Homework:线程与锁

这一部分你将使用哈希表进行多线程编程。在一个真实的pc而不是虚拟机上进行。

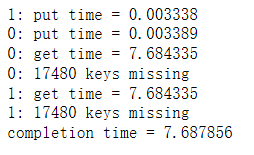
下载ph.c

编译



变量2声明了线程的数量

运行线程一段时间后，会得到下面的输出。



**结果分析：**

每个线程运行两个阶段，第一个阶段每个线程把关键字NKEYS放图哈希表，第二个阶段每个线程从哈希表获取NKEYS。打印的信息告诉你，每个阶段需要运行多久。结果显示每个线程的运算时间大概是7.7s。

如果使用单线程，那么完成时间是7.0s，比单线程少，但是在多线程情况下，进行了两次get。

从而，双线程获得了几乎两倍的速度提升。

当你运行这个应用时，如果你的cpu是单核的，你不会看到并行运行。

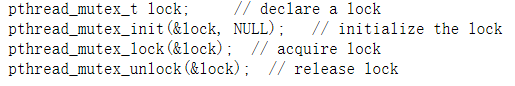
**这里有两点:**

1. 完成时间对于两线程可能的是相同的，但是get运行进行了两次
2. 双线程的输出告诉我们缺失了很多key.但是单线程从来不会缺少key。

为什么多线程会缺失key?而单线程不会缺失？

识别一系列时间可能会导致缺失key

为了避免确实key,在get和put中插入unlock声明，相关的线程调用是（pthread可以查看man）



**任务：**

在单线程测试你的代码，然后用多线程测试，结果是否正确？双线程比单线程更快吗？

修改你的代码，使得get/put并行运行，并能保持正确