

### 3.10 为什么需要两种模式？

答：使用两种模式的原因是保护操作系统和重要的操作系统表(比如进程控制块)不受用户程序的干扰。在内核模式下, 软件会完全控制控制处理器及其所有命令、寄存器和内存。为安全起见, 这种级别的控制对用户程序而言没有必要。

### 3.11 操作系统创建一个新进程的步骤是什么？

答：

1. **为新进程分配一个唯一的进程标识符。**此时, 主进程表会添加一个新表项, 每个进程一个表项
2. **为进程分配空间。**包括进程映像中的所有元素
3. **初始化进程控制块。**进程标识部分包括进程 ID 和其他相关 ID 比如父进程的 ID; 处理器的状态信息部分的多数项目通常初始化为 0 但是程序计数器(置为程序入口点)和系统栈指针(定义进程栈边界)除外。
4. **设置正确的链接。**例如, 若操作系统将每个调度队列都维护为一个链表, 则新进程必须放在就绪或者就绪/挂起链表中
5. **创建或扩充其他数据结构。**

3.3 图中包含 7 个状态。原则上, 若任意两个状态间进行转换, 则可能由 42 种

## 不同的转换

- a. 列出所有可能的转换，并举例说明什么事件会触发这些状态转换
- b. 列出所有不可能的转换并说明原因
  1. 新建态→运行态：无法跳过就绪的准备工作之间到运行态
  2. 新建态→阻塞态：新建态需要先进入就绪或就绪/挂起态完成进程的初始化后，才能依据是否需要等待事件进入阻塞态
  3. 新建态→阻塞/挂起态：新建态需要先进入就绪或就绪/挂起态完成进程的初始化后，才依据是否需要等待事件以及内存不足进入阻塞/挂起态
  4. 就绪态→新建态：进程初始化后就不会再回到新建态
  5. 就绪态→阻塞态：阻塞态是运行态需要等待事件时的状态，而就绪态还没有被调度执行，并不知道需不需要等待事件
  6. 就绪态→阻塞/挂起态：倘若需要挂起，应该是就绪/挂起态
  7. 就绪/挂起态→新建态：进程初始化后就不会再回到新建态
  8. 就绪/挂起态→运行态：全部的内存和程序必须装入内存后(也就是激活)才可能进入运行态
  9. 就绪/挂起态→阻塞态：阻塞态的进程位于内存中而且是运行过需要等待事件，就绪/挂起态必须先进入就绪态进入运行态后才可能变为阻塞态
  10. 就绪/挂起态→阻塞/挂起态：在挂起状态不会转变为阻塞状态的
  11. 阻塞态→就绪/挂起态：需要经过中间态阻塞/挂起态或者就绪态
  12. 阻塞态→运行态：需要等待的事件发生后进入就绪态等待 OS 的调度才能进

入运行态

- 13. 阻塞/挂起态→就绪态：需要经过中间态阻塞态或者就绪/挂起态
- 14. 阻塞/挂起态→运行态：需要先变成就绪态才可能由 OS 调度变为运行态
- 15. 运行态→阻塞/挂起态：需要经过中间态阻塞态
- 16. 退出态→新建态：进入退出态后 PCB 清除，整个流程结束
- 17. 18. 19. 20. 21. 22. 23. 退出态→运行态，就绪态，就绪/挂起态，阻塞/挂起态，  
阻塞态：退出态需要经过下一次新建态后才能进入这些状态

3.4 画出 7 状态进程模型的排队图