

OSLAB4

191830064 姜纪文

exercise1: 请回答一下，什么情况下会出现死锁。

例如哲学家都去拿左手边的或者右手边的叉子，就会发现另一只手所需的叉子被别人拿了，于是互相等待，产生死锁。

exercise2: 说一下该方案有什么不足？（答出一点即可）

一次只能一个哲学家能吃饭，其他哲学家要等这一个哲学家吃完才能去尝试，并发程度低。

exercise3: 正确且高效的解法有很多，请你利用信号量 PV 操作设计一种正确且相对高效（比方案 2 高效）的哲学家吃饭算法。

Solution1: 只允许最多 4 个人开始吃饭。

Solution2: 用一个互斥信号量来保护同时请求两边的筷子的操作

Solution3: 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。下面为 solution3 伪代码。

```
semaphore chopstick[5]={1,1,1,1,1};
void philosopher(int i)
{
    while(true)
    {
        think();
        if(i%2 == 0) //偶数哲学家，先右后左。
        {
            wait (chopstick[(i + 1)%5]);
            wait (chopstick[i]);
            eat();
            signal (chopstick[(i + 1)%5]);
            signal (chopstick[i]);
        }
        else //奇数哲学家，先左后右。
        {
            wait (chopstick[i]);
            wait (chopstick[(i + 1)%5]);
            eat();
            signal (chopstick[i]);
            signal (chopstick[(i + 1)%5]);
        }
    }
}
```

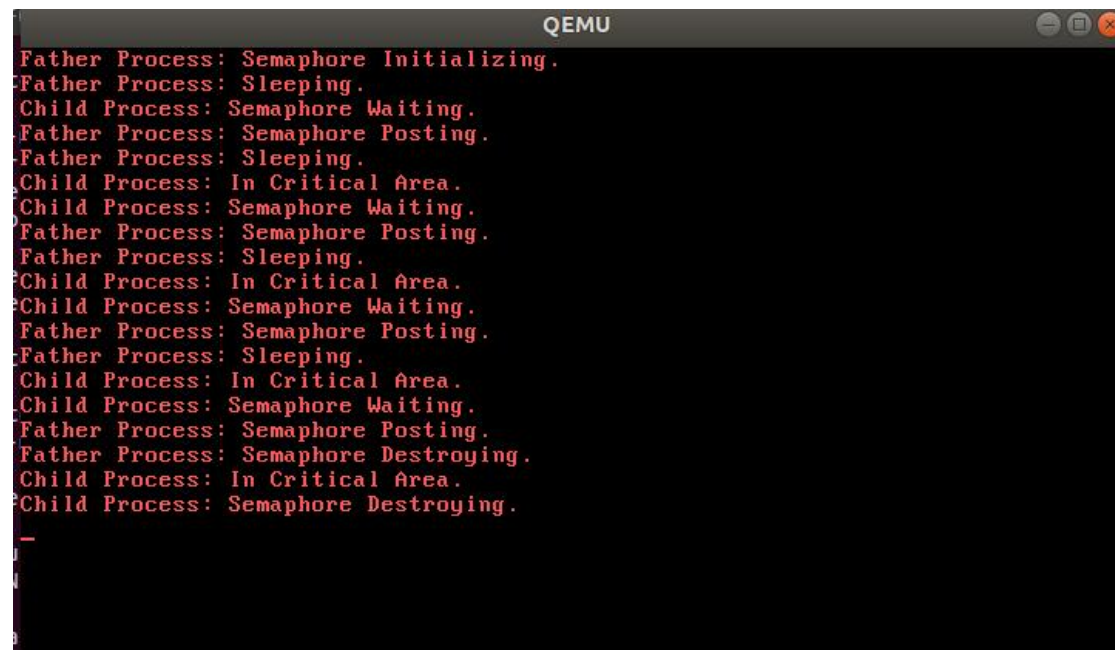
exercise4: 为什么要用两个信号量呢? emptyBuffers 和 fullBuffer 分别有什么直观含义?

因为缓冲区对于生产者和消费者的限制是不同的,也就是二者所需求的资源是不同的。生产者关注是否有空位,消费者关注是否有产品,emptyBuffers 对应于生产者的空位资源,fullBuffers 对应于消费者的产品资源。

scanf 测试

有 bug, de 不出来。

信号量测试

A screenshot of a QEMU terminal window with a black background and red text. The text shows the execution of a semaphore test involving a 'Father Process' and a 'Child Process'. The sequence of events is: Father Process initializes and sleeps; Child Process waits; Father Process posts and sleeps; Child Process enters critical area; Child Process waits; Father Process posts and sleeps; Child Process enters critical area; Child Process waits; Father Process posts and sleeps; Child Process enters critical area; Child Process waits; Father Process posts and destroys semaphore; Child Process enters critical area; Child Process destroys semaphore. The terminal window has a title bar with 'QEMU' and standard window controls.

```
QEMU
Father Process: Semaphore Initializing.
Father Process: Sleeping.
Child Process: Semaphore Waiting.
Father Process: Semaphore Posting.
Father Process: Sleeping.
Child Process: In Critical Area.
Child Process: Semaphore Waiting.
Father Process: Semaphore Posting.
Father Process: Sleeping.
Child Process: In Critical Area.
Child Process: Semaphore Waiting.
Father Process: Semaphore Posting.
Father Process: Sleeping.
Child Process: In Critical Area.
Child Process: Semaphore Waiting.
Father Process: Semaphore Posting.
Father Process: Semaphore Destroying.
Child Process: In Critical Area.
Child Process: Semaphore Destroying.
```

哲学家测试

```
QEMU
Father Process: Semaphore Initializing.
Father Process: Sleeping.
Child Process: Semaphore Waiting.
Father Process: Semaphore Posting.
Father Process: Sleeping.
Child Process: In Critical Area.
Child Process: Semaphore Waiting.
Father Process: Semaphore Posting.
Father Process: Sleeping.
Child Process: In Critical Area.
Child Process: Semaphore Waiting.
Father Process: Semaphore Posting.
Father Process: Sleeping.
Child Process: In Critical Area.
Child Process: Semaphore Waiting.
Father Process: Semaphore Posting.
Father Process: Semaphore Destroying.
Child Process: In Critical Area.
Child Process: Semaphore Destroying.
```