**Linux线程**

**作者：chw**

1. **线程的由来**

**根据前面的学习，我们知道，程序是静态的，进程是动态的。可以这么说，进程是程序一次执行的过程。它是资源管理的最小单位。我们的进程在最初的时候包含了程序、资源和执行三部分。程序指的是我们的代码。资源在操作系统的层面上指的是内存资源，IO资源，信号的出来等待。程序的执行通常理解为执行上下文的切换，这里上下文切换的主要任务是保存老进程CPU状态并加载新进程的保存状态，用新进程的内存映像替换进程的内存映像，包括对CPU的占用。随着时间的发展，我们发现进程之间的切换需要消耗系统很大的开销。**

**它还有两个很大的缺陷：**

**<1>进程只能在一个时间干一件事，如果想同时干两件事或多件事，进程就无能为力了。**

**<2>程在执行的过程中如果阻塞，例如等待输入，整个进程就会挂起，即使进程中有些工作不依赖于输入的数据，也将无法执行。**

**如果你觉得以上的说法太专业的话。我们来举个例子看看。**

**如果把我们上课的过程看成一个进程的话，那么我们要做的是耳朵听老师讲课，手上还要记笔记，脑子还要思考问题，这样才能高效的完成听课的任务。而如果只提供进程这个机制的话，上面这三件事将不能同时执行，同一时间只能做一件事，听的时候就不能记笔记，也不能用脑子思考，这是其一；如果老师在黑板上写演算过程，我们开始记笔记，而老师突然有一步推不下去了，阻塞住了，他在那边思考着，而我们呢，也不能干其他事，即使你想趁此时思考一下刚才没听懂的一个问题都不行，这是其二。**

**现在你应该明白了进程的缺陷了，而解决的办法很简单，我们完全可以让听、写、思三个独立的过程，并行起来，这样很明显可以提高听课的效率。**

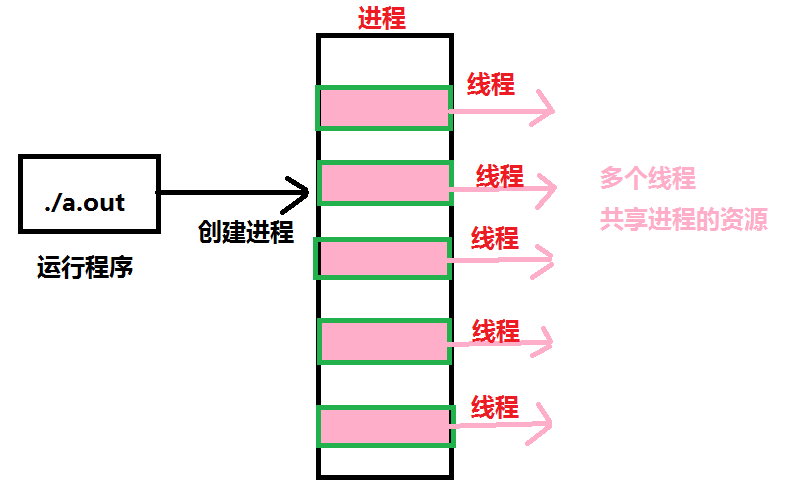
**因此，我们创建了我们的线程。它允许进程将所占有的资源从主体中剥离出来。允许某些线程共享一部分进程的资源。例如，共享进程的文件描述符。线程就是我们轻量的进程。它是程序执行的最小单元。**

1. **线程的概念**

**线程：一个程序的多个执行线路我们称之为线程。它也被称为轻量级别的线程。它是程序执行的最小单元。多个线程共享同一个进程的地址空间，参与系统的同意调度。**

**我们可这个说，一个程序至少有一个进程，一个进程至少有一个线程。既然多个线程共享进程的地址空间。那么这个线程肯定是在进程中创建的。**

**进程、程序、线程之间的关系如下图：**



一个进程只有一个执行主线(线程)，就是main函数。这个我们称为主线程（main pthread）。主线程是进程的执行线程。在多线程程序中。主线程可以创建一个或者多个对等线程(peer thread)。从创建的这个世界点开始，这些线程就是开始并发执行。

**特点：主线程总是第一个运行的线程。对等线程的执行顺序不确定。**

**OK，既然我们的多个线程是共享进程的地址空间。那么多个线程间那些进程的**

**资源是共享，那些资源是独有的呢？**

**答:共享资源：全局变量，进程中打开的文件描述符等，用户ID等。**

**私有资源：线程ID(tid) ，系统相关的数据[函数运行环境]，局部变量，私有栈**

**OK，既然线程是轻量级别的进程。或者说，进程由多个线程组成。那么我们来**

**想一个问题？**

**思考：进程和线程的异同点呢?**

**相同点：1.都参与系统的统一调度。**

1. **都有自己的ID值。（进程pid，线程tid）。**