# 第一章 绪论

内核线程是一个能被**独立调度**的执行环境

系统的根本样子和能力是由内核决定

“并发”是代表几个应用程序能够同时处于活动状态并竞争各种资源

“独立”是代表每个应用程序能执行自己的任务，无需考虑其他用户的应用程序在干什么

用户程序的非特权模式和内核的特权模式在Unix下被称为用户态和内核态

进程可以理解为一个程序的实例，也就是说一个正在运行的程序，线程是在一个进程内的可以多个线程

操作系统中有一个名叫调度程序的程序，这个程序决定了哪个进程可以执行哪个不可以

但是有一些操作系统是只允许有非抢占式进程，也就说只有进程自愿放弃CPU的时候，程序调度才会被调用

多用户系统中的进程必须是抢占式的，操作系统记录下每个进程占有的CPU时间，并周期性的激活调度程序

微内核操作系统比单块内核（Unix）更加充分地利用了RAM，因为暂且不需要执行的进程可以直接被调出或者撤销

即使模块依赖于某些特殊的硬件特点，但它不依赖于某个固定的硬件平台

模块代码一旦被连接到内核，其作用与静态链接的内核的目标代码完全等价

如果路径名的第一个字符是斜杠“/”那么这个路径就是绝对路径，它的起点是跟目录

如果第一项是目录名或文件名，那么这个路径就是相对路径，它的起点是进程当前的目录

包含在目录中的文件名就是一个文件的硬链接，简称连接，在同一目录或不同的目录中，同一文件可以有几个连接，因此对应不同文件名

软连接的路径名可以指向位于任意一个文件系统的任意文件或目录，甚至可以指向一个不存在的文件

除了设备文件和特殊文件系统文件外，每个文件都是由字符序列组成，文件内容不包含任何控制信息，每个文件都有自己的**索引节点**，文件系统用索引节点来标识文件