

第二章 进程同步

一、选择最合适的答案

1. 用 `wait()`、`signal()` 操作管理临界区 (1 个实例的临界资源) 时, 信号量的初值一般应定义为 (C)。
A. -1 B. 0 C. 1 D. 任意值
2. 有 m 个进程共享同一临界资源 (1 个实例), 若使用信号量机制实现对一临界资源的互斥访问, 则信号量的变化范围是 (A)。
A. 1 至 $-(m-1)$ B. 1 至 $m-1$ C. 1 至 $-m$ D. 1 至 m
3. 在下面的叙述中, 正确的是 (C)。
A. 临界资源是非共享资源 B. 临界资源是任意共享资源
C. 临界资源是互斥共享资源 D. 临界资源是同时共享资源
4. 对进程间互斥地使用临界资源, 进程可以 (D)。
A. 互斥地进入临界区 B. 互斥地进入各自的临界区
C. 互斥地进入同一临界区 D. 互斥地进入各自的同类资源的临界区
5. 设两个进程共用一个临界资源的互斥信号量 `mutex`, 当 `mutex=1` 时表示 (B)。
A. 一个进程进入了临界区, 另一个进程等待 B. 没有一个进程进入临界区
C. 两个进程都进入了临界区 D. 两个进程都在等待
6. 设两个进程共用一个临界资源的互斥信号量 `mutex`, 当 `mutex=-1` 时表示 (A)。
A. 一个进程进入了临界区, 另一个进程等待 B. 没有一个进程进入临界区
C. 两个进程都进入了临界区 D. 两个进程都在等待
7. 当一进程因在信号量 S 上执行 `wait(S)` 操作而被阻塞后, S 的值为 (B)。
A. >0 B. <0 C. ≥ 0 D. ≤ 0
8. 当一进程因在信号量 S 上执行 `signal(S)` 操作而导致唤醒另一进程后, S 的值为 (D)。
A. >0 B. <0 C. ≥ 0 D. ≤ 0
9. 如果信号量的当前值为 -4, 则表示系统中在该信号量上有 (A) 个进程等待。
A. 4 B. 3 C. 5 D. 0
10. 若有 4 个进程共享同一程序段, 而且每次最多允许 3 个进程进入该程序段, 则信号量的变化范围是 (B)。
A. 3, 2, 1, 0 B. 3, 2, 1, 0, -1
C. 4, 3, 2, 1, 0 D. 2, 1, 0, -1, -2
11. 若信号 S 的初值为 2, 当前值为 -1, 则表示有 (B) 个等待进程?
A. 0 B. 1 C. 2 D. 3
12. 如果有三个进程共享同一互斥段, 而且每次最多允许两个进程进入该互斥段, 则信号量的初值应设置为 (C)。
A. 3 B. 1 C. 2 D. 0
13. 并发进程之间 (D)
A. 彼此无关 B. 必须同步 C. 必须互斥 D. 可能需要同步或互斥

14. 在操作系统中, 有一组进程, 进程之间具有直接相互制约性。这组并发进程之间(B)。
- A. 必定无关 B. 必定相关 C. 可能相关 D. 相关程度相同
15. (A) 操作不是 wait() 操作可完成的。
- A. 为进程分配处理机 B. 使信号量的值变小
C. 可用于进程的同步 D. 使进程进入阻塞状态

二、选择所有正确的答案

1. 有关进程的描述中, (AC) 是正确的。
- A. 进程执行的相对速度不能由进程自己来控制
B. 利用信号量的 wait(). signal() 操作可以交换大量信息
C. 同步是指并发进程之间存在的一种制约关系
D. 并发进程在访问共享资源时, 不可能出现与时间有关的错误
2. 下列资源中, (ACD) 是临界资源。
- A. 打印机 B. 非共享的资源 C. 共享变量 D. 共享缓冲区
3. 进程从执行状态转换到阻塞状态的可能原因是(BD)。
- A. 时间片完 B. 需要等待其它进程的执行结果
C. 执行了 signal() 操作 D. 执行了 wait() 操作
4. 进程从阻塞状态转换到就绪状态的可能原因是(BC)。
- A. 时间片完 B. 其它进程执行了唤醒原语
C. 执行了 signal() 操作 D. 执行了 wait() 操作
5. 在单处理机系统中, 设系统中有 n 个进程 ($n > 2$), 且当前处理机没有执行进程调度程序, 下述情况哪些可能发生 (BCD)。
- A. 没有运行的进程, 有 2 个进程处于就绪状态, n 个进程处于等待状态。
B. 一个进程处于运行状态, $n-1$ 个进程处于等待状态。
C. 一个进程处于运行状态, 1 个进程处于就绪状态, $n-2$ 个进程处于等待状态。
D. 一个进程处于运行状态, $n-1$ 个进程处于就绪状态, 没有进程处于等待状态

三、判断正误, 错误的简要说明理由

1. 一个临界资源可以对应多个临界区。(√)
2. 互斥地使用临界资源是通过互斥地进入临界区实现的。(X)
表达不确切, 应该是互斥的进入同类临界区。
3. 同步信号量的初值一般为 1。(X)
互斥信号量的初值一般为 1; 而同步信号量的初值应视具体情况而定。
4. 生产者—消费者问题是一个既有同步又有互斥的问题。(√)
5. 进程 A、B 共享变量 x , 需要互斥执行; 进程 B、C 共享变量 y , B、C 也需要互斥执行, 因此, 进程 A、C 必须互斥执行。(X)
不具有传递性。
6. 单道程序系统中程序的执行也需要同步和互斥。(X)
单道程序系统不具有并发性, 因此不需要同步和互斥。

四、解答题

1. 某车站售票厅，任何时刻最多可容纳 20 名购票者进入，当售票厅中少于 20 购票者时，则厅外的购票者可立即进入，否则需在外面等待。若把一个购票者看作一个进程，请回答下列问题：

(1) 用 wait()、signal()操作管理这些并发进程时，应怎样定义信号量？写出信号量的初值以及信号量各种取值的含义。

(2) 根据所定义的信号量，把应执行的 wait()、signal()操作填入下述程序中，以保证进程能够正确地并发执行。

COBEGIN PROCESS P_i ($i=1,2,\dots$)

 Begin

;

 进入售票厅 ;

 购票;

;

 退出;

 End;

COEND

(3) 若欲购票者最多为 n 个人，写出信号量可能的变化范围（最大值和最小值）。

解：

售票厅问题：

(1) 定义一信号量 S ，初始值为 20。

$S > 0$ S 的值表示可继续进入售票厅的人数

$S = 0$ 表示售票厅中已有 20 名顾客

$S < 0$ $|S|$ 的值为等待进入售票厅中的人数

(2) 上框为 $P(S)$

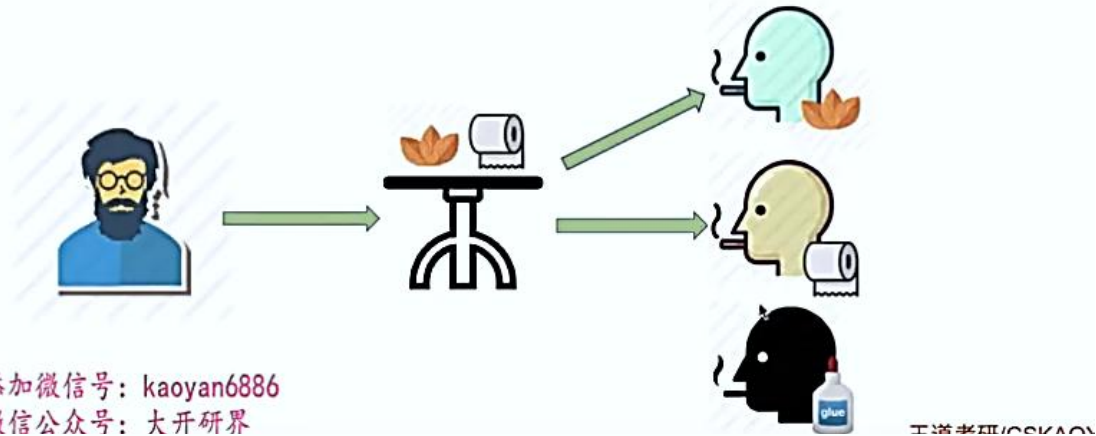
 下框为 $V(S)$

(3) S 的最大值为 20

S 的最小值为 $20-N$ ， N 为某一时刻需要进入售票厅的最大人数。

问题描述

假设一个系统有三个抽烟者进程和一个供应者进程。每个抽烟者不停地卷烟并抽掉它，但是要卷起并抽掉一支烟，抽烟者需要有三种材料：烟草、纸和胶水。三个抽烟者中，第一个拥有烟草、第二个拥有纸、第三个拥有胶水。供应者进程无限地提供三种材料，供应者每次将两种材料放桌子上，拥有剩下那种材料的抽烟者卷一根烟并抽掉它，并给供应者进程一个信号告诉完成了，供应者就会放另外两种材料再桌上，这个过程一直重复（让三个抽烟者轮流地抽烟）

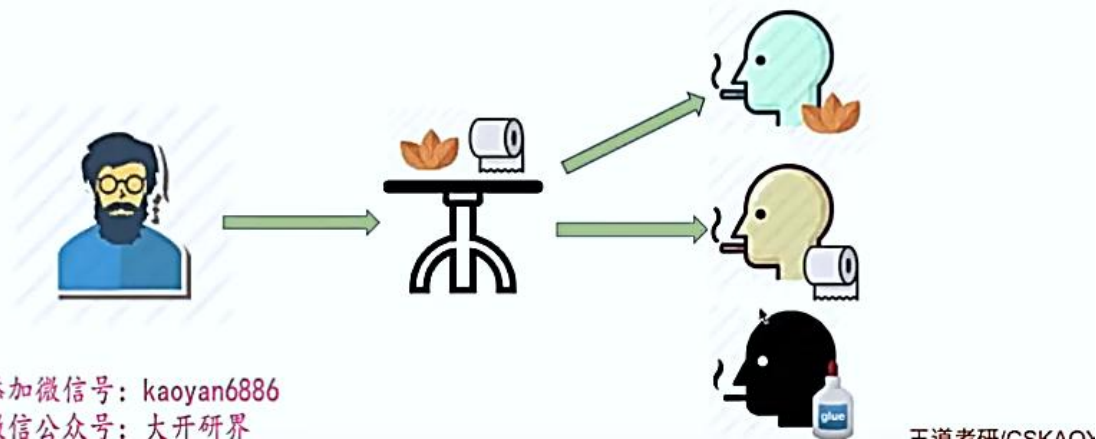


添加微信号: kaoyan6886

微信公众号: 大开研界

问题描述

假设一个系统有三个抽烟者进程和一个供应者进程。每个抽烟者不停地卷烟并抽掉它，但是要卷起并抽掉一支烟，抽烟者需要有三种材料：烟草、纸和胶水。三个抽烟者中，第一个拥有烟草、第二个拥有纸、第三个拥有胶水。供应者进程无限地提供三种材料，供应者每次将两种材料放桌子上，拥有剩下那种材料的抽烟者卷一根烟并抽掉它，并给供应者进程一个信号告诉完成了，供应者就会放另外两种材料再桌上，这个过程一直重复（让三个抽烟者轮流地抽烟）



添加微信号: kaoyan6886

微信公众号: 大开研界

```
Semaphore S1=0;  
Semaphore S1=0;  
Semaphore S1=0;  
Semaphore table=1;
```

```
Smoker1(){  
    while(true){  
        wait(S1);  
        从桌上拿走组合 1;  
        卷烟、抽掉;  
        signal(table);  
    }  
}
```

```
Provider(){  
    while(true){  
        wait(table) ;  
        if (i==0) {  
            将组合 1 放在桌上;  
            signal (S1) ;  
        } else if (i==1) {  
            将组合 2 放在桌上;  
            signal (S2) ;  
        } else if (i==2) {  
            将组合 3 放在桌上;  
            signal (S3) ;  
        }  
        i=(i+3)%3;  
    }  
}
```

```
Smoker2(){  
    while(true){  
        wait(S2);  
        从桌上拿走组合 2;  
        卷烟、抽掉;  
        signal(table);  
    }  
}
```

```
Smoker3(){  
    while(true){  
        wait(S3);  
        从桌上拿走组合 3;  
        卷烟、抽掉;  
        signal(table);  
    }  
}
```