

ch4 进程

ch4 进程

进程概念

进程

进程状态

进程控制块 PCB

进程上下文

进程调度

调度类型

调度模式

调度算法(ch6)

进程控制

原语

进程创建

进程终止

进程协作

进程间的联系

进程同步（直接作用）

进程互斥（间接作用）

进程间通信（IPC）

通信类型

高级通信方式

共享存储器系统

消息传递系统

管道通信（共享文件）

进程概念

进程

进程的组成：数据、代码、（PCB进程控制块）、（上下文）

两个进程即使与同一程序相关，但也是独立的

进程特征：并发行，动态性，独立性（进程是资源分配的一个独立单位），交互性，异步性，结构性

进程状态

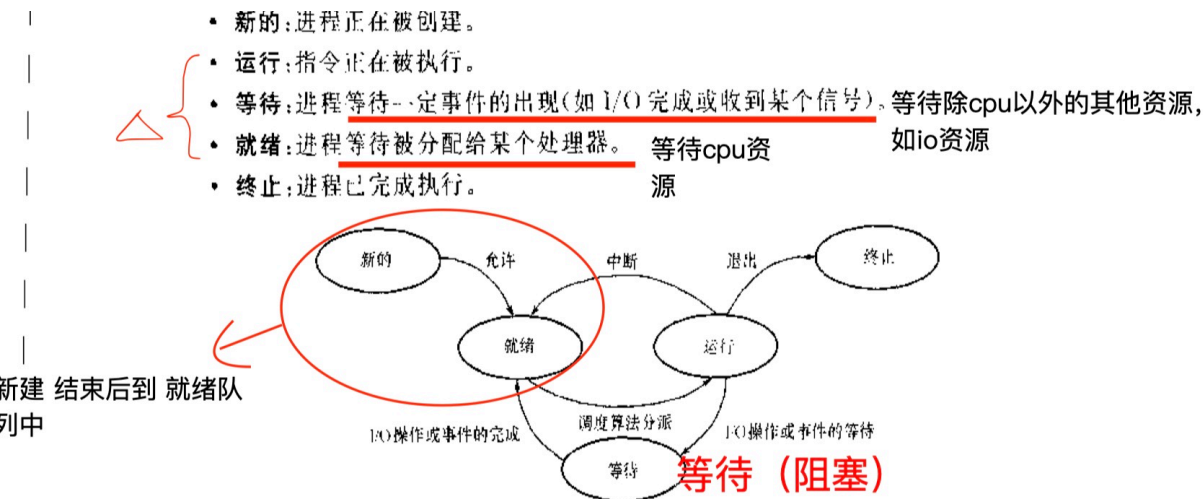


图 4.1 进程状态图

运行->就绪: 中断 (高优先级就绪了), 或用完了时间片

进程控制快 PCB

随意,且随操作系统的不同而变化。不过,它现在所有系统上。有的操作系统更为仔细地何时刻一次只能有一个进程可在任何一个处进程可能处于就绪或等待状态。与这些状态所示。

系统内用进程控制块 (process control block, PCB) 来表示。图 4.2 给出了一个 PCB 的例子。的许多信息。
可包括新的、就绪、运行、等待、停止等。
数据项表示这个进程要执行的下一个指令的地址

指针	进程状态
进程号	
程序计数器	
寄存器	
内存范围	
打开文件列表	
⋮	

图 4.2 进程控制块 (PCB)

进程状态, 程序计数器, 寄存器, 调度信息, 内存管理信息, 记账信息, I/O状态信息

进程间切换:

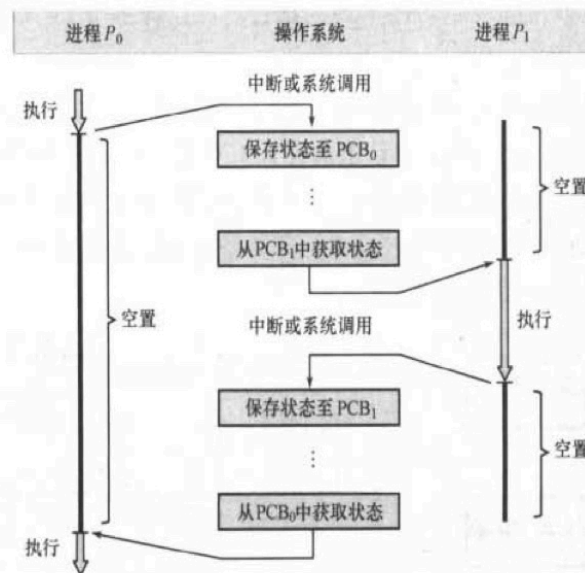
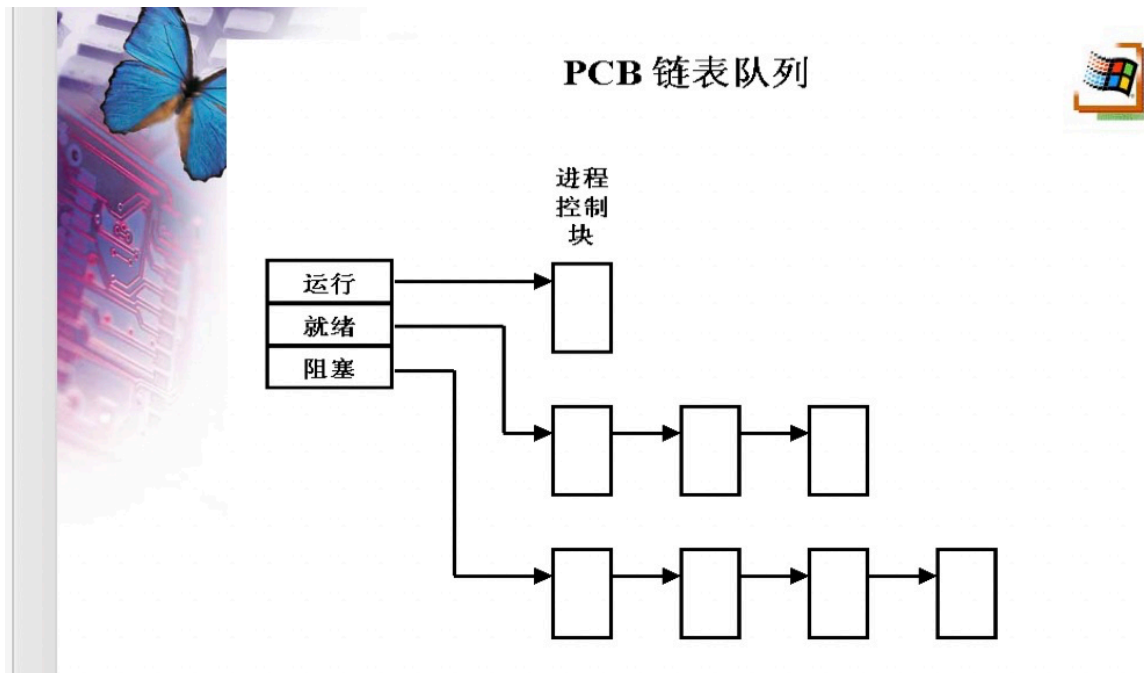
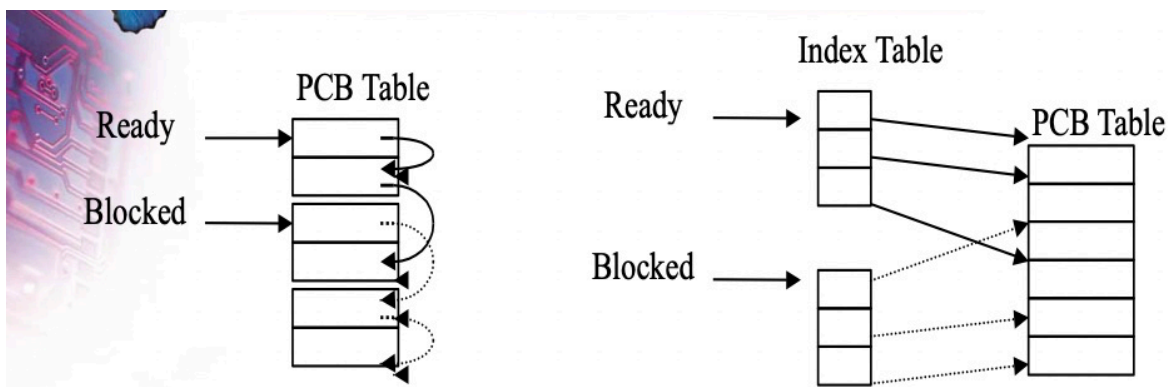


图 4.3 CPU 在进程间的切换图

进程映像（要素）：程序、数据、PCB（执行上下文）、栈

PCB表：内存中 链表/索引表 pcb表的大小决定并发度



进程上下文

- 对进程执行活动全过程的静态描述
- 由进程用户地址空间内容、硬件寄存器内容及与该进程相关的核心数据结构组成

上下文切换：CPU切换进程时，保存原进程的状态，装入新进程的保存状态

进程调度

调度类型

高级调度（长程调度）：作业调度/宏观调度

中级调度（中程调度）：存储器资源管理的角度 设计在内外存间交换

低级调度（短程调度）：微观调度 处理机资源分配的角度

调度模式

非剥夺方式：简单，开销小，实时性差

剥夺方式：原则：优先级原则，时间片原则

调度算法(ch6)

进程控制

原语

- 进程创建原语create()
- 进程撤消原语exit()
- 阻塞原语block()
- 唤醒原语wakeup()
- 挂起原语suspend()
- 激活原语active()

进程创建

创建方式

- 系统程序模块统一创建

- 父进程创建

引起进程创建的事件

- 用户登录
- 作业调度
- 提供服务
- 应用请求

创建内容

1. 申请并创建一个PCB
2. 赋予一个统一进程标识符
3. 为进程映象分配空间
4. 初始化进程控制块
5. 设置相应的链接

如: 把新进程加到就绪队列的链表中

进程终止

父进程可以终止子进程（利用父进程创建子进程是得到的进程标示符）

进程协作

进程间的联系

相交（协作）进程/无关进程

相交进程进一步分：直接作用（同步、通信）/间接作用（互斥）



相互感知程度	交互关系	一个进程对其他进程的影响
相互不感知(完全不了解其它进程的存在)	竞争	一个进程的操作对其他进程的结果无影响
间接感知(双方都与第三方交互，如共享资源)	通过共享进行协作	一个进程的结果依赖于从其他进程获得的信息
直接感知(双方直接交互，如通信)	通过通信进行协作	一个进程的结果依赖于从其他进程获得的信息



进程同步（直接作用）

- 指系统中多个进程中发生的事件存在某种**时序关系**，需要相互合作，共同完成一项任务。进程间，因为在某些位置上需要协调它们的工作而相互等待、相互交换信息所产生的制约关系。
- 也可以说是并发进程互相**共享**对方的私有资源而引起的直接制约。

例：司机和售票员

进程互斥（间接作用）

- 进程间因竞争**共享公有资源**而引起的**间接制约关系**，称为互斥。
- 间接是指：各并发进程的速度受公有资源制约，而不是进程间直接制约。

进程间通信（IPC）

通信类型

低级通信（定长）：只传递状态和整数

高级通信（变长）：任意数量的数据

高级通信方式

共享存储器系统

两种类型：

基于共享数据结构的通信方式：效率低，少量数据

基于共享存储区的通信方式：高级，大量数据

例：生产者与消费者

消息传递系统

两个原语：send，receive

两种类型：

直接通信：如消息缓冲通信方式

间接通信：如信箱

管道通信（共享文件）

基于文件系统，利用一个打开的共享文件连接两个相互通信的进程，文件作为缓冲传输介质。