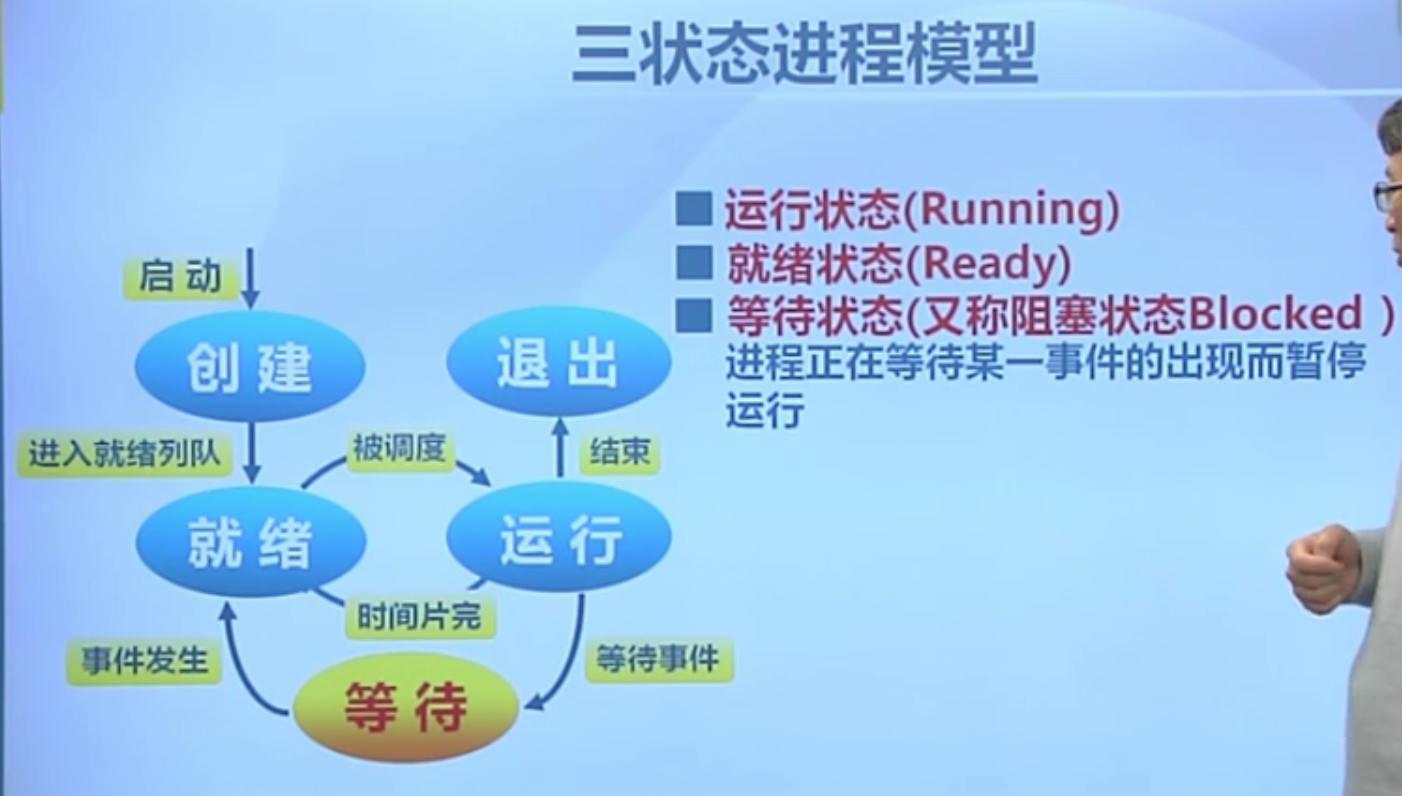
1. 线程的基本概念、线程的基本状态及状态之间的关系？

答：定义：线程，有时称为轻量级进程，是CPU使用的基本单位；由线程ID、程序计数器、寄存器集合和堆栈组成。属于同一进程的其他线程共享其代码段、数据段和其他操作系统资源。

与进程的关系：是进程的执行控制单元，执行路径；一个进程至少有一个线程负责控制程序执行；一个进程如果只有一个线程，这个程序称为单线程；一个进程如果只有多个线程，这个程序称为多线程；

基本状态：运行态、就绪态、阻塞态（synchronize），wait和sleep挂起，结束。

状态间关系：调用线程的start方法后线程进入就绪状态，线程调度系统将就绪状态的线程转为运行状态，遇到synchronized语句时，由运行状态转为阻塞，当synchronized获得锁后，由阻塞转为运行，在这种情况可以调用wait方法转为挂起状态，当线程关联的代码执行完后，线程变为结束状态。



1. 线程与进程的区别？

答：（1）调度：  
        在传统的操作系统中，CPU调度和分派的基本单位是进程。而在引入线程的操作系统中，则把线程作为CPU调度和分派的基本单位，进程则作为资源拥有的基本单位，从而使传统进程的两个属性分开，线程编程轻装运行，这样可以显著地提高系统的并发性。同一进程中线程的切换不会引起进程切换，从而避免了昂贵的系统调用，但是在由一个进程中的线程切换到另一进程中的线程，依然会引起进程切换。  
 （2）并发性：  
      在引入线程的操作系统中，不仅进程之间可以并发执行，而且在一个进程中的多个线程之间也可以并发执行，因而使操作系统具有更好的并发性，从而更有效地提高系统资源和系统的吞吐量。例如，在一个为引入线程的单CPU操作系统中，若仅设置一个文件服务进程，当它由于某种原因被封锁时，便没有其他的文件服务进程来提供服务。在引入线程的操作系统中，可以在一个文件服务进程设置多个服务线程。当第一个线程等待时，文件服务进程中的第二个线程可以继续运行；当第二个线程封锁时，第三个线程可以继续执行，从而显著地提高了文件服务的质量以及系统的吞吐量。  
（3）拥有资源：  
      不论是引入了线程的操作系统，还是传统的操作系统，进程都是拥有系统资源的一个独立单位，他可以拥有自己的资源。一般地说，线程自己不能拥有资源（也有一点必不可少的资源），但它可以访问其隶属进程的资源，亦即一个进程的代码段、数据段以及系统资源（如已打开的文件、I/O设备等），可供同一个进程的其他所有线程共享。  
（4）独立性：  
        在同一进程中的不同线程之间的独立性要比不同进程之间的独立性低得多。这是因为  
为防止进程之间彼此干扰和破坏，每个进程都拥有一个独立的地址空间和其它资源，除了共享全局变量外，不允许其它进程的访问。但是同一进程中的不同线程往往是为了提高并发性以及进行相互之间的合作而创建的，它们共享进程的内存地址空间和资源，如每个线程都可以访问它们所属进程地址空间中的所有地址，如一个线程的堆栈可以被其它线程读、写，甚至完全清除。

（5）系统开销：

       由于在创建或撤销进程时，系统都要为之分配或回收资源，如内存空间、I/O设备等。因此，操作系统为此所付出的开销将显著地大于在创建或撤消线程时的开销。类似的，在进程切换时，涉及到整个当前进程CPU环境的保存环境的设置以及新被调度运行的CPU环境的设置，而线程切换只需保存和设置少量的寄存器的内容，并不涉及存储器管理方面的操作，可见，进程切换的开销也远大于线程切换的开销。此外，由于同一进程中的多个线程具有相同的地址空间，致使他们之间的同步和通信的实现也变得比较容易。在有的系统中，现成的切换、同步、和通信都无需操作系统内核的干预。

（6）支持多处理机系统：  
       在多处理机系统中，对于传统的进程，即单线程进程，不管有多少处理机，该进程只能运行在一个处理机上。但对于多线程进程，就可以将一个进程中的多个线程分配到多个处理机上，使它们并行执行，这无疑将加速进程的完成。因此，现代处理机OS都无一例外地引入了多线程。