

HW 3.

复习题

3.2 触发进程创建的事件通常有4个,分别为新的批处理作业,交互登录,为提供服务而由操作系统创建,由现有进程派生。

3.3 运行态: 进程占有处理器正在运行的状态, 进程已获得CPU, 其程序正在执行。  
就绪态: 进程具备运行条件, 等待系统分配处理器以便运行的状态。  
阻塞/等待态: 进程不具备运行条件, 正在等待某些事件发生(如请求I/O而等待I/O完成)的状态。

新建态: 进程被创建, 尚未进入就绪队列的状态。

退出态: 进程到达自然结束点时, 因~~正常~~出现不可恢复的错误而异常终止或者被具有相应权限的另一个进程取消而终止的状态。

习题

3.3 a. 新建态 → 就绪态: 创建新进程, 且操作系统有足够空间分配给新进程。  
新建态 → 就绪/挂起态: 创建新进程, 且操作系统没有足够空间分配给新进程。

就绪态 → 就绪/挂起态: 为释放内存来得到足够空间, 而挂起一个就绪态进程。

就绪态 → 运行态: 选择一个处于就绪态的进程运行。

就绪/挂起态 → 就绪态: 内存中没有就绪态进程或就绪/挂起态的某一进程比处于就绪态的所有进程优先级都更高时, 进行就绪/挂起态到就绪态的转换。

阻塞态 → 就绪态: 所等待的事件发生后, 处于阻塞态的进程转换到就绪态。

阻塞态 → 阻塞挂起态: 将阻塞态进程挂起, 为另一未阻塞进程腾出空间。

阻塞/挂起态 → 就绪/挂起态: 等待的事件发生后, 可发生这样的转换。

阻塞/挂起态 → 阻塞态: 一个进程终止后, 会释放一些内存空间, 而阻塞/挂起态队列中有一个进程的优先级比就绪/挂起队列中任何进程的优先级都高, 并且操作系统有理由相信阻塞进程的事件很快就会发生。

运行态 → 就绪态: 进程自愿释放对处理器的控制

运行态 → 就绪/挂起态: 一个运行进程的分配时间到期, 且存在阻塞/挂起态进程优先级较高的。

运行态 → 阻塞态: 进程请求其必须等待的某些事件时, 则进入阻塞态。  
不再被阻塞的进程时

各种状态(退出态外的6种状态) → 退出态: 进程被父进程终止, 或在父进程终止时终止。

b.

新建态 → 阻塞态/阻塞挂起态/运行态: 一个新创建的进程在处理器准备好接收一个新进程前处于新建态, 被接收后处于就绪态。  
不能直接进入运行态。如果存在需等待的事件, 则在之后进入阻塞态。

就绪态 → 阻塞态/阻塞挂起态/新建态: <sup>显然</sup>不能进入新建态。一般情况下, 一个



已经处于就绪态的进程不能在运行前突然发现需等待的事件或转入阻塞态、阻塞/挂起态。

就绪/挂起态  $\rightarrow$  阻塞态/阻塞/挂起态/新建态：原因与就绪态转其它态相同

阻塞态  $\rightarrow$  就绪/挂起态：若要发生这样的状态转换需分2步进行，阻塞态不能一次性实现转换到就绪并被挂起，因为造成这样两个状态转换的原因不同，需要分两次分别作状态转移。

阻塞态  $\rightarrow$  运行态：阻塞态不被阻塞后，必定要先经过就绪态，才能进入运行态。

$\downarrow$  阻塞态  $\rightarrow$  新建态：显然不可能。

阻塞/挂起态  $\rightarrow$  就绪态：与“阻塞态  $\rightarrow$  就绪挂起态”原因类似。

阻塞/挂起态  $\rightarrow$  运行态：与“阻塞态  $\rightarrow$  运行态”原因类似。

运行态  $\rightarrow$  阻塞/挂起态：这一状态转移需分2步进行，原因类似“阻塞/挂起态  $\rightarrow$  运行态”

退出态  $\rightarrow$  各种状态：不能在进程终止后，又回到进程执行或等待中的状态。

访问：

3.7 a. 优点在于：能够更灵活地控制存取内存的途径，可以更好地管理内存、协调操作系统的进程以达到更高的效率。

缺点在于：更多的模式意味着更复杂的系统结构和处理过程，可能会占用更大的空间，开销更大。

b. 4种以上的模式似乎很难实现。

3.9 a. 可行, 例如一个进程下一步需要获得从另一进程中写出的数据并保存到磁盘。若此时有另外的数据正在写出而不能执行这一进程, 磁盘又正在被使用, 那么这一进程就同时在等待数据写出和磁盘可用两个事件。

b. 可以将进程放入两个(多个)独立的队列中, 等待事件发生。