

同步与互斥

40. 【2010 统考真题】设与某资源关联的信号量初值为 3，当前值为 1。若 M 表示该资源的可用个数， N 表示等待该资源的进程数，则 M, N 分别是 (B)。
- A. 0, 1 B. 1, 0 C. 1, 2 D. 2, 0

41. 【2010 统考真题】进程 P_0 和进程 P_1 的共享变量定义及其初值为：

```
boolean flag[2];
```

```
int turn=0;
```

```
flag[0]=false; flag[1]=false;
```

若进程 P_0 和进程 P_1 访问临界资源的类 C 代码实现如下：

```
void P0()    //进程 P0
```

```
void P1()    //进程 P1
```

```
{
```

```
{
```

```
    while(true)
```

```
        while(true)
```

```
    {
```

```
    {
```

```
        flag[0]=true;turn=1;
```

```
        flag[1]=true;turn=0;
```

```
        while(flag[1]&&(turn==1));
```

```
        while(flag[0]&&(turn==0));
```

```
        临界区;
```

```
        临界区;
```

```
        flag[0]=false;
```

```
        flag[1]=false;
```

```
    }
```

```
    }
```

```
}
```

```
}
```

则并发执行进程 P_0 和进程 P_1 时产生的情况是 (D)。

- A. 不能保证进程互斥进入临界区，会出现“饥饿”现象
- B. 不能保证进程互斥进入临界区，不会出现“饥饿”现象
- C. 能保证进程互斥进入临界区，会出现“饥饿”现象
- D. 能保证进程互斥进入临界区，不会出现“饥饿”现象

42. 【2011 统考真题】有两个并发执行的进程 P_1 和进程 P_2 ，共享初值为 1 的变量 x 。 P_1 对 x 加 1， P_2 对 x 减 1。加 1 和减 1 操作的指令序列分别如下：

<pre>//加 1 操作 load R1, x //取 x 到寄存器 R1 inc R1 store x, R1 //将 R1 的内容存入 x</pre>	<pre>//减 1 操作 load R2, x //取 x 到寄存器 R2 dec R2 store x, R2 //将 R2 的内容存入 x</pre>
--	--

两个操作完成后， x 的值 (C)。

- A. 可能为 -1 或 3
B. 只能为 1
C. 可能为 0, 1 或 2
D. 可能为 -1, 0, 1 或 2
43. 【2016 统考真题】进程 P_1 和 P_2 均包含并发执行的线程，部分伪代码描述如下所示。

// 进程 P_1	// 进程 P_2
<pre>int x=0; Thread1() { int a; a=1; x += 1; } Thread2() { int a; a=2; x += 2; }</pre>	<pre>int x=0; Thread3() { int a; a=x; x += 3; } Thread4() { int b; b=x; x += 4; }</pre>

下列选项中，需要互斥执行的操作是 (C)。

- A. $a=1$ 与 $a=2$
B. $a=x$ 与 $b=x$
C. $x += 1$ 与 $x += 2$
D. $x += 1$ 与 $x += 3$

44. 【2016 统考真题】使用 TSL (Test and Set Lock) 指令实现进程互斥的伪代码如下所示。

```
do{
    ...
    while(TSL(&lock));
    critical section;
    lock=FALSE;
    ...
} while(TRUE);
```

下列与该实现机制相关的叙述中, 正确的是 (B)。

- A. 退出临界区的进程负责唤醒阻塞态进程
- B. 等待进入临界区的进程不会主动放弃 CPU
- C. 上述伪代码满足“让权等待”的同步准则
- D. while(TSL(&lock))语句应在关中断状态下执行

45. 【2016 统考真题】下列关于管程的叙述中, 错误的是 (A)。

- A. 管程只能用于实现进程的互斥
- B. 管程是由编程语言支持的进程同步机制
- C. 任何时候只能有一个进程在管程中执行
- D. 管程中定义的变量只能被管程内的过程访问

46. 【2018 统考真题】属于同一进程的两个线程 thread1 和 thread2 并发执行, 共享初值为 0 的全局变量 x。thread1 和 thread2 实现对全局变量 x 加 1 的机器级代码描述如下。

thread1	thread2
mov R1, x // (x)→R1	mov R2, x // (x)→R2
inc R1 // (R1)+1→R1	inc R2 // (R2)+1→R2
mov x, R1 // (R1)→x	mov x, R2 // (R2)→x

在所有可能的指令执行序列中, 使 x 的值为 2 的序列个数是 (B)。

- A. 1
- B. 2
- C. 3
- D. 4

47. 【2018 统考真题】若 x 是管程内的条件变量, 则当进程执行 x.wait()时所做的工作是 (D)。

- A. 实现对变量 x 的互斥访问
- B. 唤醒一个在 x 上阻塞的进程
- C. 根据 x 的值判断该进程是否进入阻塞态
- D. 阻塞该进程, 并将之插入 x 的阻塞队列中

48. 【2018 统考真题】在下列同步机制中, 可以实现让权等待的是 (C)。

- A. Peterson 方法
- B. swap 指令
- C. 信号量方法
- D. TestAndSet 指令

49. 【2020 统考真题】下列准则中, 实现临界区互斥机制必须遵循的是 (C)。

- I. 两个进程不能同时进入临界区
 - II. 允许进程访问空闲的临界资源
 - III. 进程等待进入临界区的时间是有限的
 - IV. 不能进入临界区的执行态进程立即放弃 CPU
- A. 仅 I、IV
 - B. 仅 II、III
 - C. 仅 I、II、III
 - D. 仅 I、III、IV

18. 【2009 统考真题】三个进程 P_1, P_2, P_3 互斥使用一个包含 $N (N > 0)$ 个单元的缓冲区。 P_1 每次用 `produce()` 生成一个正整数并用 `put()` 送入缓冲区某一空单元； P_2 每次用 `getodd()` 从该缓冲区中取出一个奇数并用 `countodd()` 统计奇数个数； P_3 每次用 `geteven()` 从该缓冲区中取出一个偶数并用 `counteven()` 统计偶数个数。请用信号量机制实现这三个进程的同步与互斥活动，并说明所定义的信号量的含义（要求用伪代码描述）。

答：

```
semaphore mutex = 1          // 用于缓冲区的互斥访问
semaphore empty = N          // 表示缓冲区的空单元数，初始为N
semaphore odd = 0             // 表示缓冲区中奇数的个数
semaphore even = 0            // 表示缓冲区中偶数的个数
int count_odd = 0             // 统计奇数个数
int count_even = 0            // 统计偶数个数
```

// P1, 生产正整数并放入缓冲区

```
process P1 {
    while (true) {
        int item = produce();
        P(empty);
        P(mutex);
        put(item);
        V(mutex);
        if (item % 2 == 0) {
            V(even);
        } else {
            V(odd);
        }
    }
}
```

// P2, 从缓冲区取奇数并统计

```
process P2 {
    while (true) {
        P(odd);
        P(mutex);
        int item = getodd();
        count_odd = countodd();
        V(mutex);
        V(empty);
    }
}
```

// P3, 从缓冲区取偶数并统计

```
process P3 {
    while (true) {
        P(even);
        P(mutex);
        int item = geteven();
        count_even = counteven();
        V(mutex);
        V(empty);
    }
}
```

19. 【2011 统考真题】某银行提供 1 个服务窗口和 10 个供顾客等待的座位。顾客到达银行时，若有空座位，则到取号机上领取一个号，等待叫号。取号机每次仅允许一位顾客使用。当营业员空闲时，通过叫号选取一位顾客，并为其服务。顾客和营业员的活动过程描述如下：

```
cobegin
{
    process 顾客 i
    {
        从取号机获取一个号码；
        等待叫号；
        获取服务；
    }
    process 营业员
    {
        While (TRUE)
        {
            叫号；
            为客户服务；
        }
    }
}coend
```

请添加必要的信号量和 P, V [或 wait(), signal()] 操作，实现上述过程中的互斥与同步。要求写出完整的过程，说明信号量的含义并赋初值。

答：

```
semaphore waitingSeats = 10;    // 空座位数量
semaphore ticketMachine = 1;    // 取号机的互斥信号量
semaphore customerReady = 0;    // 准备好接受服务的顾客数量
semaphore serve = 0;            // 服务同步信号
```

```
process 顾客i {
    P(ticketMachine);
    // 从取号机获取一个号码
    V(ticketMachine);
    P(waitingSeats);
    // 等待叫号
    V(customerReady);
    P(serve);
    // 获取服务
    V(waitingSeats);
}
```

```
process 营业员 {
    while (TRUE)
    {
        P(customerReady);
        // 叫号
        // 为顾客服务
        V(serve);
    }
}
```

20. 【2013 统考真题】某博物馆最多可容纳 500 人同时参观，有一个出入口，该出入口一次仅允许一人通过。参观者的活动描述如下：

```
cobegin
    参观者进程 i:
    {
        ...
        进门。
        ...
        参观;
        ...
        出门;
        ...
    }
coend
```

请添加必要的信号量和 P, V [或 wait(), signal()] 操作，以实现上述过程中的互斥与同步。要求写出完整的过程，说明信号量的含义并赋初值。

答：

```
semaphore door = 1;        // 控制门的互斥信号量
semaphore visitors = 500; // 控制博物馆内人数的信号量
```

```
// 参观者进程i
process Visitor_i {
    P(visitors);
    P(door);
    // 进门
    V(door);
    // 参观
    P(door);
    // 出门
    V(door);
    V(visitors);
}
```

21. 【2014 统考真题】系统中有多个生产者进程和多个消费者进程，共享一个能存放 1000 件产品的环形缓冲区（初始为空）。缓冲区未滿时，生产者进程可以放入其生产的一件

产品，否则等待；缓冲区未空时，消费者进程可从缓冲区取走一件产品，否则等待。要求一个消费者进程从缓冲区连续取出 10 件产品后，其他消费者进程才可以取产品。请使用信号量 P, V (wait(), signal()) 操作实现进程间的互斥与同步，要求写出完整的过程，并说明所用信号量的含义和初值。

答：

```
semaphore empty = 1000; // 缓冲区空位数
semaphore full = 0;     // 缓冲区产品数
semaphore mutex = 1;    // 互斥信号量
semaphore turn = 1;     // 控制消费者顺序，取10件后才能到下一个
```

```
// 生产者进程
process Producer {
    while (TRUE) {
        // 生产产品
        P(empty);
        P(mutex);
        // 将产品放入缓冲区
```

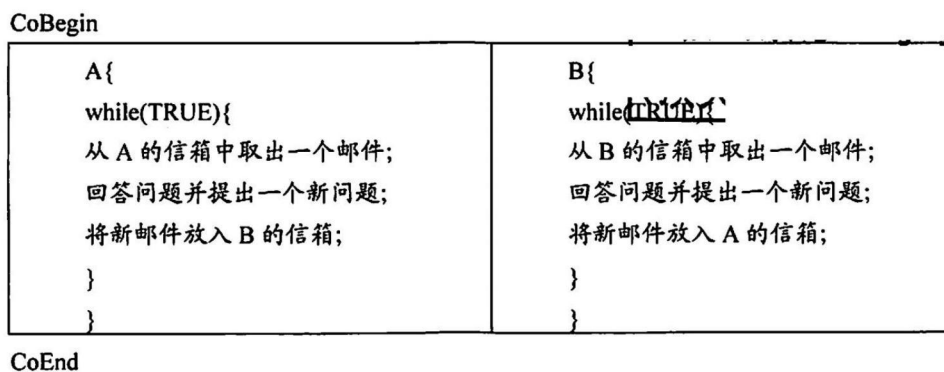
```

    V(mutex);
    V(full);
}
}

// 消费者进程
process Consumer {
    while (TRUE) {
        P(turn);
        for (int i = 0; i < 10; i++) {
            P(full);
            P(mutex);
            // 从缓冲区取产品
            V(mutex);
            V(empty);
        }
        V(turn);          // 允许其他消费者取产品
    }
}

```

22. 【2015 统考真题】有 A, B 两人通过信箱进行辩论, 每个人都从自己的信箱中取得对方的问题。将答案和向对方提出的新问题组成一个邮件放入对方的邮箱中。假设 A 的信箱最多放 M 个邮件, B 的信箱最多放 N 个邮件。初始时 A 的信箱中有 x 个邮件 ($0 < x < M$), B 的信箱中有 y 个邮件 ($0 < y < N$)。辩论者每取出一个邮件, 邮件数减 1。A 和 B 两人的操作过程描述如下:



当信箱不为空时, 辩论者才能从信箱中取邮件, 否则等待。当信箱不满时, 辩论者才能将新邮件放入信箱, 否则等待。请添加必要的信号量和 P, V [或 wait(), signal()] 操作, 以实现上述过程的同步。要求写出完整的过程, 并说明信号量的含义和初值。

答:

```

semaphore mailsInA = x;          // A信箱中的邮件数量
semaphore mailsInB = y;          // B信箱中的邮件数量
semaphore emptyInA = M - x;      // A信箱中的空位数量
semaphore emptyInB = N - y;      // B信箱中的空位数量
semaphore mutexA = 1;            // A信箱的互斥信号量
semaphore mutexB = 1;            // B信箱的互斥信号量

```

```

process A {
    while (TRUE) {
        P(mailsInA);
        P(mutexA);
        // 从A的信箱中取出一个邮件
    }
}

```

```

    V(mutexA);
    V(emptyInA);
    // 回答问题并提出一个新问题
    P(emptyInB);
    P(mutexB);
    // 将新邮件放入B的信箱
    V(mutexB);
    V(mailsInB);
}
}

// 辩论者B的过程
process B {
    while (TRUE) {
        P(mailsInB);
        P(mutexB);
        // 从B的信箱中取出一个邮件
        V(mutexB);
        V(emptyInB);
        // 回答问题并提出一个新问题
        P(emptyInA);
        P(mutexA);
        // 将新邮件放入A的信箱
        V(mutexA);
        V(mailsInA);
    }
}

```

23. 【2017 统考真题】某进程中有 3 个并发执行的线程 thread1, thread2 和 thread3, 其伪代码如下所示。

<pre> // 复数的结构类型定义 typedef struct { float a; float b; } cnum; cnum x, y, z; // 全局变量 // 计算两个复数之和 cnum add(cnum p, cnum q) { cnum s; s.a = p.a+q.a; s.b = p.b+q.b; return s; } </pre>	<pre> thread1 { cnum w; w = add(x, y); ... } thread2 { cnum w; w = add(y, z); ... } </pre>	<pre> thread3 { cnum w; w.a = 1; w.b = 1; z = add(z, w); y = add(y, w); ... } </pre>
--	---	--

请添加必要的信号量和 P, V [或 wait(), signal()] 操作, 要求确保线程互斥访问临界资源, 并且最大限度地并发执行。

答:

```

semaphore mutex_y1; // 用于thread1和thread3对变量y的互斥访问
semaphore mutex_y2; // 用于thread2和thread3对变量y的互斥访问
semaphore mutex_z; //用于变量z的互斥访问

```



```

//复数的结构类型定义
typedef struct
{
    float a;
    float b;
}cnum;
cnum x,y,z; //全局变量

//计算两个复数之和
cnum add(cnum p, cnum q)
{
    cnum s;
    s.a=p.a+q.a;
    s.b=p.b+q.b;
    return s;
}
thread1{
cnum w;
P (mutex_y1) ;
w=add (x, y) ;
V (mutex_y1) ;
.....
}
thread2{
cnum w;
P (mutex_y2) ;
P (mutex_z) ;
w=add (y, z) ;
V (mutex_y2) ;
V (mutex_z) ;
.....
}
thread3{
cnum w;
w.a=1;
w.b=1;
P (mutex_z) ;
z=add (z, w) ;
V (mutex_z) ;
P (mutex_y1) ;
P (mutex_y2) ;
y=add (y, w) ;
V (mutex_y1) ;
V (mutex_y2) ;
.....
}

```

24. 【2019 统考真题】有 n ($n \geq 3$) 名哲学家围坐在一张圆桌边, 每名哲学家交替地就餐和思考。在圆桌中心有 m ($m \geq 1$) 个碗, 每两名哲学家之间有一根筷子。每名哲学家必须取到一个碗和两侧的筷子后, 才能就餐, 进餐完毕, 将碗和筷子放回原位, 并继续思考。为使尽可能多的哲学家同时就餐, 且防止出现死锁现象, 请使用信号量的 P, V 操作[wait(), signal()操作] 描述上述过程中的互斥与同步, 并说明所用信号量及初值的含义。

答:

```
semaphore chopsticks[n]; // 假设有n根筷子, 每根筷子一个信号量
for (int i = 0; i < n; i++) {
    chopsticks[i] = 1; // 每根筷子的初始值为1
}
semaphore bowls = m; // 假设有m个碗
semaphore mutex = 1; // 避免死锁的互斥信号量

process 哲学家i {
    while (TRUE) {
        // 哲学家思考
        P(mutex);
        P(bowls);
        P(chopsticks[i]);
        P(chopsticks[(i + 1) % n]);
        V(mutex);
        // 哲学家就餐
        V(chopsticks[i]);
        V(chopsticks[(i + 1) % n]);
        V(bowls);
    }
}
```

25. 【2020 统考真题】现有 5 个操作 A、B、C、D 和 E, 操作 C 必须在 A 和 B 完成后执行, 操作 E 必须在 C 和 D 完成后执行, 请使用信号量的 wait()、signal()操作 (P、V 操作) 描述上述操作之间的同步关系, 并说明所用信号量及其初值。

答:

```
semaphore semA = 0; // A操作未完成
semaphore semB = 0; // B操作未完成
semaphore semC = 0; // C操作未完成
semaphore semD = 0; // D操作未完成
process A {
    ... // 执行操作A的相关任务
    V(semA);
}
process B {
    ... // 执行操作B的相关任务
    V(semB);
}
process C {
    P(semA);
    P(semB);
    ... // 执行操作C的相关任务
```

```

    V(semC);
}
process D {
    ... // 执行操作D的相关任务
    V(semD);
}
process E {
    P(semC);
    P(semD);
    ... // 执行操作E的相关任务
}

```

26. 【2021 统考真题】下表给出了整型信号量 S 的 wait()和 signal()操作的功能描述, 以及采用开/关中断指令实现信号量操作互斥的两种方法。

功能描述	方法 1	方法 2
<pre> Semaphore S; wait(S){ while(S<=0); S=S-1; } signal(S){ S=S+1; } </pre>	<pre> Semaphore S; wait(S){ 关中断; while(S<=0); S=S-1; 开中断; } signal(S){ 关中断; S=S+1; 开中断; } </pre>	<pre> Semaphore S; wait(S){ 关中断; while(S<=0){ 开中断; 关中断; } S=S-1; 开中断; } signal(S){ 关中断; S=S+1; 开中断; } </pre>

请回答下列问题。

- 1) 为什么在 wait()和 signal()操作中对信号量 S 的访问必须互斥执行?
- 2) 分别说明方法 1 和方法 2 是否正确。若不正确, 请说明理由。
- 3) 用户程序能否使用开/关中断指令实现临界区互斥? 为什么?

答:

1) 信号量S是多个进程共享的变量, 它们都能对S进行读写操作, 为了避免数据不一致, 对S的访问需要是互斥的。

2) 方法1是错误的, 在wait()中, 当 $S \leq 0$ 时, 关中断后, 其他进程都无法修改S的值, while语句会陷入死循环。方法2是正确的, 方法2在每次循环中都开中断, 使得其他进程可以修改S的值, 从而避免while语句陷入死循环。

3) 用户程序不能使用开/关中断指令实现临界区互斥, 因为这两个指令都是特权指令, 不能在用户态下执行, 只能在内核态下执行。

死锁

26. 【2009 统考真题】某计算机系统中有 8 台打印机，由 K 个进程竞争使用，每个进程最多需要 3 台打印机。该系统可能会发生死锁的 K 的最小值是 (C)。

A. 2 B. 3 C. 4 D. 5

27. 【2011 统考真题】某时刻进程的资源使用情况见下表，此时的安全序列是 (D)。

A. P_1, P_2, P_3, P_4 B. P_1, P_3, P_2, P_4
C. P_1, P_4, P_3, P_2 D. 不存在

进程	已分配资源			尚需分配			可用资源		
	R_1	R_2	R_3	R_1	R_2	R_3	R_1	R_2	R_3
P_1	2	0	0	0	0	1	0	2	1
P_2	1	2	0	1	3	2			
P_3	0	1	1	1	3	1			
P_4	0	0	1	2	0	0			

28. 【2012 统考真题】假设 5 个进程 P_0, P_1, P_2, P_3, P_4 共享三类资源 R_1, R_2, R_3 ，这些资源总数分别为 18, 6, 22。 T_0 时刻的资源分配情况如下表所示，此时存在的一个安全序列是 (D)。

进程	已分配资源			资源最大需求		
	R_1	R_2	R_3	R_1	R_2	R_3
P_0	3	2	3	5	5	10
P_1	4	0	3	5	3	6
P_2	4	0	5	4	0	11
P_3	2	0	4	4	2	5
P_4	3	1	4	4	2	4

A. P_0, P_2, P_4, P_1, P_3 B. P_1, P_0, P_3, P_4, P_2 C. P_2, P_1, P_0, P_3, P_4 D. P_3, P_4, P_2, P_1, P_0

29. 【2013 统考真题】下列关于银行家算法的叙述中，正确的是 (B)。

A. 银行家算法可以预防死锁
B. 当系统处于安全状态时，系统中一定无死锁进程
C. 当系统处于不安全状态时，系统中一定会出现死锁进程
D. 银行家算法破坏了死锁必要条件中的“请求和保持”条件

30. 【2014 统考真题】某系统有 n 台互斥使用的同类设备，三个并发进程分别需要 3, 4, 5 台设备，可确保系统不发生死锁的设备数 n 最小为 (B)。

A. 9 B. 10 C. 11 D. 12

31. 【2015 统考真题】若系统 S_1 采用死锁避免方法， S_2 采用死锁检测方法。下列叙述中，正确的是 (B)。

I. S_1 会限制用户申请资源的顺序，而 S_2 不会
II. S_1 需要进程运行所需的资源总量信息，而 S_2 不需要
III. S_1 不会给可能导致死锁的进程分配资源，而 S_2 会

A. 仅 I、II B. 仅 II、III C. 仅 I、III D. I、II、III

32. 【2016 统考真题】系统中有 3 个不同的临界资源 R_1, R_2 和 R_3 ，被 4 个进程 P_1, P_2, P_3, P_4 共享。各进程对资源的需求为： P_1 申请 R_1 和 R_2 ， P_2 申请 R_2 和 R_3 ， P_3 申请 R_1 和 R_3 ， P_4 申请 R_2 。若系统出现死锁，则处于死锁状态的进程数至少是 (C)。

A. 1 B. 2 C. 3 D. 4

33. 【2018 统考真题】假设系统中有 4 个同类资源，进程 P_1, P_2 和 P_3 需要的资源数分别为 4, 3 和 1， P_1, P_2 和 P_3 已申请到的资源数分别为 2, 1 和 0，则执行安全性检测算法的结果是 (A)。

A. 不存在安全序列，系统处于不安全状态

- B. 存在多个安全序列，系统处于安全状态
 C. 存在唯一安全序列 P_3, P_1, P_2 ，系统处于安全状态
 D. 存在唯一安全序列 P_3, P_2, P_1 ，系统处于安全状态
34. 【2019 统考真题】下列关于死锁的叙述中，正确的是 (B)。
- I. 可以通过剥夺进程资源解除死锁
 II. 死锁的预防方法能确保系统不发生死锁
 III. 银行家算法可以判断系统是否处于死锁状态
 IV. 当系统出现死锁时，必然有两个或两个以上的进程处于阻塞态
- A. 仅 II、III B. 仅 I、II、IV C. 仅 I、II、III D. 仅 I、III、IV
35. 【2020 统考真题】某系统中有 A、B 两类资源各 6 个， t 时刻的资源分配及需求情况如下表所示。

进程	A 已分配数量	B 已分配数量	A 需求总量	B 需求总量
P_1	2	3	4	4
P_2	2	1	3	1
P_3	1	2	3	4

- t 时刻安全性检测结果是 (B)。
- A. 存在安全序列 P_1, P_2, P_3 B. 存在安全序列 P_2, P_1, P_3
 C. 存在安全序列 P_2, P_3, P_1 D. 不存在安全序列
36. 【2021 统考真题】若系统中有 n ($n \geq 2$) 个进程，每个进程均需要使用某类临界资源 2 个，则系统不会发生死锁所需的该类资源总数至少是 (C)。
- A. 2 B. n C. $n+1$ D. $2n$