

## E07: UNIX 进程管理

### 简答题:

1. UNIX 中存在核心态进程从运行状态转变为就绪状态的情况吗？如果有这种情况请给出实例，若无请给出理由。存在用户态进程从运行状态转变为就绪状态的情况吗？
2. 请说明核心态就绪和用户态就绪的区别。
3. UNIX 中抢占调度和非抢占调度分别在什么情况下发生？

1.在UNIX中，核心态进程从运行状态转变为就绪状态的情况是存在的。这通常发生在进程正在等待系统调用或等待资源时被阻塞的情况下。在执行swtch切换调度时，0#进程会经常从运行状态转变为就绪状态，因为0#进程被更适合上台的进程抢占，0#进程因此进入就绪队列。

至于用户态进程从运行状态转变为就绪状态的情况，这也是可能的。例如，当进程等待输入或等待另一个进程完成任务时，它可能会被迫等待，直到其他进程释放资源才能继续处理。

2.

两个就绪状态的进程核心栈不同，再次上台后的任务不同，进程优先数不同核心栈就绪对应着非抢占调度，设置优先数<100，由于是非抢占调度，是进程在执行系统调用处理时使用外设而入睡，在中断或系统调用入口程序最后，PA会回来执行之前没有完成的核心态代码，而抢占调度下，进程的核心栈只剩下swtch和系统调用栈帧，进程被迫让出处理机，进入就绪状态，由于它已经完成了核心态的全部工作，所以再次上台后将返回用户态执行

尽管二者有很大的区别，但是它们当前都处在核心态。

在UNIX系统中，抢占调度发生的情况包括：

一个更高优先级的进程变得可运行，并且当前进程不是在关键区域内。

当前进程已经使用了其分配的时间片，并且一个新的进程已经准备好运行。

当前进程正在等待某个事件发生，而该事件已经发生并且有一个更高优先级的进程准备运行。

非抢占调度发生的情况包括：

进程主动放弃CPU，并请求将其状态设置为就绪状态。

进程正在等待某个事件发生，并且操作系统没有其他任务需要执行。

实时任务被分配了CPU，并且需要保证具有固定的响应时间。

存在核心态进程从运行状态转变为就绪状态的情况。UNIX 中的进程状态转换都是发生在核心态下的。一个用户态下执行的进程，因为响应中断请求或系统调用，而由用户态进入核心态执行。由于中断或系统调用返回用户态前的例行调度，可能会下台，这时候，进程的状态由运行状态转变为就绪状态。

需要注意的是，这时，进程在核心态下的工作都已经完成，是一个马上要回到用户态的进程，因为我们把这种就绪状态称为用户态就绪。用户态就绪的进程优先数是由计算获得的，它和核心态就绪是不同的。核心态就绪指的是进程由于系统调用入睡，醒来之后进入就绪状态，这时进程优先数是设置获得的，优先级较高，因而可以尽快上台完成后续的核心态下需要完成的工作。不存在用户态进程从运行状态转变为就绪状态的情况。因为UNIX 中的进程状态转换都是发生在核心态下的。

非抢占调度：现运行进程入睡或终止。抢占调度：现运行进程用户态运行时响应外设中断

或进行系统调用进入核心态，即将返回用户态时例行调度，如果RunRun 标志被设置，则发生抢占调度。