

E03: 并发进程 (进程通信)

一、单项选择题

- 临界区是指 A D **访问共用资源**。
 - 并发进程中用于实现进程互斥的程序段
 - 并发进程中用于实现进程同步的程序段
 - 并发进程中用户实现进程通信的程序段
 - 并发进程中访问共享变量的程序段
- 相关临界区是指 B D **若干进程共享一个临界区, 涉及相同变量那么称为相关临界区**。
 - 一个独占资源
 - 并发进程中与共享变量有关的程序段
 - 一个共享资源
 - 并发进程中涉及相同变量的那些程序段
- P、V 操作是 A。
 - 两条低级进程通信原语
 - 两组不同的机器指令
 - 两条系统调用命令
 - 两条高级进程通信原语
- 若 P、V 操作的信号量 S 初值为 2, 当前值为 -1, 则表示有 B 等待进程。
 - 0 个
 - 1 个
 - 2 个
 - 3 个
- 用 V 操作唤醒一个等待进程时, 被唤醒进程的状态变为 B。
 - 等待
 - 就绪
 - 运行
 - 完成
- 在执行一次 P 操作时, 信号量的值应 C。
 - 不变
 - 加 1
 - 减 1
 - 减指定数值
- 在执行 V 操作时, 当信号量的值 D 时, 应释放一个等待该信号量的进程。
 - >0
 - <0
 - >=0
 - <=0
- 进程间的同步是指进程间在逻辑上的相互 B 关系。
 - 联接
 - 制约
 - 继续
 - 调用
- D 是一种只能进行 P 操作和 V 操作的特殊变量。
 - 调度
 - 进程
 - 同步
 - 信号量
- P、V 操作这种不可被中断的过程称为 B。
 - 初始化程序
 - 原语
 - 子程序
 - 控制模块
- 实现进程互斥时, 用 C 对应, 对同一个信号量调用 P、V 操作实现互斥。
 - 一个信号量与一个临界区
 - 一个信号量与一个相关临界区
 - 一个信号量与一组相关临界区
 - 一个信号量与一个消息
- 实现进程同步时, 每一个消息与一个信号量对应, 进程 C D 可把不同的消息发送出去。
 - 在同一信号量上调用 P 操作
 - 在不同信号量上调用 P 操作
 - 在同一信号量上调用 V 操作
 - 在不同信号量上调用 V 操作

三、填空题

- 临界资源的概念是 一次只能允许一个进程使用的资源 ①, 而临界区是指 使用临界资源的互斥执行的程序段 ②。
- 若一个进程已进入临界区, 其他欲进入临界区的进程必须 等待其使用完临界资源。

15. 用 P、V 操作管理临界区时, 任何一个进程在进入临界区之前应调用 P操作 ① 操作, 退出临界区时应调用 V操作 ② 操作。
16. 有 m 个进程共享同一临界资源, 若使用信号量机制实现对临界资源的互斥访问, 则信号量值的变化范围是 $[-m+1, 1]$
17. 操作系统中, 对信号量 S 的 P 原语操作定义中, 使进程进入相应等待队列等待的条件是 $S < 0$ 。
18. 进程的 互斥 是指当有若干进程都要使用某一共享资源时, 任何时刻最多只允许一个进程去使用。
19. 利用 PV 操作管理相关临界区时, 必须成对出现, 在进入临界区之前要调用 P操作 ①, 在完成临界区操作后要调用 V操作 ②。
20. 若信号量的初值为 1, 用 PV 操作能限制一次 只有一个 进程进入临界区操作。
21. 进程的 同步 是指并发进程之间存在一种制约关系, 一个进程的执行依赖另一个进程的消息。
22. 用 PV 操作实现进程同步时, 调用 P操作 ① 测试消息是否到达, 调用 V操作 ② 发送消息。

五、简答题

23. 请给出 PV 操作的定义。假设 PV 操作用信号量 s 管理某个共享资源, 请问当 $s.value > 0$, $s.value = 0$ 和 $s.value < 0$ 时, 它们的物理意义是什么?
24. 复印室里有一个操作员为顾客复印资料, 有 5 把椅子供顾客休息等待复印。如果没有顾客, 则操作员休息。当顾客来到复印室时, 如果有空椅子则坐下来, 并唤醒复印操作员; 如果没有空椅子则必须离开复印室。

23. PV操作由P操作原语和V操作原语组成, 对信号量进行操作, 实现进程的同步和互斥具体定义如下: P(S) 申请资源: 将信号量 S 的值减 1, 即 $S=S-1$; 如果 $S>0$, 则该进程继续执行; 否则该进程置为等待状态, 排入等待队列。
V(S) 释放资源: 将信号量 S 的值加 1, 即 $S=S+1$;

$S.value > 0$ 表示当前有一个可用的临界资源, $S.value = 0$ 表示当前可用资源全部被占用, 但是也没有进程在等待, $S.value < 0$ 表示当前无可用资源, 且有 S 的绝对值个进程在等待队列中

```
/*信号量:
operator=1 操作员数量 -1表示空闲, =0表示正在服务顾客
waiting=0 顾客等待的数量
*/

semaphore operator=1, waiting=0;
void operator()
{
    while(1)
    {
        wait(consumer); //等待顾客到来
        p(operator); //申请一个操作员进行服务
        operate(); //复印
        v(operator); //完成复印, 操作员空闲
        leave(consumer) //顾客离开
    }
}

void consumer()
{
    while(1)
    {
        if(waiting==0)
        {
            operator(); //如果没有人等待那么可以直接复印
        }
        else{
            if(waiting<5)
            {
                sitin();
                waiting++;
                operate(); //等待排第一的人完成复印
                waiting--;
            }
            else{
                leave() //如果等待人数大于5就离开
            }
        }
    }
}
```

<p>变量:</p> <p>waiting: 表示等待的顾客数量。</p> <p>信号量:</p> <p>mutex: 用于对waiting的互斥访问</p> <p>customers: 有等待复印的顾客</p> <p>operator: 有等候顾客的操作员</p>	<pre>process cusotmer() //顾客进程 { p(mutex); if(waiting<5) { waiting++; v(customers); v(mutex); p(operator); 复印; p(mutex); waiting--; v(mutex); 离开复印室; } }</pre>
<pre>int waiting = 0; semaphore mutex, customers, barbers; mutex.value = 1; customers.value = 0; barbers.value = 0;</pre>	<pre>process operator() //操作员进程 { while(1) { p(customers); //等待顾客到来 复印; v(operator); //顾客完成复印 } }</pre>
<pre>main() { cobegin { operator(); customer(); } }</pre>	