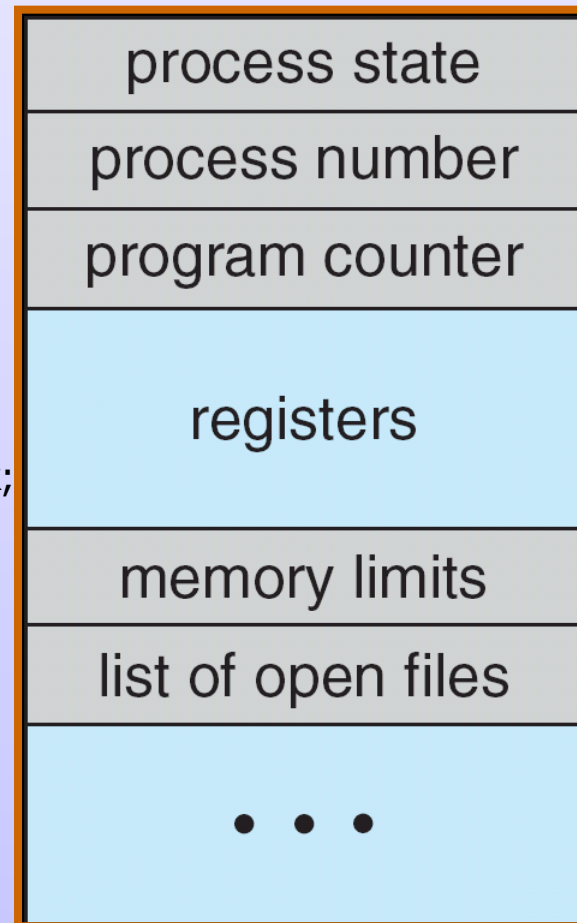
4. 进程控制信息：

- 程序段和数据段的地址；
- 进程间同步和通信；
- 资源占用信息：除 CPU 外的进程所需的全部资源及已分配资源清单
- 链接指针：本进程所在队列的下一个进程的 PCB 首地址。

```

struct task_struct {
/* these are hardcoded - don't touch */
    long state; /* -1 unrunnable, 0 runnable, >0 stopped */
    long counter;
    long priority;
    long signal;
    struct sigaction sigaction[32];
    long blocked;          /* bitmap of masked signals */
/* various fields */
    int exit_code;
    unsigned long start_code,end_code,end_data,brk,start_stack;
    long pid,father,pgrp,session,leader;
    unsigned short uid,euid,suid;
    unsigned short gid,egid,sgid;
    long alarm;
    long utime,stime,cutime,cstime,start_time;
    unsigned short used_math;
/* file system info */
    int tty;                /* -1 if no tty, so it must be signed */
    unsigned short umask;
    struct m_inode * pwd;
    struct m_inode * root;
    struct m_inode * executable;
    unsigned long close_on_exec;
    struct file * filp[NR_OPEN];
/* ldt for this task 0 - zero 1 - cs 2 - ds&ss */
    struct desc_struct ldt[3];
/* tss for this task */
    struct tss_struct tss; // 进程的任务状态段
};

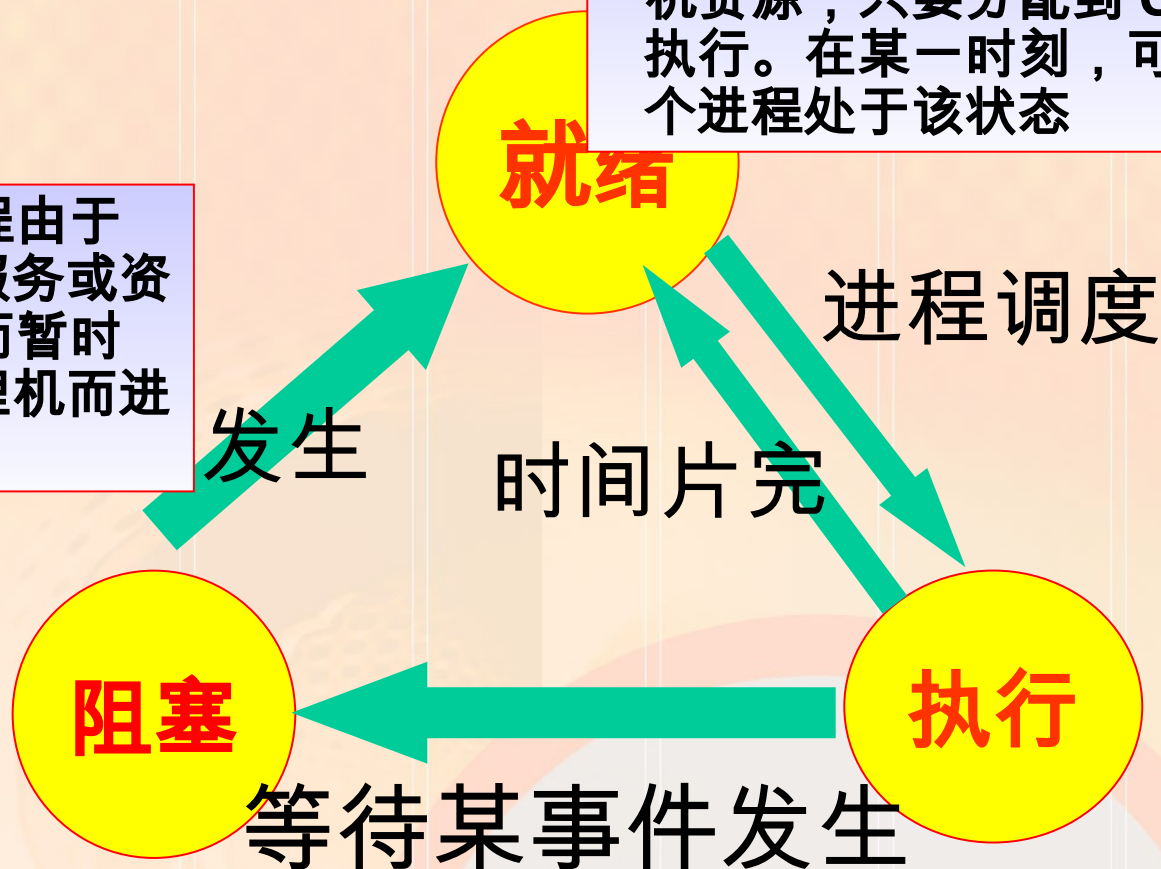
```



2.2.2 进程的基本状态及转换

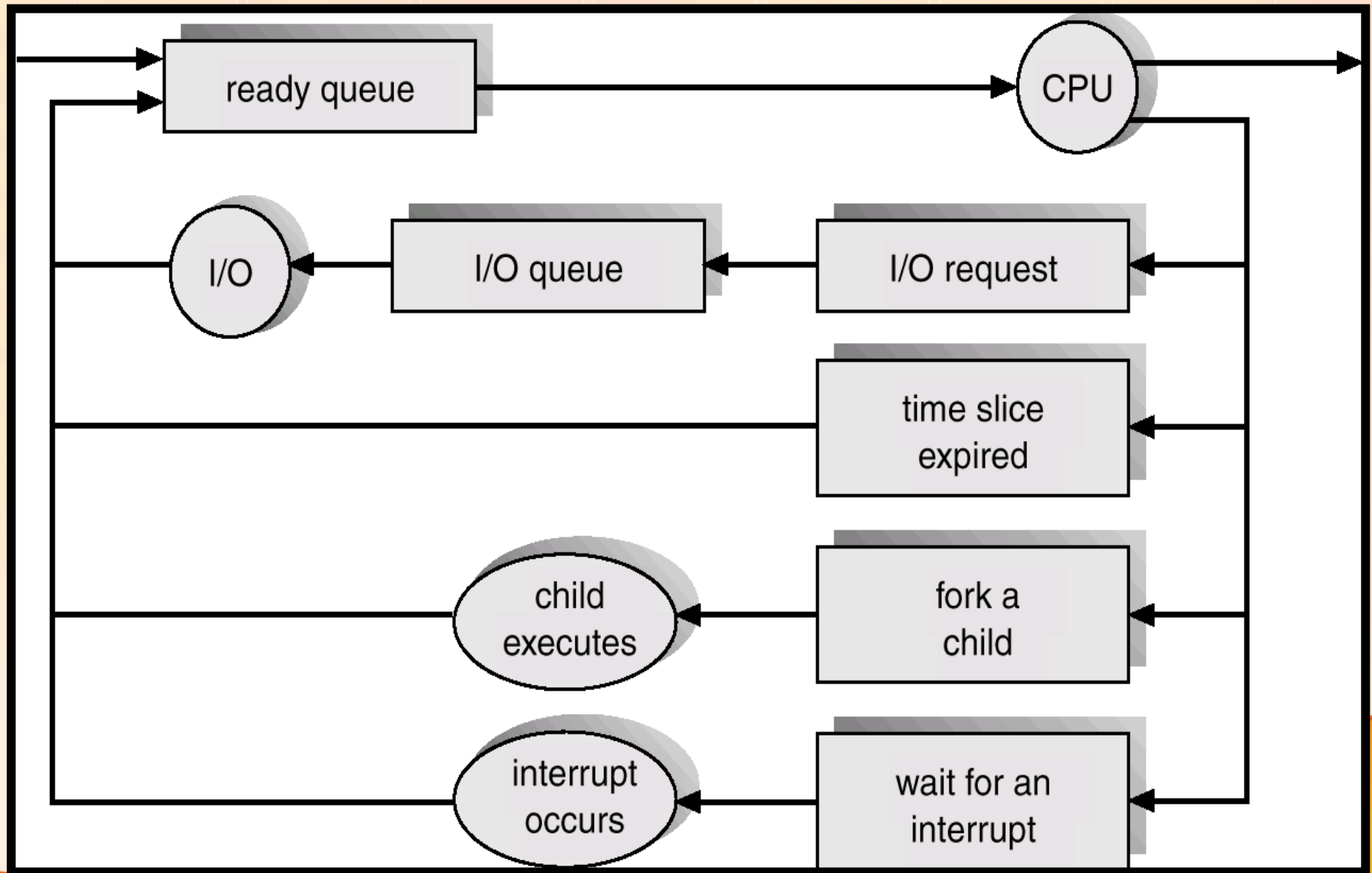
正在执行的进程由于事件，如申请系统服务或资源、I/O 操作等，而暂时放弃处理机而进入阻塞状态，又称等待状态。

就绪状态：进程已获得除处理机外的所需资源，等待分配处理机资源；只要分配到 CPU 就可执行。在某一时刻，可能有若干个进程处于该状态



执行状态：占用处理机资源运行；处于此状态的进程的数目小于等于 CPU 的数目。

2.2.2 进程的基本状态及转换



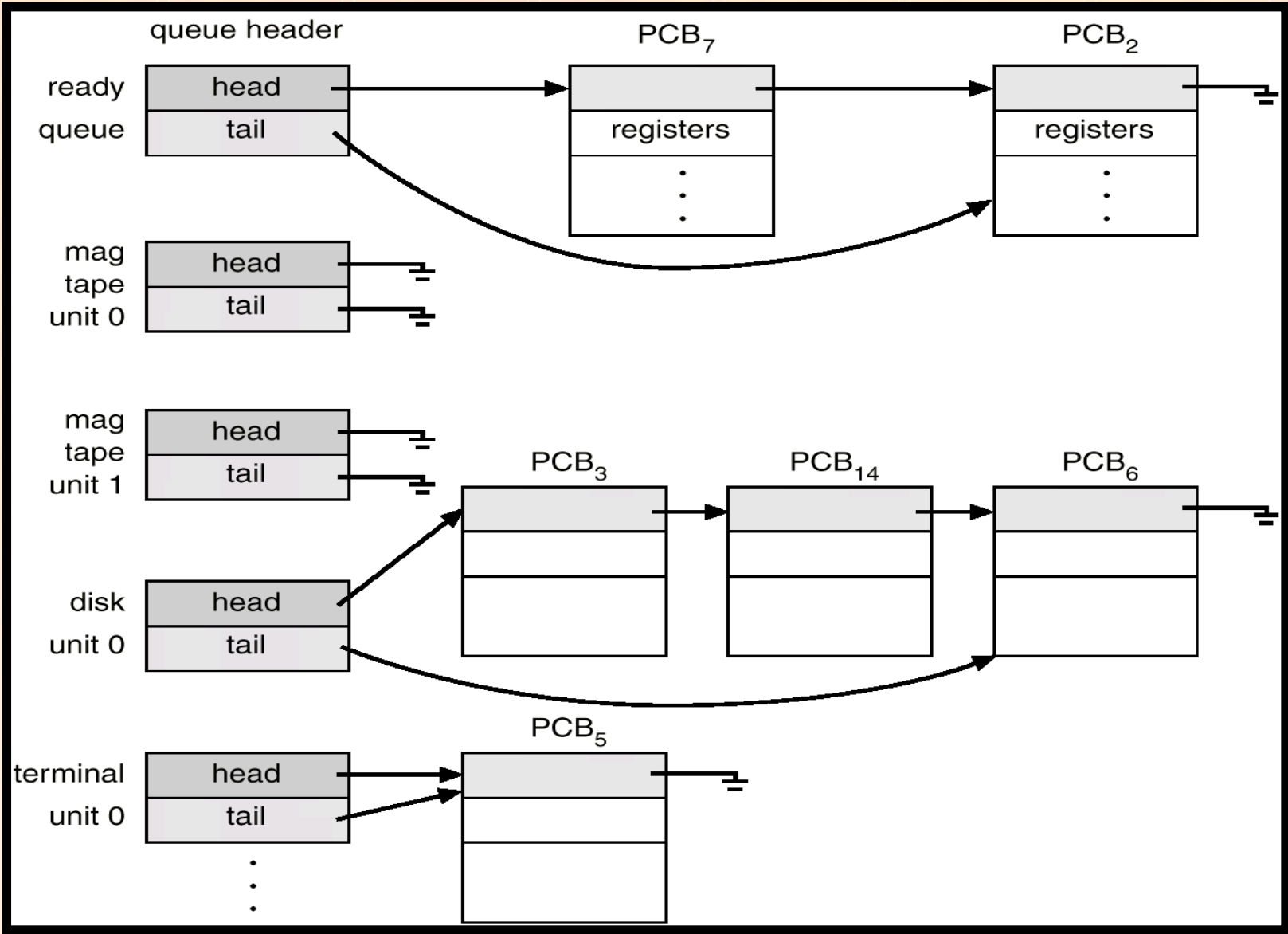
问题：OS 怎么通过 PCB 和定义的进程状态来管理 PCB，帮助完成进程的调度过程？

排队！

进程调度队列

- 由操作系统来维护一组队列，用来表示系统当中所有进程的当前状态
- 不同的状态分别用不同的队列来表示（就绪队列、各种类型的阻塞队列）：
 - 作业队列——系统中所有进程集合
 - 就绪队列——驻留主存的所有就绪态进程集合
 - 设备队列——等待 I/O 设备的进程集合
- 进程在各种队列中迁移：
 - 每个进程的 PCB 都根据它的状态加入到相应的队列当中，当一个进程的状态发生变化时，它的 PCB 从一个状态队列中脱离出来，加入到另外一个队列。

就绪队列和各种 I/O 队列



PCB 组织方式

- 将处于同一状态的 PCB 组织在一起，有两种方式：
 - 链表：同一状态的进程其 PCB 构成一个链表，多个状态对应多个不同的链表，如：就绪链表、阻塞链表
 - 索引表：同一状态的进程归入一个 index 表（由 index 指向 PCB），多个状态对应多个不同的 index 表

链表方式



索引方式

执行指针

就绪表指针

阻塞表指针

就绪索引表

阻塞索引表

PCB1

PCB2

PCB3

PCB4

PCB5

PCB6

PCB7



利用 PCB 完成进程间切换

