

进程

进程概念学习

进程的组成

一个进程应该包括:

1. 程序的代码
2. 程序处理的数据
3. 程序计数器的值，指示吓一跳将运行的指令
4. 一组通用的寄存器的当前值，堆，栈
5. 一组系统资源（如打开的文件）

总之，进程包含了正在运行的一个程序的所有状态信息。

进程与程序的联系

1. 程序是产生进程的基础
2. 程序的每次运行构成不同的进程
3. 进程是程序功能的体现
4. 通过多次执行，一个程序可对应多个进程，通过调用关系，一个进程可包括多个程序。

进程与程序的区别

- 进程是动态的，程序是静态的，程序是有序代码的集合，进程是程序的执行，进程有内核态/用户态
- 进程是暂时的，程序是永久的，进程是一个状态变化的过程，程序可以永久保存
- 进程与程序的组成不同：进程的组成包括程序，数据和进程控制块（即进程状态信息）

进程的特点

- 动态性：可动态的创建，结束进程；
- 并发性：进程可以被独立调度并占用处理机运行：并发并行
- 独立性：不同进程的工作不相互影响（通过页表，使得不同的进程访问不同的地址空间）
- 制约性：因访问共享数据/资源或进程间同步而产生制约

PCB

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是进程存在的唯一标志。

进程的创建：为该进程生成一个PCB

进程的终止：回收它的PCB

进程的组织管理：通过对PCB的组织管理来实现

PCB构成

pcb包含以下三大类信息

1. 进程标识信息

如本进程的标识，本进程的产生者标识（父进程标识）用户标识

2. 处理机状态信息保存区：保存进程的运行现场信息

1. 用户可见寄存器：用户程序可以使用的数据，地址等寄存器
2. 控制和状态寄存器：如程序计数器（PC），程序状态字（PSW）
3. 栈指针：过程调用/系统调用/中断处理和返回时需要用到它。

3. 进程控制信息

1. 调度和状态信息：用于操作系统调度进程并占用处理机使用
2. 进程间通信信息：为支持进程间的与通信系统相关的各种标识，信号，信件等这些信息存在接收方的进程控制块中
3. 存储管理信息：包含了有指向本进程映像存储空间的数据结构
4. 进程所有资源：说明由进程打开，使用的系统资源，如打开的文件等
5. 有关数据结构连接信息：进程可以连接到一个进程队列中，或连接到相关的其他进程的PCB

PCB的组织方式

1. 链表（性能更好）：同一状态的进程其PCB成一链表，多个状态用多个不同的链表，各状态的进程形成不同的链表：就绪链表，阻塞链表

方便动态的创建和删除进程

2. 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，各状态的进程形成不同的索引表：就绪索引表，阻塞索引表

进程的生命周期

进程创建

引起进程创建的三个主要事件：

1. 系统初始化时
2. 用户请求创建一个新进程
3. 正在运行的进程执行了创建进程的系统调用

进程运行

内核选择一个就绪的进程，让它占用CPU并执行

进程等待

在以下情况下，进程等待(阻塞)：

1. 需要并等待系统服务，无法马上完成
2. 启动某种操作，无法马上完成
3. 需要的数据没有到达（比如等待读取文件，此时进程等待，将CPU让出给其他等待的进程执行）

进程只能自己阻塞自己，因为只有进程自身才能知道何时需要等待某种事件的发生

进程唤醒

1. 被阻塞进程需要的资源可被满足
2. 被阻塞进程等待的事件到达
3. 将该进程的PCB插入到就绪队列

进程只能被别的进程或操作系统唤醒

进程结束

在以下四种情况：

1. 正常退出(自愿的)
2. 错误退出(自愿的)
3. 致命错误(强制的)
4. 被其他进程所杀(强制的)

进程状态变化模型

进程的三种基本状态：

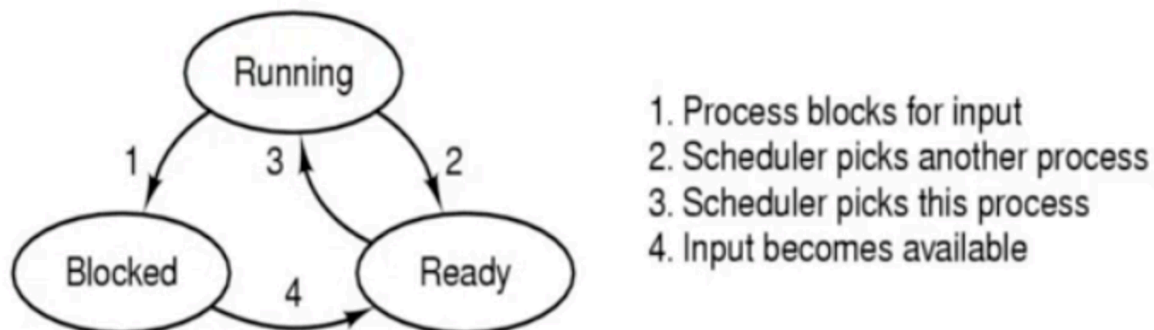
进程在生命结束前处于且仅处于三种基本状态之一

不同系统设置的进程状态数目不同

运行状态(Running): 当一个进程正在处理机上运行时

就绪状态(Ready): 一个进程获得了除处理机之外的一切所需资源，一旦得到处理机即可运行

等待(又称阻塞状态Blocked): 一个进程正在等待某一事件而暂停运行时。如等待某资源，等待输入/输出完成



进程的其他基本状态

创建状态(New): 一个进程正在被创建，还没有被转到就绪状态之前的状态

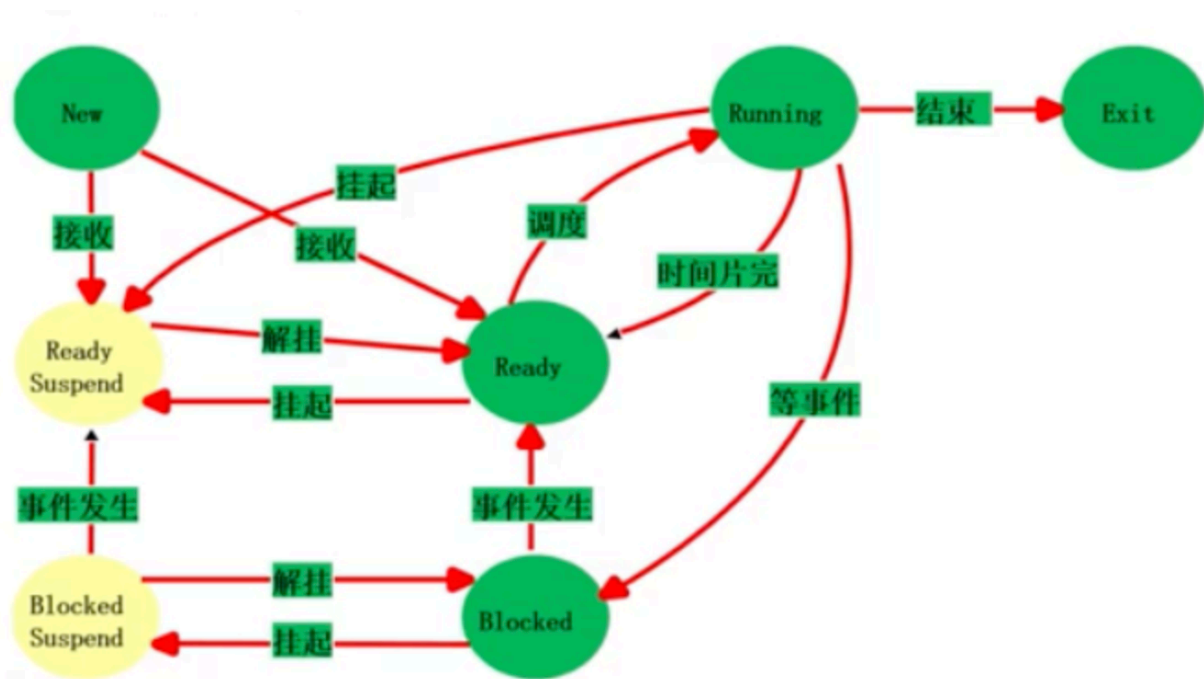
结束状态(Exit): 一个进程正在从系统中消失时的状态，这是因为进程结束或者由于其他原因所导致

状态变化图



进程挂起模型

进程在挂起状态时，意味着进程没有占用内存空间。处于挂起状态的进程映像磁盘上。



挂起状态

- 阻塞挂起状态(Blocked-suspend): 进程在外存并等待某事件的出现
- 就绪挂起状态(Ready-suspend): 进程在外存，但只要进入内存，即可运行

与挂起相关的状态转换

挂起 (Suspend) 把一个进程从内存转到外存，可能有以下几种情况：

- 阻塞到阻塞挂起：没有进程处于就绪状态或者就绪进程要求更多内存资源时，会进行这种转换，腾出内存空间提供给就绪进程使用
- 就绪到就绪挂起：当有高优先级阻塞(系统认为会很快就绪)进程和低优先就绪进程时，系统会选择挂起低优先级就绪进程
- 运行到就绪挂起：对抢先式分时系统，当有高优先级阻塞挂起进程因事件出现而进入就绪挂起时，系统可能会把运行进程转到就绪挂起状态；

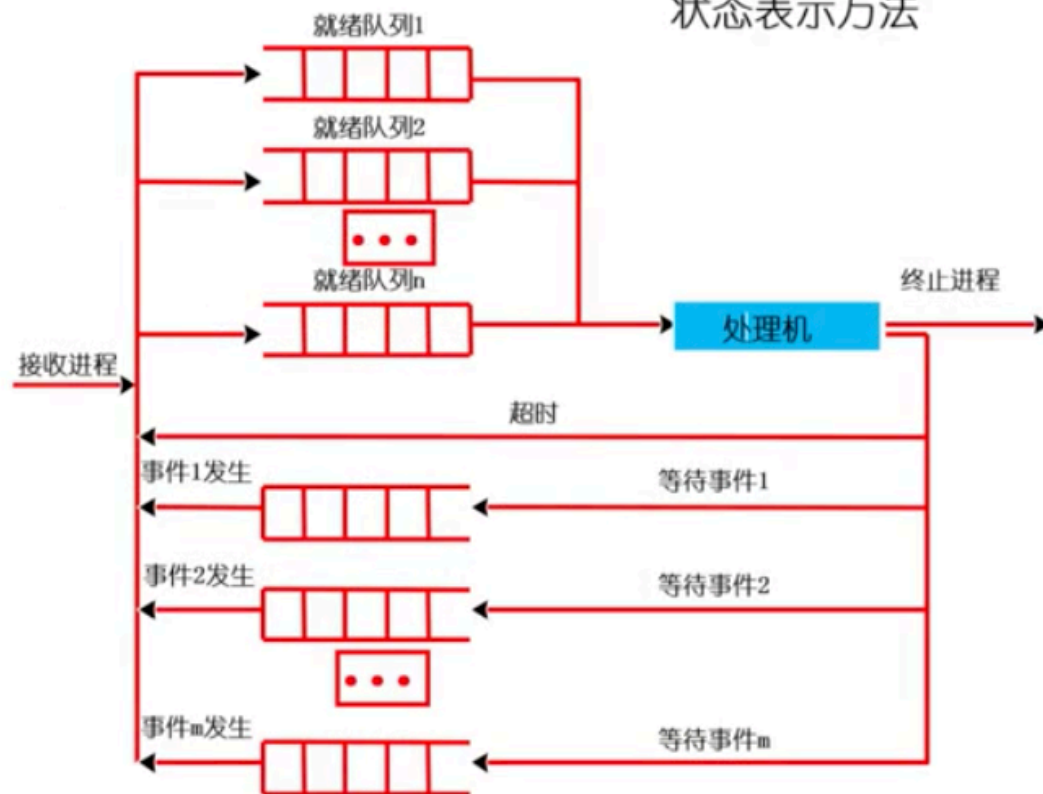
在外存的状态转换

- 阻塞挂起到就绪挂起：当有阻塞挂起进程因相关事件出现时，系统会把阻塞挂起进程转换为就绪挂起进程

状态队列

- 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)
- 每个进程PCB都根据它的状态假如到相应的队列中，当一个进程状态发生变化时，它的PCB从一个状态队列中脱离出来，加入到另一个队列

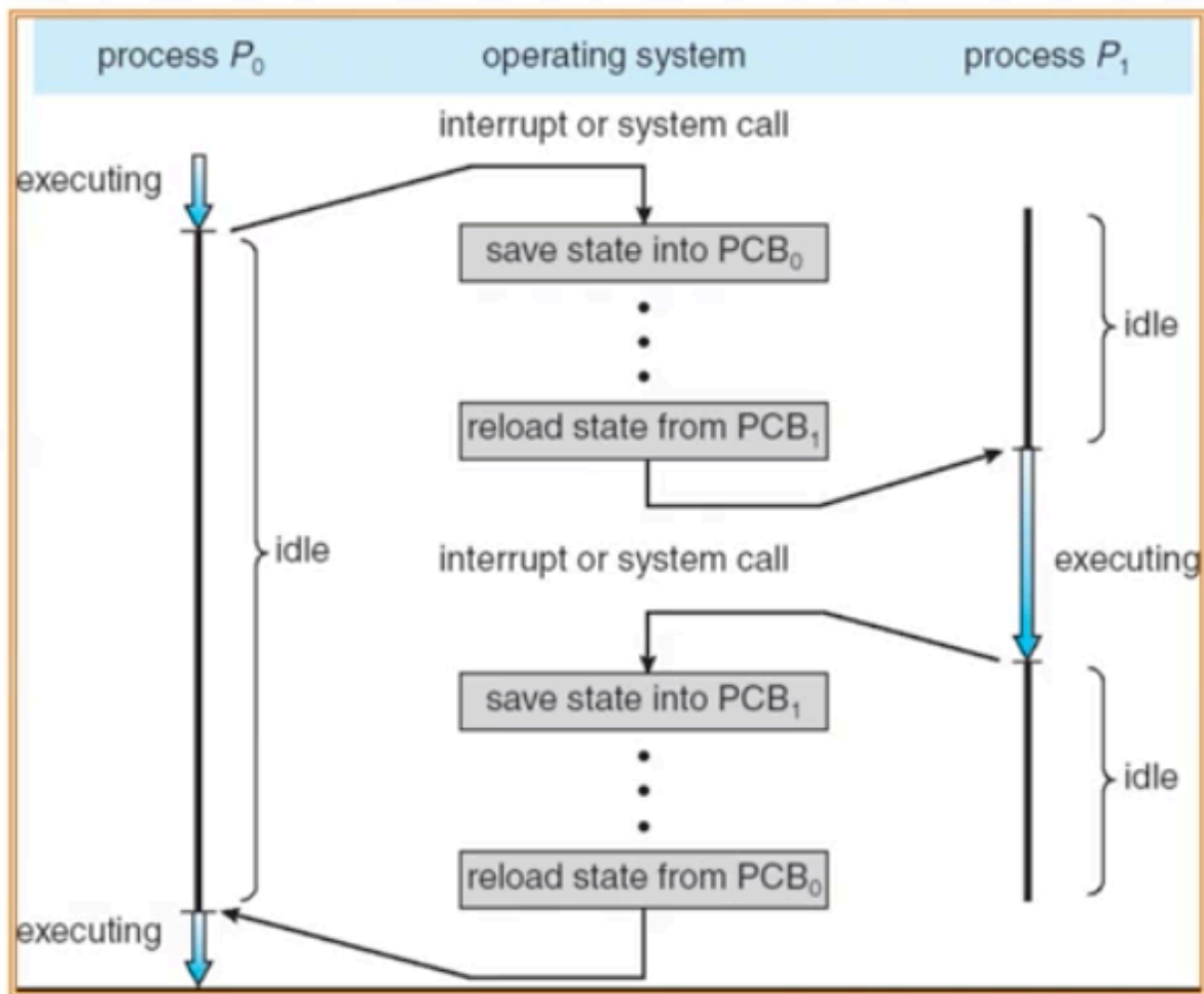
状态表示方法



上下文切换

当一个进程让出CPU资源给另一个进程执行时，被称作进程上下文切换(context-switch)

当一个进程让出Cpu资源时进行上下文切换，将cpu当前的寄存器里的信息保存到自己的pcb当中，然后准备执行的进程会将自己的pcb里的信息load到寄存器当中，完成上下文切换



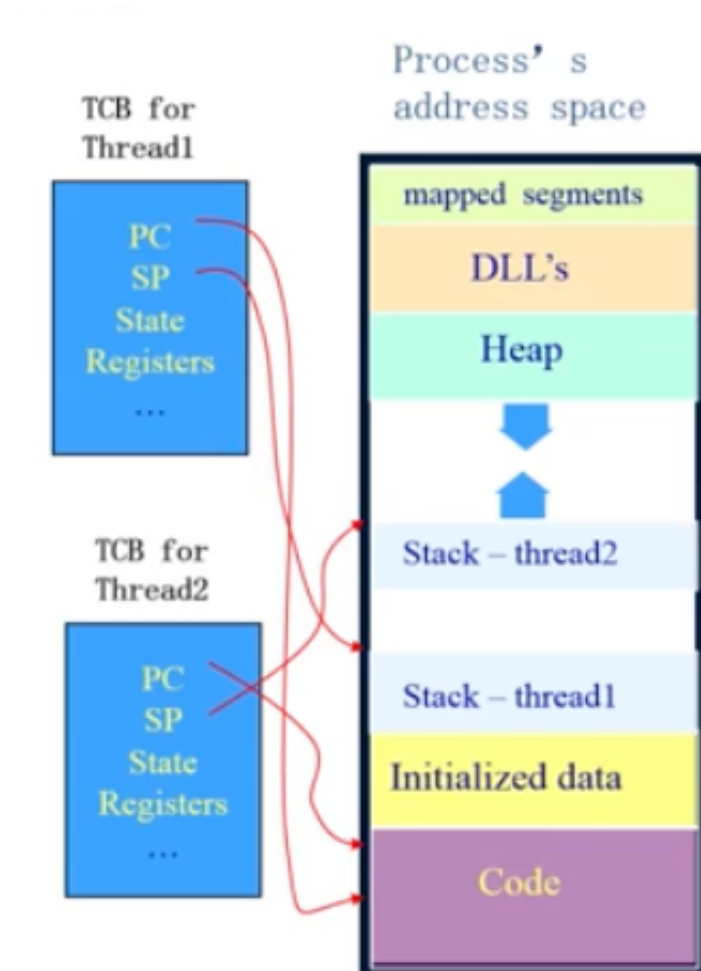
线程管理

定义: 进程当中的一条执行流程。

进程是系统进行资源分配和调度的基本单位, 而线程则是程序执行的最小单位, 线程生活在进程中。

从资源组合的角度: 进程把一组相关的资源组合起来, 构成了一个资源平台(环境), 包括地址空间(代码段, 数据段), 打开的文件等资源

从运行的角度: 代码在这个资源平台上的一条执行流程(线程)



相对于进程有PCB，线程也有TCB(线程控制块),只负责执行流程相关的一系列信息：

PC(程序计数器)

SP(栈)

Registers(寄存器) 需要寄存器来保存控制流的执行状态

公有的部分:代码，数据，堆

线程的优点

- 一个进程中可以同时存在多个线程
- 各个线程之间可以并发地执行
- 各个线程可以共享地址空间和文件等资源

线程的缺点

- 一个线程崩溃，会导致其所属进程的所有进程崩溃