作业答案

参考答案：

1．B，2．C ，3. C，4. B，5. A，6. D，7. C，8. B，9. C，10. B

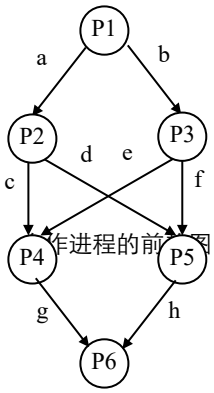
11．信号量 mutex 的作用是保证各生产者进程和消费者进程对缓冲池的互斥访问。信号量 empty 和 full 均是资源信号量，它们分别对应于缓冲池中的空闲缓冲区和缓冲池中的产品，生产者需要通过 wait(empty)来申请使用空闲缓冲区，而消费者需要通过 wait(full)才能取得缓冲中的产品，可见，这两个信号量起着同步生产者和消费者的作用，它们保证生产者不会将产品存放到满缓冲区中，而消费者不会从空缓冲区中取产品。

在生产者—消费者问题中，如果将两个 wait 操作，即 wait(full)和 wait(mutex)互换位置，或者 wait(empty)和 wait(mutex)互换位置，都可能引起死锁。考虑系统中缓冲区全满时，若一生产者进程先执行了 wait(mutex)操作并获得成功，当再执行 wait(empty)操作时，它将因失败而进入阻塞状态，它期待消费者执行 signal(empty)来唤醒自己，在此之前，它不可能执行 signal(mutex)操作，从而使企图通过 wait(mutex)进入自己的临界区的其他生产者和所有的消费者进程全部进入阻塞状态，系统进入死锁状态。类似地，消费者进程若先执行 wait(mutex)，后执行 wait(full)同样可能造成死锁。

signal(full)和 signal(mutex)互换位置，或者 signal(empty)和 signal(mutex)互换位置，则不会引起死锁，其影响只是使某个临界资源的释放略为推迟一些。

12.

如图示并发进程之间的前趋关系，为了使上述进程同步，可设置 8 个信号量 a、b、c、d、e、f、g、h，它们的初值均为 0，而相应的进程可描述为（其中“…”表示进程原来的代码）：



作进程的前

main( )

cobegin{

P1( ) { …; signal(a); signal(b); }

P2( ){ wait(a); …; signal(c); signal(d); }

P3( ){ wait(b); …; signal(e); signal(f); }

P4( ){ wait(c); wait(e); …; signal(g); }

P5( ){ wait(d); wait(f); …; signal(h); }

P6( ){ wait(g); wait(h); …; }

}coend

13.

两个进程分别详细列出，如下的 P0 和 P1。如果 P0 和 P1 两个并发进程按照代码左边的带圈的数值的顺序执行，P0 和 P1 两个进程同时进入临界区。下表中列出个变量的值是交替执行的结果。两个进程按照这样顺序交替执行时，在执行到⑿步时，P0 进入临界区，执行到⒁步时，P1 进入临界区，这时 P0 和 P1 同时进入了临界区，没有实现互斥。

执行顺序

P_0 :

```
do{
    blocked[0]=true;          ①⑩
    while(turn !=0)           ③⑾
   {
    while(blocked[1]);
    turn=0;
   }
     编号为 0 的进程的临界区      ⑤⑿
     blocked[0]=false;          ⑦
     编号为 0 的进程的非临界区    ⑨
 } while (true)
```

P_1:

```
do{
    blocked[1]=true;          ②
    while(turn !=1)           ④
   {
    while(blocked[0]);        ⑥⑧
    turn=1;                   ⑬
   }
     编号为 1 的进程的临界区      ⑭
     blocked[1]=false;
     编号为 1 的进程的非临界区
 } while (true)
```

表 两个进程交替执行时各变量值的变化

	blocked[0]	blocked[1]	turn	
○	true	false	0	
○	true	true	0	
○	true	true	0	
○	true	true	0	
○	true	true	0	P_0 进入临界区
○	true	true	0	
○	false	true	0	P_0 退出临界区
○	false	true	0	
○	false	true	0	
○	true	true	0	
○	true	true	0	
○	true	true	0	P_0 进入临界区
○	true	true	1	
○	true	true	1	P_1 进入临界区

14.

（1）缓冲区是一互斥资源，因此设互斥信号量 mutex。

（2）同步问题：P1、P2 因为奇数的放置与取用而同步，设同步信号量 odd；P1、P3 因为偶数的放置于取用而同步，设同步信号量 even；P1、P2、P3 因为共享缓冲区，设同步信号量 empty。伪代码描述如下：

```
semaphore mutex = 1;
semaphore odd = 0; even = 0;
semaphore empty = N;

main()
  cobegin{
    Process P1
    while(true){
```

```
    number = produce();
    P(empty);
    P(mutex);
    put();
    V(mutex);
    If number % 2 == 0
        V(even);
    else
        V(odd);
}

Process P2
while(true){
    P(odd);
    P(mutex);
    getodd();
    V(mutex);
    V(empty);
    countodd();
```

```
    }

    Process P3
    while(true){
        P(even);
        P(mutex);
        geteven();
        V(mutex);
        V(empty);
        counteven();
    }
}coend
```