东北大学
工程学院
电气与计算机工程系

EECE7376:操作系统:接口和实现
作业5

指示

- 对于编程问题：

1. 你的代码必须有很好的注释,解释你的程序各行的作用。
   每 4 行代码至少有一个注释。
2. 在源代码文件的开头写下您的全名、学生 ID 和任何
   特殊编译/运行指令（如果有）。
3. 在提交源代码文件之前,您的代码必须使用标准 GCC 编译器（在 CoE Linux 服务器中提供)进行编译和测试。
4. 不要提交任何编译对象或可执行文件。

- 将以下内容提交到 Canvas 上的作业作业页面： 1. 您的作业报告由文字处理程序（例
  如 Microsoft Word)编写并作为一个 PDF 文件提交。报告封面页必须包含您的全名、学生证、课程/部门/学期信息。对于需要绘图的答案,并且如果您很难使用绘图应用程序（这是首选）,您可以“仅”巧妙地手绘这些图表,扫描它们,然后将扫描的图像插入报告文档中。报告内容包括以下内容（取决于作业内容）：

   A。非编程问题的答案,显示了为获得这些答案而遵循的步骤的所有详细信息。

   b.解决编程问题的方法的总结。 C。程序示例运行的屏幕截图

2. 包含注释良好的代码的源文件（即.c、.h 或.S 文件）。

请勿以压缩形式提交任何文件（例如 PDF 报告文件和源代码文件）
包裹。相反,单独上传每个文件。

注意:您可以多次尝试提交此作业,但是,只有您最后一次尝试提交的内容才会被评分。这意味着最后一次提交尝试中必须包含所有必需的文件。

问题1 （50分）

在此问题中,您将检查课程幻灯片中提供的 Michael & Scott 并发队列代码。在您的代码 main 函数中创建一个队列并初始化它。此外,在 main 函数中创建至少两个线程来运行Queue_Enqueue函数,并至少创建另外两个线程来运行Queue_Dequeue函数。

a) 在不使用任何锁的情况下测试您的程序。模拟一系列入队和出队操作。对结果进行评论并强调预计会发生的数据争用问题。

b) 通过使用两个锁（如幻灯片中所示)来比较程序的性能,而不是在入队和出队操作中仅使用一个锁。为了使比较明显,请在临界区中使用系统调用,例如 sleep(1)。模拟所有线程的大量重叠（非顺序)入队和出队操作,并使用系统时间（这是为了简单起见,因为这不是最好的方法)来比较这两种方法的性能。对结果的评论。

c) 如果部署了两个锁,为什么我们需要一个虚拟节点?支持您的回答
不使用虚拟节点测试代码。

将程序附加为三个文件,分别命名为h5-p1-nolock.c、 h5-p1-onelock.c 和h5-p1-twolocks.c

问题2 （50分）

该问题的目标是设计一种多线程归约算法,用于计算向量中所有元素的总和,并提供单个标量值作为结果。实现时请遵循以下准则： a) 使用全局变量sum来累加每个单独线程计算的部分和。声明一个名为 sum_mutex 的全局互斥锁,每个线程在访问变量 sum 之前应锁定该互斥锁。按照课

程幻灯片中的说明初始化全局互斥体。 b) 声明一个名为struct runner_args_t 的数据类型,其中包含创建线程时传递给每个线程的参数集。该结构具有以下字段： · 字段v,表示要处理的整数数组的基地址。 · 字段索引,表示线程将要访问的数组v中元素的索引

开始求和。

· 字段大小,表示一个线程负责的元素数量
总结。

c) 编写一个runner()函数,其中包含每个线程要执行的代码。这
函数应执行以下操作： · 将其类型为void *的通用输入参
数转换为指向*类型结构的指针,并从该结构中获取其输入参数
结构 runner_args_t字段。

·从位置索引开始遍历向量v的大小元素,并计算它们的总和作为局部变量中的部分结果。

· 将部分结果累加到全局变量sum 中。之前锁定互斥体
访问这个全局变量然后解锁它。

· 释放线程的参数结构,因为它应该动态分配为
下面解释。 d) 编写
一个main()程序,创建一个向量,并行计算其总和,并打印
结果。该程序应运行以下中间操作：

·从用户在运行程序时传递的两个参数中读取所需元素的数量(num_elements)和所需线程的数量(num_threads) 。如果程序不
是用两个参数执行的,则打印错误并退出。

· 动态分配具有所需元素数量的向量,并将向量的每个元素初始化为等于其索引的值（即， v[0] = 0、 v[1]
= 1 等）。此初始化保证了可预测和可验证的结果。

· 生成所需数量的线程。为每个线程分配一个新的结构体
runner_args_t动态地,并使用该线程的适当参数对其进行初始化。计算负担应该在线程之间平均分配。

· 等待所有子线程完成。 · 释放矢量并打印结果。

这是程序执行的示例,其中创建了一个包含 10,000 个元素的向量,并使用 5 个并发线程计算其总和：

```
$ ./主要 10000 5
总和 = 49995000
```

使用更多涵盖可能输入的不同场景的示例来测试您的程序。
将完整程序附加到名为h5 -p2.c的文件中