虚拟场景下的 Linux 并发编程

一个典型的做饭场景可能会包括“多道步骤”，这次作业对这个场景进行模拟 。

场景要求说明

0 多个厨师使用多个资源（比如锅），资源有多有少，每道菜有一定的步骤，其中有些步骤可以并行有些步骤不能并行执行。并且步骤之间具有一定的顺序，执行步骤则需要一定的时间。
1 厨师之间的角色不能完全相同，即不能执行完全相同的步骤。
2 厨师和厨师形成了生产者（顾客生产需求 ）另一个厨师为消费者（厨师使用上一个厨师完成的半成品）的关系，程序应当通过发送信号模拟用户的请求，这个请求应该以一定的策略发送给厨师，在厨师忙时可能需要进行缓存。因为 unix 中存在两个用户定义信号，可以将目标订单的类型定为两类（当然可以通过使用无用信号定义更多的类）。
3 对于资源的访问来说就可能要限制资源的获取数量，可能要使用（锁，信号量）保证步骤的顺序，则要求若干步骤之间要进行同步。
4 而由于角色的差异，厨师之间要进行通信,使用内存共享来完成线程间的通信.
5 模拟用户的需求则通过命令发送信号(kill 命令或者其他)的方式。
6 不能并行的步骤一个人同时只能做一个，可以并行的步骤一个人可以同时完成多个（比如烧水，大部分时间是在等待），这也设计一个同步问题，保证同一个人一个时刻只能在做一个非并行任务。
7 多个不同的角色需要使用不同的进程或是线程来模拟。
8 至少具有两个进程，每个进程至少联系两个线程。每个线程具有明确的逻辑语义，即联系固定的角色。
9 每个菜品（产品），流程至少具有两个，并且一定具有所需要的**资源**，比如某种资源只有两份，但是厨师有 4 个，这就要求使用上述设计解决这个问题。
10 每个小组成员至少设计其中一个角色线程的行为。
11 可以使用 socket 进行通讯。

作业要求

规则

0 设计一个做饭的场景（如上述场景描述），添加自己的设计细节
1 完成代码逻辑。
2 提交代码和设计文档，设计文档包含场景的详细设计和如何实现场景，两个部分。
3 由组长提交。
4 提交时间 2019 年 6 月 28 日 23:59:59 之前，尽量不要超时，已经到烤漆之后了，很可能

你补交的时候成绩已经给出了。

建议

0 主进程接受用户输入，并生成多个进程。
1 每个进程可有多个线程
2 进程之间通过 IPC 通信
3 线程之间通过互斥量、条件变量同步或互斥。
4 unix 高级编程书中提供的 apue.h 可用于支持编程（非必需）
5 有任何问题向助教提问。

评分规则

0 不要求场景过度复杂，
1 在自拟的需求内高效完成即可。
2 尽量使用到所有的要素，
3 尽可能完善错误处理，
4 尽可能使得程序高效率的完成任务，
5 最主要的是保证虚拟场景逻辑的正确性。

样例

提供了一份简单的样例，样例中使用了简单的**同步手段**保证逻辑正确性，也没有**错误处理**，没有资源的获取等等 ，单一的厨师进程同时并行的完成了两项任务， 并不满足本次实验要求，仅参考。