



东北师范大学

2020 年春季学期操作系统第一次作业

2018 级计算机科学与技术（中美合作）

包亦航
2018011890

2020 年 4 月

问题一：

N 个共享缓冲区读写问题，我们现在有 N 个共享缓冲区，一个写进程，一个打印进程，需要我们设计代码来实现这个过程

解答方案一（使用上课讲的新增临界资源的方法）：

首先判断这是个同步互斥问题，有两个进程：一个写进程，一个打印进程

我们使用 process 变量来记录缓冲区中目前有的进程数，mutex 用来保护该变量

```
Int process=0;
```

```
Semaphore mutex=1;
```

```
Write(){
```

```
    While(1){
```

```
        P(mutex);
```

```
        If (process<N){
```

```
            Process = Process+1;
```

```
            Write data into the memory
```

```
            V(mutex);
```

```
        }
```

```
        Else{
```

```
            V(mutex);
```

```
        }
```

```
    }
```

```
}
```

```
Print(){
```

```
    While(1){
```

```
        P(mutex);
```

```
        If (process>0){
```

```
            Process = Process-1;
```

```
            Print the data out
```

```
            V(mutex);
```

```
        }
```

```
        Else{
```

```
            V(mutex);
```

```
        }
```

```
    }
```

```
}
```

```
Int main()
```

```
{
```

```
    Process = 0;
```

```
    Mutex = 1;
```

```
}
```

解答方案二（使用纯信号量方法）：

我们使用 use 信号量记录缓冲区是否被占用，用 available 记录还有多少剩余空间，用 record 记录目前有几个被占用

Semaphore use=1; available=N; record=0;

```
Write(){
    While(1){
        Prepare data
        P(available); P(use);
        Write data into the memory
        V(record);V(use);
    }
}
```

```
Print(){
    While(1){
        P(record); P(use);
        read data from memory
        V(available);V(use);
        Print data
    }
}
```

```
Int main()
{
    use=1;
    available=N;
    record=0;
}
```

问题二：桌上有个能盛下 N 个水果的空盘子。父亲不停地向盘中放苹果或桔子，儿子不停地从盘中取出桔子，女儿不停地从盘中取出苹果。规定 3 人不能同时从盘中取放水果。试用信号量实现 3 人代表的进程之间的同步。

解答方案：

这是一个同步互斥问题

我们这里的信号量设置方式如下：设置一个 plate 信号量，用来标记这个盘子是否被占用；设置一个 cap 信号量，用来标记盘子剩余的空间；设置一个 apple 信号量，用来标记盘子里有几个苹果；设置一个 orange 信号量，用来标记盘子里有几个橘子
代码如下：

Semaphore plate=1, cap=N, apple=0, orange=0;

Dad(){

```

        While(1){
            Prepare an apple;
            P(cap); P(plate);
            Put the apple on the plate;
            V(apple); V(plate);
            Prepare an orange;
            P(cap); P(plate);
            Put the orange on the plate;
            V(orange); V(plate);
        }
    }
    Son(){
        While(1){
            P(orange); P(plate)
            Take an orange from the table;
            V(cap);V(plate);
            Eat the orange;
        }
    }
    daughter(){
        While(1){
            P(apple); P(plate)
            Take an apple from the table;
            V(cap);V(plate);
            Eat the apple;
        }
    }
    Int main()
    {
        plate=1;
        cap=N;
        apple=0;
        orange=0;
    }

```

问题三：利用管程解决多读者多写者问题

解答方案：

```

Monitor reader_writer(){
    int reader_num=0,
    bool writer=false;
    condition read_q, write_q;

    void start_read{
        if (writer) read_q.wait();
        reader_num ++;
    }

```

```

        if (read_q.queue) read_q.signal();
    }

Void end_read(){
    reader_num--;
    if (reader_num==0) write_q.signal();
}

Void start_write(){
    If (reader_num>0 || writer) write_q.wait();
    writer = true
}

Void end_write(){
    writer = false;
    if (read_q.queue) read_q.signal();
    else if (write_q.queue) write_q.signal();
}
Int main()
{
    reader_num=0,
    writer=false;
}

```