

作业 2

2019011008

无 92 刘雪枫

1. 进程不能从就绪态转换为阻塞态。因为就绪态的进程没有在执行，因此不会进行 I/O 等等操作导致其阻塞，因此不会转换为阻塞态。
2. 处于运行状态的进程不超过 4 个；处于就绪状态的进程不超过 $(n-4)$ 个；处于阻塞状态的进程数为 $0 \sim n$ 个。
3. ULT 在进行线程切换时不需要陷入到内核，因此相对于内核级线程来说具有较高的效率。
4. 因为每个线程都要进行自己的过程调用、传递参数、返回地址、使用局部变量，而这些都要依赖栈去完成，因此每个线程都要有自己的栈。对于内核级线程，线程由内核进行管理，每个线程应当独立地被内核调度，当一个线程进行系统调用等操作或是发生异常时，其他线程应当不受影响，因此每个线程都应当存在一个核心栈。而用户级线程则不需要每个线程都有自己的核心栈，因为内核并不知道线程的存在，一个进程内所有线程对于内核来说都是同一个进程，对于系统调用或异常，内核也是对整个进程进行处理，因此不需要每个线程都有一个核心栈，但是每个线程仍然具有自己的用户栈。
5. 如果是单线程，则需要处理每个请求平均花费时间为 $15 + 1/3 * 75 = 40\text{ms}$ ，则每秒可以执行 $1000/40 = 25$ 个请求；
如果是多线程，由于平均处理磁盘的时间为 $1/3 * 75 = 25\text{ms}$ ，而 CPU 的时间平均为 15ms ，两者可以并行，因此每秒可以执行 $1000/25 = 40$ 个请求。
6. 对于 Linux 的线程实现方式来说，Linux 不单独实现线程，而是用可以共享资源的进程实现线程；而 Windows 则区分了线程与进程。Linux 的实现方式更加简洁，使用起来更加简单；而 Windows 的实现方式使得创建线程的代价小于创建进程的代价，因此可以根据自己的实际需求选择使用线程还是进程，从而最大限度地提高程序运行的效率与安全性。