**操作系统实习**

Lab3 同步机制 实习说明

本实习希望通过修改Nachos系统平台的底层源代码，达到“扩展同步机制，实现同步互斥实例”的目标。

【实习内容】

**Exercise 1 调研**

调研Linux中实现的同步机制。具体内容见课堂要求。

**Exercise 2 源代码阅读**

阅读下列源代码，理解Nachos现有的同步机制。

* + code/threads/synch.h和code/threads/synch.cc
  + code/threads/synchlist.h和code/threads/synchlist.cc

**Exercise 3 实现锁和条件变量**

可以使用sleep和wakeup两个原语操作（注意屏蔽系统中断），也可以使用Semaphore作为唯一同步原语（不必自己编写开关中断的代码）。

**Exercise 4 实现同步互斥实例**

基于Nachos中的信号量、锁和条件变量，采用两种方式实现同步和互斥机制应用（其中使用条件变量实现同步互斥机制为必选题目）。具体可选择“生产者-消费者问题”、“读者-写者问题”、“哲学家就餐问题”、“睡眠理发师问题”、“养猪场生产问题”等。（也可选择其他经典的同步互斥问题）

**\*Challenge 1 实现barrier**（至少选做一个Challenge）

可以使用Nachos 提供的同步互斥机制（如条件变量）来实现barrier，使得当且仅当若干个线程同时到达某一点时方可继续执行。

**\*Challenge 2 实现read/write lock**

基于Nachos提供的lock(synch.h和synch.cc)，实现read/write lock。使得若干线程可以同时读取某共享数据区内的数据，但是在某一特定的时刻，只有一个线程可以向该共享数据区写入养猪数据。

**\*Challenge 3 研究Linux的kfifo机制是否可以移植到Nachos上作为一个新的同步模块。**