

认识信号量

在本练习中,您将重做我们之前练习中的 pingpong.c 程序,但是,您将使用 POSIX (匿名)信号量,而不是使用忙等待同步解决方案。这将让我们看看如何使用正确的同步机制可以为我们提供比简单编码的解决方案更有效的解决方案。

我正在为您提供部分实现 pingpong.c,以帮助您入门。它主要只包含一些标题包括一个空的 main() 方法。和以前一样,您需要添加代码来创建两个新线程,让它们交替执行固定数量的迭代,然后让它们在主线程之前与它们连接退出。这一次,我们将使用我们自己的忙等待解决方案让线程轮流使用信号量。下面的代码说明了每个线程应该做什么。

平	乒乓球
<pre>for (int i = 0; i < ITERATIONS; i++) 获取 (pingNext) ;释放 (pongNext) ; }</pre>	<pre>for (int i = 0; i < ITERATIONS; i++) 获取(pongNext);发布 (pingNext) ; }</pre>

上面,我正在使用我们的信号量伪代码。您需要改用 POSIX 信号量 API。您需要将您的两个信号量声明为全局变量,在 main 中初始化它们并在完成后销毁它们。

程序运行后,您可以使用 time shell 命令计时它的执行时间。你应该看到很多执行速度比该程序的先前版本更快,并且您应该看不到单个版本之间的太大差异核心和多核心系统。虽然 pingpong.c 程序没有做任何有用的事情,但它很好地说明了我们希望我们的多线程程序如何工作。通过正确使用操作系统提供的同步机制,我们通常可以获得性能更好的程序,同时为其他程序留出更多的系统资源。

当您的 pingpong.c 程序运行时,将其提交给名为 EX06 的 Gradescope 作业。