

第2章 进程与线程

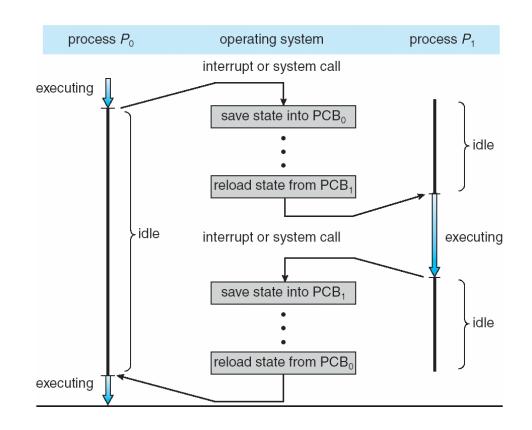
2.7 进程切换与CPU模式切换





■进程切换

- 中断处于运行态的 进程的运行,让出 处理器
- 完成一次进程上下 文切换,即保存老 进程状态而装入被 保护了的新进程的 状态,以便新进程 运行





■进程切换的步骤

- ■保存被中断进程的处理器现场信息
- 修改被中断进程的进程控制块有关信息,如 进程状态等
- ■把被中断进程的PSW加入有关队列
- ■选择下一个占有处理器运行的进程
- ■修改被选中进程的PSW的有关信息
- 根据被选中进程设置操作系统用到的地址转 换和存储保护信息
- ■根据被选中进程恢复处理器现场



-

- ■调度和切换时机问题
 - ■请求调度的事件发生后,就会运行低级调度程序,低级调度程序选中新的就绪进程后,就会进行上下文切换。实际上,由于种种原因,调度和切换并不一定能一气呵成。
 - 通常的做法是,由内核置上请求调度标志, 延迟到敏感性操作完成后再进行调度和进程 上下文切换,
 - Linux 进程调度标志位 need-resched, Windows延迟过程调用 DPC/dispatch软件中断。

-

■处理器模式切换

- ■当中断/系统调用发生时,暂时中断正在执行的用户进程,把进程从用户状态切换到内核状态,去执行操作系统服务程序以获得服务,这就是一次模式切换,
- ■此时,内核仍然在被中断进程的上下文中进行处理



- CPU上所执行的进程所处活动范围:
 - ■用户空间中,处于进程上下文,用户进程在 运行,使用用户栈。
 - ■进程正常运行
 - ■内核空间中,处于进程上下文,内核代表某 进程在运行,使用核心栈。
 - ■进程请求系统调用,陷入内核
 - ■内核空间中,处于中断上下文,与任何进程 无关,中断服务程序正在处理特定中断。
 - ■进程被中断



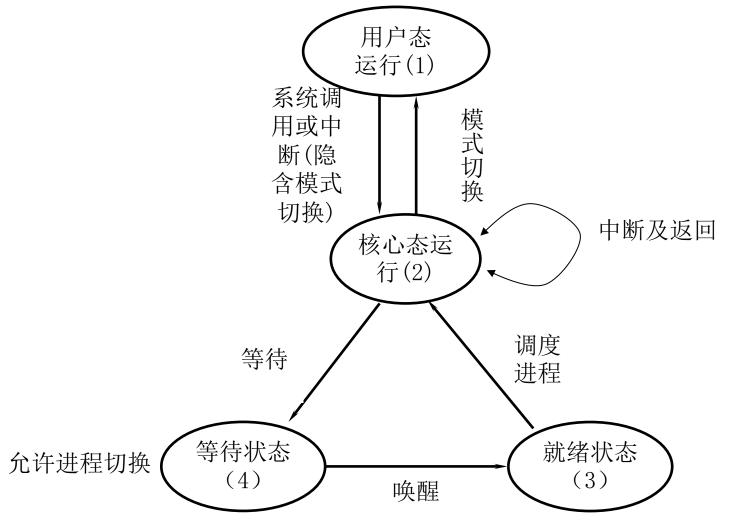
■Linux进程与任务

- Linux把内核空间中运行的程序称为任务, 而在用户空间中运行的程序称为进程。
- 系统中存在两种进程(任务): 系统进程(任务) 和用户进程(任务), 实质上是指一个进程(任务) 务)的两个侧面,。
- ■两个进程(任务)所执行的程序不同,映射到 不同物理地址空间、使用不同的堆栈。



-

■ UNIX/Linux中上下文切换和模式切换







■讨论

■ 请围绕进程的7态模型来讨论一下,处理器模式转换和进程状态转换是怎样的关系?

