



软件设计师

--操作系统基础知识

高级项目经理 任铄

QQ: 1530841586

第三章 操作系统基础知识

- 3.1 操作系统功能、类型和层次结构
- 3.2 处理机管理（进程管理）
- 3.3 存储管理
- 3.4 设备管理
- 3.5 文件管理
- 3.6 作业管理
- 3.7 嵌入式操作系统

二、进程互斥与同步

1. 进程间两种形式的制约关系

- (1) 间接相互制约关系 --- 源于资源共享
- (2) 直接相互制约关系 --- 源于进程合作

高校教师 任铄
QQ: 1530841586

向上人生路!

2. 临界资源

- 临界资源 (Critical Resource) :把一段时间内只允许一个进程访问的资源称为临界资源或独占资源
- 临界区 (Critical Section) :每个进程中访问临界资源的那段代码称为临界区

三、信号量机制

- 信号量是OS提供的管理公共资源的有效手段。
- 信号量是一个整数，当信号量**大于等于零时，代表可供并发进程使用的资源数量**，当信号量**小于零时**，表示处于阻塞态进程的个数。

高校教师 任铄

QQ: 1530841586

Wait 操作：

- 申请资源，减量操作， $S.value := S.value - 1$
- 当 $S.value < 0$ 时，表示资源分配完，进行自我阻塞。

Signal操作：

- 释放资源，增量操作， $S.value := S.value + 1$
- 当 $S.value \leq 0$ ，唤醒S.L链表中的等待进程。

四、信号量的应用

1. 利用信号量实现进程互斥（模式）
2. 利用信号量实现前驱关系（模式）
3. 利用记录型信号量实现同步（模式）

高校教师、高级项目经理 任铎

QQ: 1530841586

向上人生路!

1.利用信号量实现进程互斥（模式）

为使多个进程互斥的访问某临界资源，须为该资源设置一互斥信号量mutex，并设其初始值为1，然后将各进程访问资源的临界区CS置于wait(mutex)和signal(mutex)之间即可。


```
Var mutex: semaphore :=1;  
begin  
  parbegin  
    process 1: begin  
      repeat  
        wait(mutex);  
        critical section  
        signal(mutex);  
        remainder section  
      until false;  
    end  
  parend
```

wait(mutex)和signal(mutex)
必须成对出现

```
process 2: begin  
  repeat  
    wait(mutex);  
    critical section  
    signal(mutex);  
    remainder section  
  until false;  
end
```

```
Var mutex: semaphore :=1; //表示打印机
```

```
begin
```

```
parbegin
```

```
  p1: begin
```

```
    repeat
```

```
      ... ..
```

```
      wait(mutex);
```

```
      使用打印机
```

```
      signal(mutex);
```

```
      ... ..
```

```
    until false;
```

```
  end
```

```
parend
```

例：用记录型信号量实现两个进程互斥使用一台打印机

练习：用记录型信号量实现三个进程互斥使用一台打印机

```
  p2: begin
```

```
    repeat
```

```
      ... ..
```

```
      wait(mutex);
```

```
      使用打印机
```

```
      signal(mutex);
```

```
      ... ..
```

```
    until false;
```

```
  end
```

```
  p3: begin
```

```
    repeat
```

```
      ... ..
```

```
      wait(mutex);
```

```
      使用打印机
```

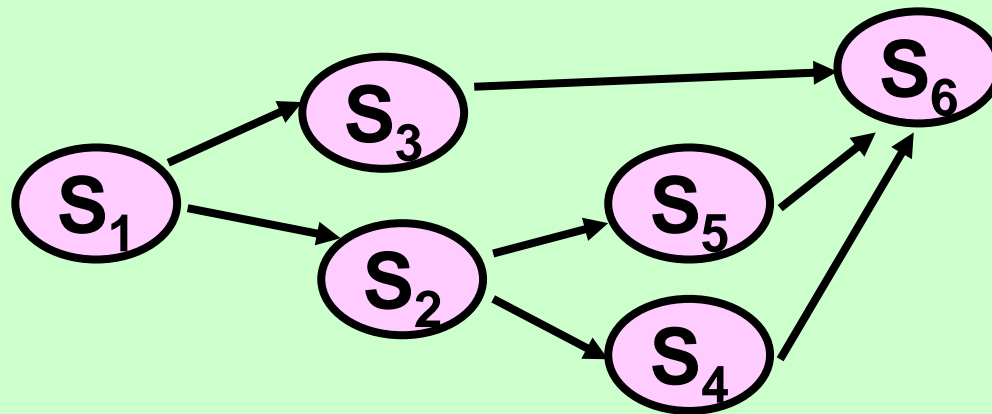
```
      signal(mutex);
```

```
      ... ..
```

```
    until false;
```

```
  end
```

2.利用信号量实现前驱关系 (模式)



```
Var a, b, c, d, e, f, g; semaphore :=0, 0, 0, 0, 0, 0, 0;  
begin  
  parbegin  
    begin S1; signal(a); signal(b); end;  
    begin wait(a); S2; signal(c); signal(d); end;  
    begin wait(b); S3; signal(e); end;  
    begin wait(c); S4; signal(f); end;  
    begin wait(d); S5; signal(g); end;  
    begin wait(e); wait(f); wait(g); S6; end;  
  parend  
end
```

S2

3.利用记录型信号量实现同步（模式）

P1 , p2两进程因合作完成一项任务而共用一个变量x。
进程p2将处理结果送入x；进程P1将x的结果打印。

即：p2：x=处理结果；

p1：Print(x)；

如何实现该合作关系？

高校教师、高级项目经理 任铄

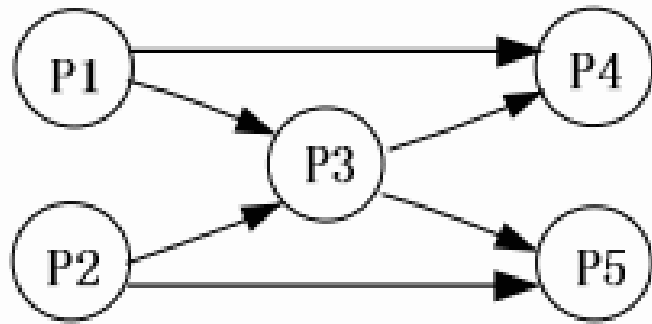
QQ: 1530841586

```
Var empty: semaphore :=1; //变量x可赋值使用，即P1  
                                的print(x)已完成  
full: semaphore :=0; //变量x已赋值，即P2可  
                                print(x)
```

```
begin  
parbegin  
  p1: begin  
    repeat  
      ... ..  
      wait(full);  
      print(x);  
      signal(empty);  
      ... ..  
    until false;  
  end  
parend
```

```
p2: begin  
  repeat  
    ... ..  
    wait(empty);  
    x:=处理结果;  
    signal(full);  
    ... ..  
  until false;  
end
```

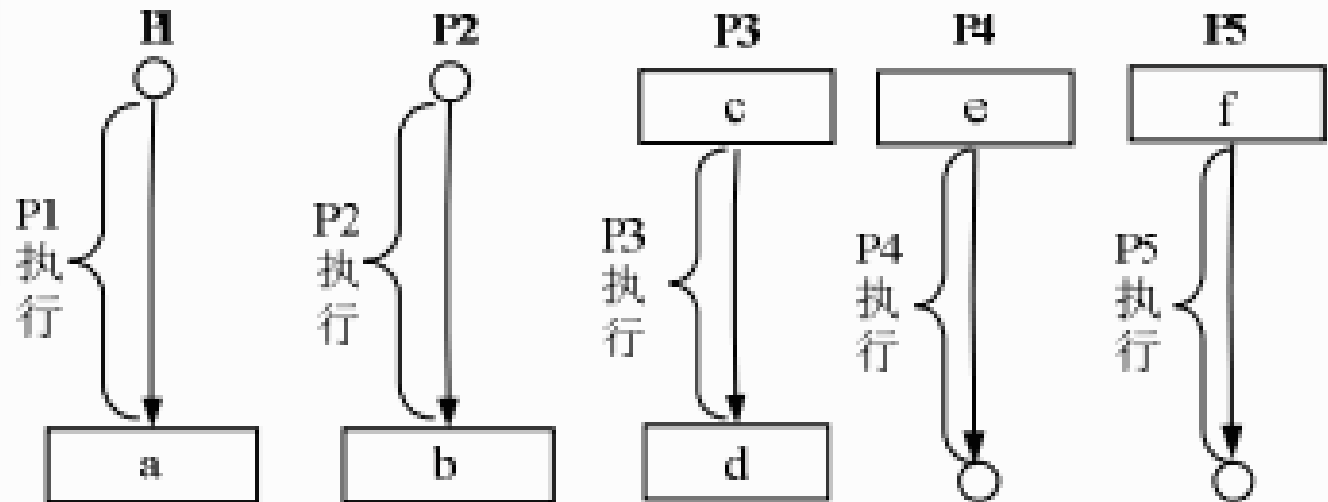
进程P1、P2、P3、P4和P5的前趋图如下，若用PV操作控制进程P1~P5并发执行的过程，则需要设置6个信号量S1、S2、S3、S4、S5和S6，且信号量S1~S6的初值都等于零。下图中a和b处应分别填写（C）；c和d处应分别填写（B），e和f处应分别填写（C）。



高校教师 任钰

QQ: 1530841586

向上人生路!



(23

- A. $P(S1) P(S2)$ 和 $P(S3) P(S4)$ B. $P(S1) V(S2)$ 和 $P(S2) V(S1)$
C. $V(S1) V(S2)$ 和 $V(S3) V(S4)$ D. $P(S1) P(S2)$ 和 $V(S1) V(S2)$

(24

- A. $P(S1) P(S2)$ 和 $V(S3) V(S4)$ B. $P(S1) P(S3)$ 和 $V(S5) V(S6)$
C. $V(S1) V(S2)$ 和 $P(S3) P(S4)$ D. $P(S1) V(S3)$ 和 $P(S2) V(S4)$

(25

- A. $P(S3) P(S4)$ 和 $V(S5) V(S6)$ B. $V(S5) V(S6)$ 和 $P(S5) P(S6)$
C. $P(S2) P(S5)$ 和 $P(S4) P(S6)$ D. $P(S4) V(S5)$ 和 $P(S5) V(S6)$

高校教师、高级项目经理 任铄

QQ: 1530841586

向上人生路!

可以通过下列渠道沟通联系：

- 1、QQ:1530841586
- 2、QQ群：164955673

向上人生路！