**# 进程/线程的同步**

**## 任务描述**

1. 创建两个线程，在每个线程里对同一个初始值为0的变量进行y次数值+1操作，使两个线程进入一个竞争状态，观察该变量最后的值是否为20万。要求输入从in.txt读入：结果输出到out.txt。（1分）

样例输入和输出：

in.txt:

200000

out.txt:

400000

说明：out.txt的值可能不为400000，请自行判断结果是否正确，下同。

1. 分别使用互斥锁（2分）和Peterson方法（2分）实现解决（1）中的竞争条件问题，并比较两者的性能。在实验过程中可能出现不符合预期的情况，如果不符合预期请给出原因和解决方案。（共4分）

样例输入和输出：

in.txt:

200000

out.txt:

400000

1. 使用对一般信号量的P操作和V操作解决（1）中的临界区冲突问题；在实验过程中观察最终结果是否符合预期，如果不符合预期请进行相关记录并给出原因和解决方案。（2分）

样例输入和输出：

in.txt:

200000

out.txt:

400000

1. 使用信号量实现生产者和消费者问题，要求在生产者和消费者线程之间实现一个有界缓冲区，研究分析两个线程的并发行为。（3分）

样例输入和输出：

in.txt:

3

5

out.txt:

p0

p1

p2

c2

c1

c0

p3

p4

c4

c3

说明：输入包括两行，第一行为缓冲区大小，第二行为顾客的个数；输出中p0表示producer生产者生产了id为0的物品，c0表示consumer消费者消费了id为0的物品，要求代码使用多线程实现，分别编写生产者和消费者线程，一个线程生产/消费一件物品，保证先生产后消费。输出结果不唯一，输出一组符合的顺序即可。

**## 任务说明**

1. 请使用c或c++完成对应程序的开发，对应.c或.cpp均可；
2. 代码命名规范：任务1命名为1-competition.c/.cpp，任务2中的两个代码分别命名为2-mutex.c/.cpp和2-peterson.c/.cpp，任务3命名为3-pv.c/.cpp，任务4命名为4-sem.c/.cpp；如果有代码优化或拓展，请参考命名：2-peterson-optimization.c/.cpp。
3. 提交材料包括实验报告（提交markdown格式，命名为：班级-学号-姓名-操作系统第二次作业.md）、源代码（./code/源代码）和.md所需的图片(非必需，请使用相对路径确保图片正常显示，./image/图片)，要求将全部材料压缩并命令为“班级-学号-姓名-操作系统第二次作业.rar”，实验报告要求内容详实，且包含但不限于实验内容、实验思路、实验源码与注解、实验结果(建议图片+文字说明)、实验总结与反思，源代码要求格式规范且有适当的注释。
4. 对上述实验有拓展内容的同学，可以视情况适当加分，但不超过总分10分。