作业 7

7.1 设有两个优先级相同的进程 T1，T2 如下。令信号量 S1，S2 的初值为 0，已知 z=2，试 问 T1，T2 并发运行结束后 x=?y=?z=?

线程 T1

y:=1;

y:=y+2;

V(S1);

z:=y+1;

P(S2);

y:=z+y;

线程 T2

x:=1;

x:=x+1;

P(S1);

x:=x+y;

V(S2);

z:=x+z;

注:请分析所有可能的情况，并给出结果与相应执行顺序。

7.2 在生产者-消费者问题中，假设缓冲区大小为 5，生产者和消费者在写入和读取数据时都会更新写入/读取的位置 offset。现有以下两种基于信号量的实现方法，

方法一

```
Class BoundedBuffer {
    mutex = new Semaphore(1);
    fullBuffers = new Semaphore(0);
    emptyBuffers = new Semaphore(n);
}
BoundedBuffer::Deposit(c) {
    emptyBuffers->P();
```

```
    mutex->P();
    Add c to the buffer;
    offset++;
    mutex->V();
    fullBuffers->V();
}
BoundedBuffer::Remove(c) {
    fullBuffers->P();
    mutex->P();
    Remove c from buffer;
    offset--;
    mutex->V();
    emptyBuffers->V();
}
```

方法二：

```
Class BoundedBuffer {
    mutex = new Semaphore(1);
    fullBuffers = new Semaphore(0);
    emptyBuffers = new Semaphore(n);
}
BoundedBuffer::Deposit(c) {
    mutex->P();
    emptyBuffers->P();
    Add c to the buffer;
    offset++;
    fullBuffers->V();
    mutex->V();
}
BoundedBuffer::Remove(c) {
```

```
    mutex->P();
    fullBuffers->P();
    Remove c from buffer;
    offset--;
    emptyBuffers->V();
    mutex->V();
}
```

请对比分析上述方法一和方法二，哪种方法能让生产者、消费者进程正常运行，并说明分析原因。

7.3 银行有 n 个柜员,每个顾客进入银行后先取一个号,并且等着叫号,当一个柜员空闲后,就叫下一个号.

请使用 PV 操作分别实现:

//顾客取号操作 Customer_Service

//柜员服务操作 Teller_Service

7.4 多个线程的规约(Reduce)操作是把每个线程的结果按照某种运算(符合交换律和结合律) 两两合并直到得到最终结果的过程。

试设计管程 monitor 实现一个 8 线程规约的过程，随机初始化 16 个整数，每个线程通过调用 monitor.getTask 获得 2 个数，相加后，返回一个数 monitor.putResult ，然后再 getTask( ) 直到全部完成退出，最后打印归约过程和结果。

要求：为了模拟不均衡性，每个加法操作要加上随机的时间扰动，变动区间 1~10ms。

提示：使用 pthread_系列的 cond_wait，cond_signal，mutex 实现管程；使用 rand( ) 函数产生随机数，和随机执行时间。