

2.1.1 线程与进程的区别：

线程与进程有很多类似的性质，因此人们习惯上也称线程为轻量级进程（lightweight process, LWP），也是 CPU 调度和分派的基本单元；而传统意义上的进程则被称为重量级进程（heavyweight process, HWP），从现代的角度来看，它就是只拥有一个线程的进程。如果进程有多个控制线程，那么它就能同时执行多个任务。他们之间的关系可以简单的由下图表示：

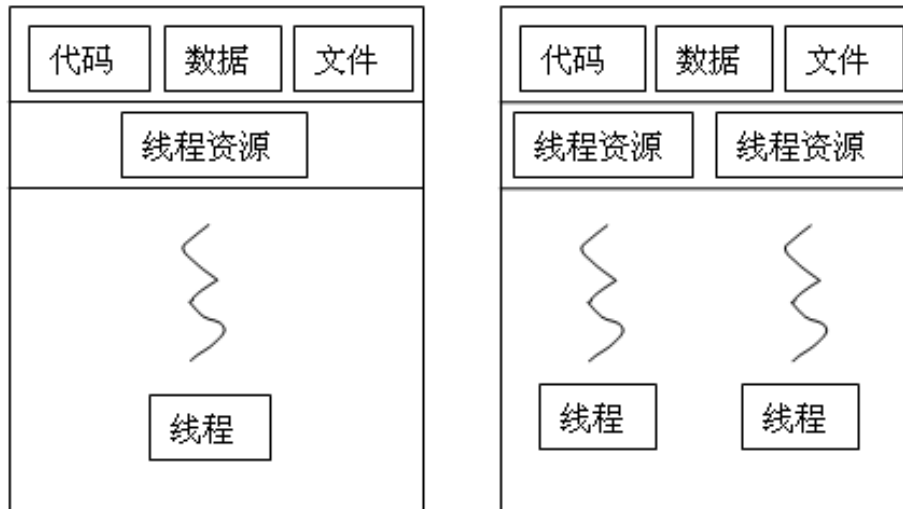


图 2.1 进程与线程的区别

下面，我们主要从调度、并发性、系统开销、拥有资源等方面来对线程和进程进行比较。

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，线程便能轻装运行，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时，依然会引起进程切换。

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，从而能更有效地使用系统资源和提高系统的吞吐量。例如，在一个未引入线程的单 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。在引入了线程的操作系统中，可以在一个文件服务进程中设置多个服务线程。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。

系统开销

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，它可以拥有自己的资源。一般地说，线程自己不拥有系统资源（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），可供同一进程的其他所有线程共享。

拥有资源

由于在创建或撤消进程时，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。类似地，在进行进程切换时，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进程的 CPU 环境的设置。而线程切换只需保存和设置少量寄存器的内容，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，致使它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，线程的切换、同步和通信都无需操作系统内核的干预。