

# 实验6：并发与锁机制-2023.04.28

---

## Assignment 1 自旋锁与信号量的实现

子任务1—利用自旋锁和信号量实现同步

子任务2—自实现锁机制

## Assignment 2 生产者-消费者问题

总要求与问题场景

子任务1—线程的竞争与冲突

子任务2—利用信号量解决问题

## Assignment 3 哲学家就餐问题

问题场景

子任务1—简单解决方法

子任务2—死锁应对策略

提交要求

## Assignment 1 自旋锁与信号量的实现

### 子任务1—利用自旋锁和信号量实现同步

在实验6中，我们实现了自旋锁和信号量机制。现在，请同学们分别利用指导书中实现的自旋锁和信号量方法，解决实验6指导书中的“消失的芝士汉堡”问题，保存结果截图并说说你的总体思路。**注意：**请将你姓名的英文缩写包含在某个线程的输出信息中（比如，代替母亲或者儿子），用作结果截图中的个人信息表征。

### 子任务2—自实现锁机制

实验6教程中使用了原子指令 `xchg` 来实现自旋锁。但这种方法并不是唯一的。例如，x86指令中提供了另外一个原子指令 `bts` 和 `lock` 前缀等，这些指令也可以用来实现锁机制。现在，同学们需要结合自己所学的知识，实现一个与指导书实现方式不同的锁机制。最后，尝试用你实现的锁机制解决“消失的芝士汉堡”问题，保存结果截图并说说你的总体思路。

# Assignment 2 生产者-消费者问题

## 总要求与问题场景

1. 同学们请在下述问题场景A或问题场景B中**选择一个**，然后在实验6教程的代码环境下创建多个线程来模拟你选择的问题场景。同学们需自行决定每个线程的执行次数，以方便观察临界资源变化为首要原则。
2. 请将你学号的后4位包含在其中一个线程的输出信息中，用作结果截图中的个人信息表征。例如，学号为 21319527 的同学选择问题A，并扮演服务生A，则服务生A放入1块蛋糕时，程序输出 `"Waiter-A 9527 put a piece of matcha cake in the plate."`

- **问题场景A：宴会蛋糕服务**

- **问题描述：**某位商人在餐厅举行生日宴会。餐桌上有一个点心盘，最多可容纳5块蛋糕，每个人（服务生或者来宾）每次只能放入/拿出1块蛋糕。服务生A负责向点心盘中放入抹茶蛋糕，服务生B负责向点心盘中放入芒果蛋糕。生日宴会上一共有6位男性来宾和4位女性来宾，男性来宾等待享用抹茶蛋糕，女性来宾则等待享用芒果蛋糕。如果盘中没有对应口味的蛋糕且点心盘没有放满，来宾会给相应的服务生发送一个请求服务信号，服务生收到信号后会放入1块蛋糕。

- **问题场景B：抽烟者与供应商**

- **问题描述：**酒馆的吧台坐着3位抽烟者和1位供应商。每位抽烟者会不停地卷烟并抽掉，但是要卷起并抽掉一支烟，抽烟者需要三种材料：烟草、纸和胶水。三位抽烟者中，第1位拥有烟草，第2位拥有纸，第3位拥有胶水。供应商会源源不断地提供三种材料，每次他会将两种随机的材料组合放在吧台的桌面上，拥有剩下那种材料的抽烟者，会立刻取走材料卷一根烟并抽掉它，然后给供应商发送一个完成信号，供应商就会放另外两种不同的材料组合在桌面上，这样的过程一直重复。

## 子任务1—线程的竞争与冲突

在子任务1中，要求**不使用任何实现同步/互斥的工具**。因此，不同的线程之间可能会产生竞争/冲突，从而无法得到预期的运行结果。请同学们将线程竞争导致错误的场景呈现出来，保存相应的截图，并描述发生错误的场景。（提示：可通过输出共享变量的值进行观察）

## 子任务2—利用信号量解决问题

针对你选择的问题场景，简单描述该问题中各个线程间的互斥关系，并使用信号量机制实现线程的同步。说说你的实现方法，并保存能够证明你成功实现线程同步的结果截图。

# Assignment 3 哲学家就餐问题

## 问题场景

假设有5位哲学家，他们在午餐时只能思考或吃面。午餐时，这些哲学家共用一个圆桌，每位哲学家都坐在一把指定的椅子上。在桌子上放着5根筷子，每两根筷子之间都放着一碗葱油面。下面是一些约束条件：

- 当一位哲学家处于思考状态时，他对其他哲学家不会产生影响。
- 当一位哲学家感到饥饿时，他会试图拿起与他相邻的两根筷子。
- 一个哲学家一次只能拿起一根筷子，在拿到两根筷子之前不会放下手里的筷子。
- 如果筷子在其他哲学家手里，则需要等待。
- 当一个饥饿的哲学家同时拥有两根筷子时，他会开始吃面。
- 吃完面后的哲学家会同时放下两根筷子，并开始思考。

## 子任务1—简单解决方法

同学们需要在实验6教程的代码环境下，创建多个线程来模拟哲学家就餐的场景。然后，同学们需要结合信号量来实现理论课教材（参见《操作系统概念》中文第9版第187页的内容）中给出的关于哲学家就餐问题的简单解决方法。最后，保存结果截图并说说你是怎么做的

注意：

1. 截图结果中不能出现饥饿/死锁，可以通过输出不同哲学家的状态信息，验证你使用教材的方法确实能解决哲学家就餐问题。
2. 请将你学号的后4位包含在其中一个哲学家线程的输出信息中，用作结果截图中的个人信息表征。

## 子任务2—死锁应对策略

子任务1的解决方案保证两个相邻的哲学家不能同时进食，但是这种方案可能导致死锁。请同学们描述子任务1解决方法中会导致死锁的场景，并将其复现出来。进一步地，请同学们在下述4种策略中选择1种，解决子任务1中的死锁问题，并在代码中实现。最后，保存结果截图并说说你的实现思路。

- **策略1：利用抽屉原理（鸽笼原理）。**即允许最多4个哲学家同时坐在桌子上，保证至少有1位哲学家能吃到面。
- **策略2：利用AND信号量机制或信号量保护机制。**仅当哲学家的左右两支筷子都可用时，才允许他拿起筷子进餐（或者说哲学家必须在临界区内拿起两根筷子）。

- **策略3：使用非对称的解决方案。**即规定奇数号的哲学家先拿起他左边的筷子，然后再去拿他右边的筷子；而偶数号的哲学家则先拿起他右边的筷子，然后再去拿他左边的筷子。按照这一规定，将是1、2号哲学家竞争1号筷子，3、4号哲学家竞争3号筷子。即五个哲学家都竞争奇数号筷子，获得后，再去竞争偶数号筷子，最后总会有一个哲学家能获得两支筷子而进餐。
- **策略4：基于管程的解决方法。**参考理论课教材（见《操作系统概念》中文第9版190–191页），采用类似策略2的思路，定义管程来控制筷子的分布，控制哲学家拿起筷子和放下筷子的顺序，确保两个相邻的哲学家不会同时用餐。

## 提交要求

1. **截止日期：**2023年5月20日00: 00
2. **提交邮箱：**sysu\_os2023@163.com
3. **邮件主题、压缩包、报告文件命名：**实验6–学号–姓名
4. **内容要求：**请大家根据上述要求，完成Assignment 1~Assignment 3，按规范撰写实验报告，并提交.asm汇编源码/C/C++源码。实验结果截图放到实验报告中，截图中尽量包含个人信息要素。
5. **实验6指导书网址：**<https://gitee.com/nelsoncheung/sysu-2023-spring-operating-system/tree/main/lab6>