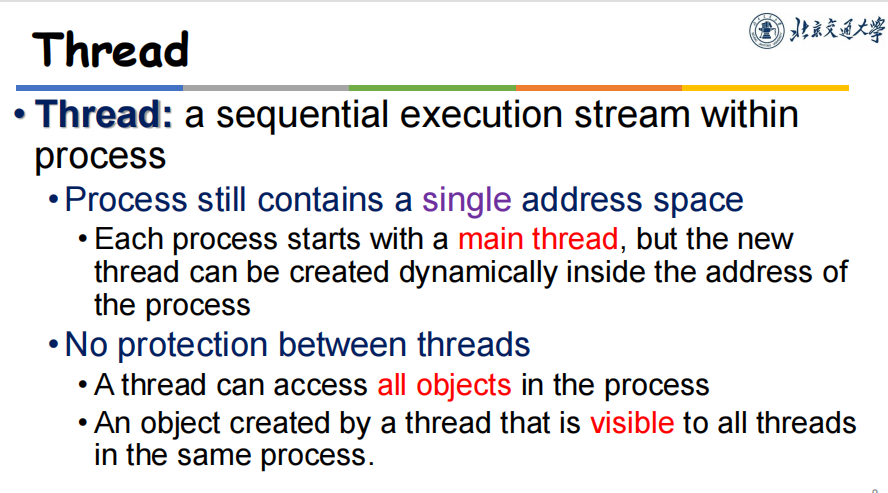
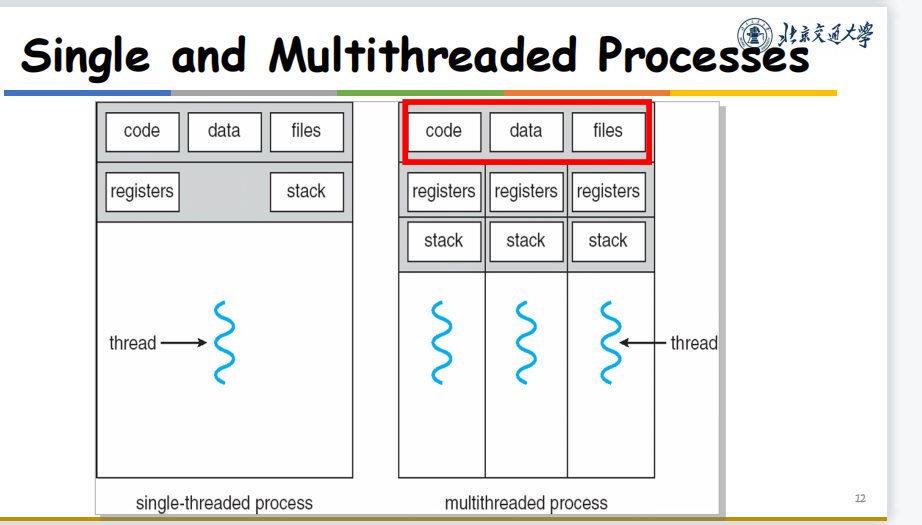
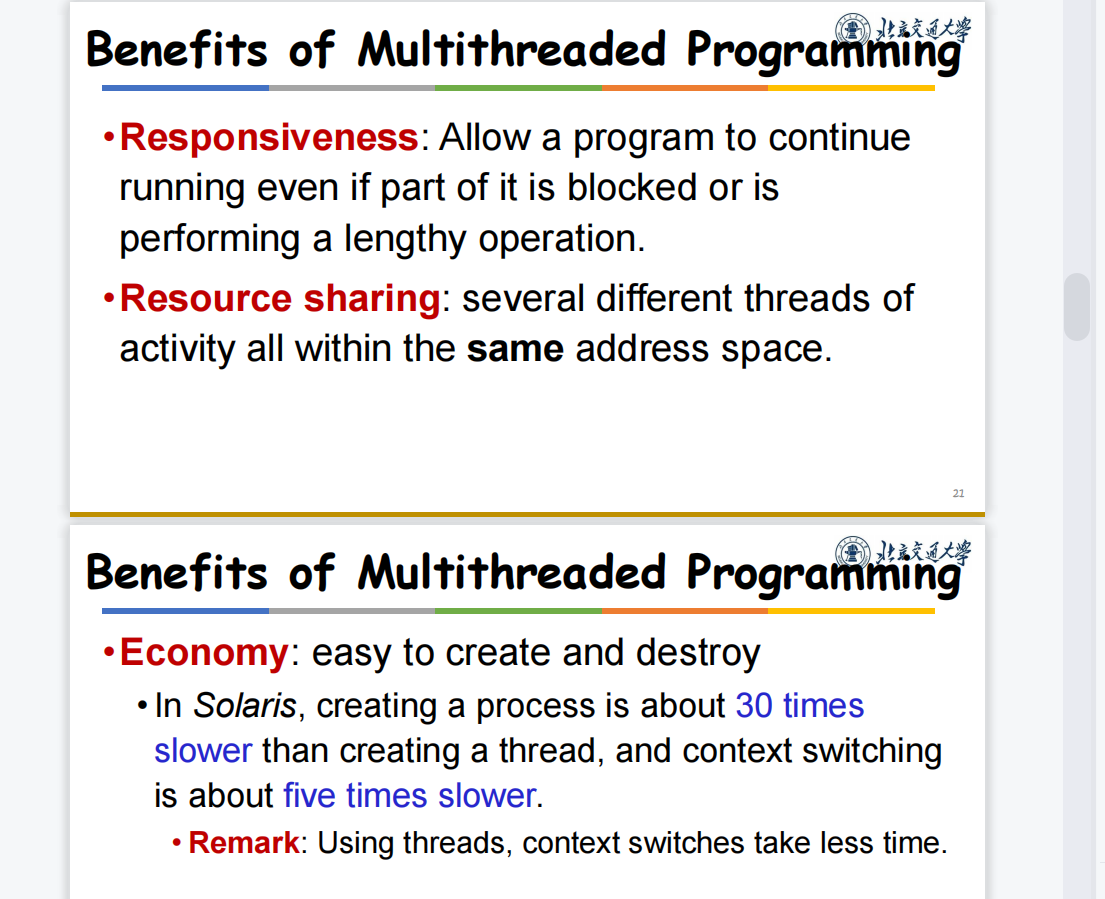
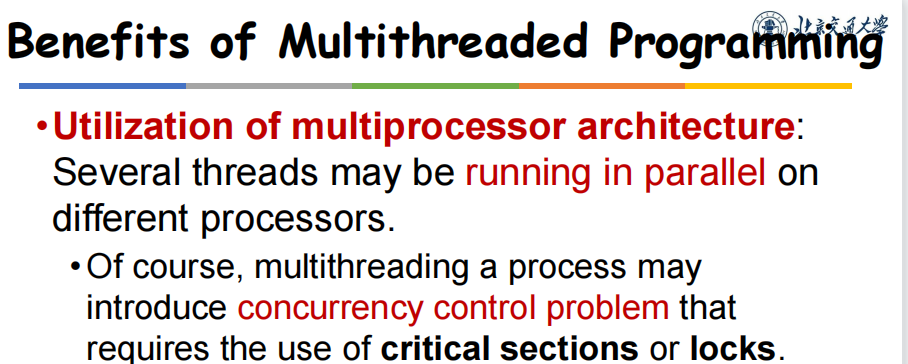


其他线程的空间会在Process里面

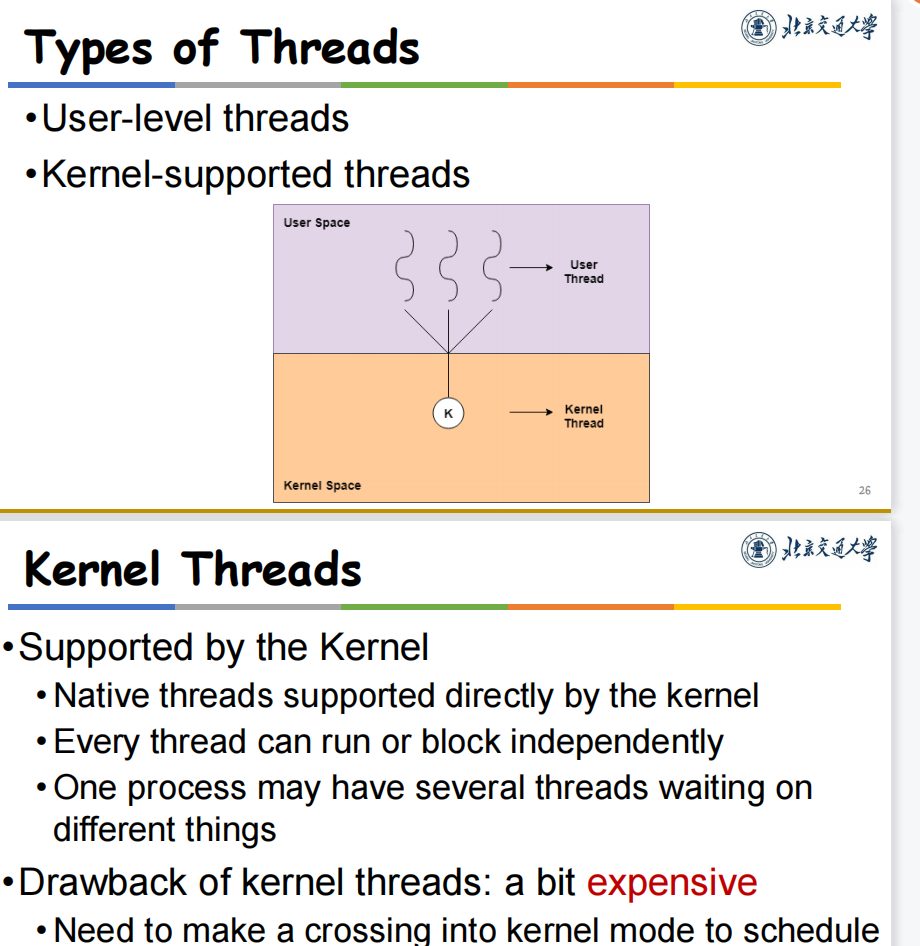








高级之kernel线程



线程的实现可以分两类：用户级线程，内核级线程和混合式线程。

用户级线程是指不需要内核支持而在用户程序中实现的线程，它的内核的切换是由用户态程序自己控制内核的切换，不需要内核的干涉。但是它不能像内核级线程一样更好的运用多核CPU。

优点：

（1） 线程的调度不需要内核直接参与，控制简单。

（2） 可以在不支持线程的操作系统中实现。

（3） 同一进程中只能同时有一个线程在运行，如果有一个线程使用了系统调用而阻塞，那么整个进程都会被挂起，可以节约更多的系统资源。

缺点：

（1） 一个用户级线程的阻塞将会引起整个进程的阻塞。

（2） 用户级线程不能利用系统的多重处理，仅有一个用户级线程可以被执行。

内核级线程:切换由内核控制，当线程进行切换的时候，由用户态转化为内核态。切换完毕要从内核态返回用户态。可以很好的运用多核CPU，就像Windows电脑的四核八线程，双核四线程一样。

优点：

（1）当有多个处理机时，一个进程的多个线程可以同时执行。

（2） 由于内核级线程只有很小的数据结构和堆栈，切换速度快，当然它本身也可以用多线程技术实现，提高系统的运行速率。

缺点：

（1） 线程在用户态的运行，而线程的调度和管理在内核实现，在控制权从一个线程传送到另一个线程需要用户态到内核态再到用户态的模式切换，比较占用系统资源。（就是必须要受到内核的监控）

关联性

（1） 它们之间的差别在于性能。

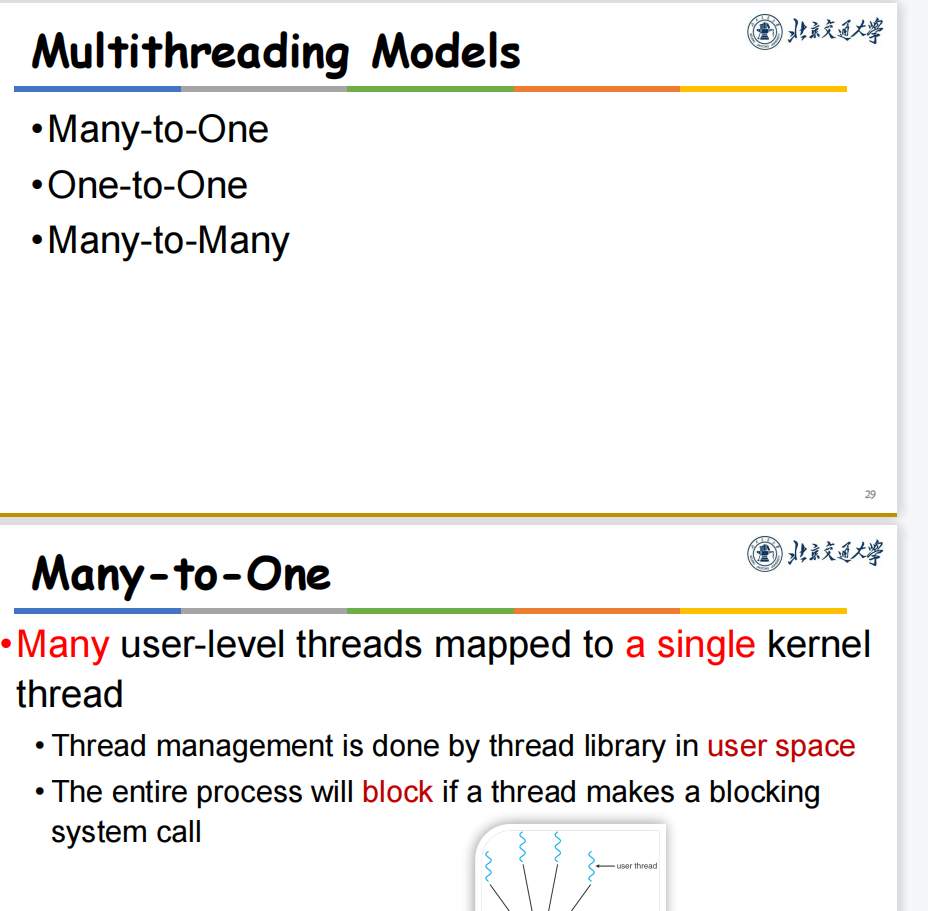
（2） 内核支持线程是OS内核可感知的，而用户级线程是OS内核不可感知的。

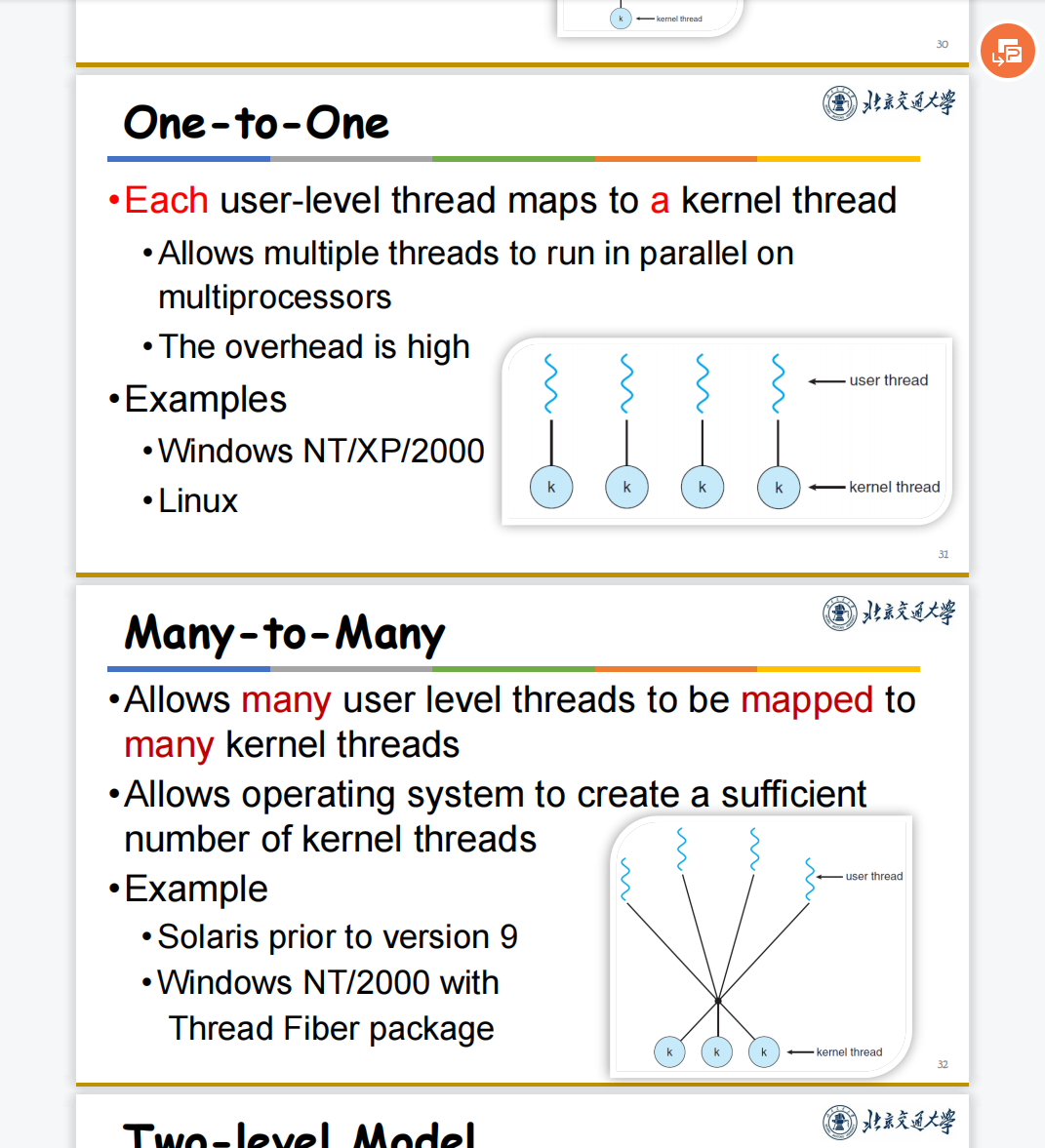
（3） 用户级线程的创建、撤消和调度不需要OS内核的支持。

（4） 用户级线程执行系统调用指令时将导致其所属进程被中断，而内核支持线程执行系统调用指令时，只导致该线程被中断。

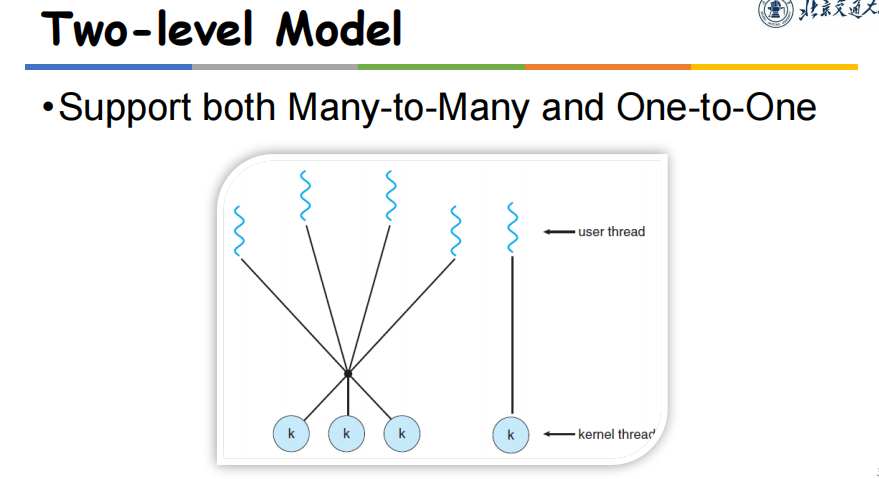
（5） 在只有用户级线程的系统内，CPU调度还是以进程为单位，处于运行状态的进程中的多个线程，由用户程序控制线程的轮换运行；在有内核支持线程的系统内，CPU调度则以线程为单位，由OS的线程调度程序负责线程的调度。

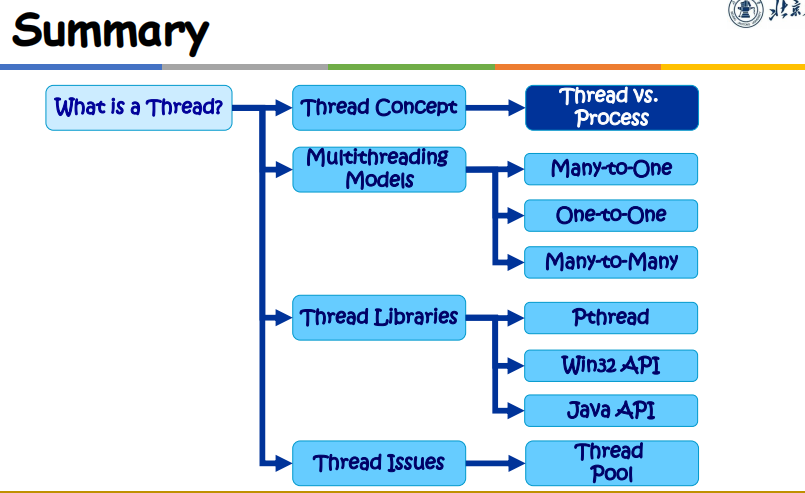
（6） 用户级线程的程序实体是运行在用户态下的程序，而内核支持线程的程序实体则是可以运行在任何状态下的程序。





间接费用： overhead





线程池：

