

E03: 并发进程（进程通信）参考答案

参考答案与说明

1. D
2. D
3. A
4. B
5. B
6. C
7. D
8. B
9. D
10. B
11. C
12. D
13. ①一次仅允许一个进程访问的资源 ②进程中访问临界资源的那段程序代码
14. 等待
15. ①P ②V
16. 1 至 $-(m-1)$
17. $s.value < 0$
18. 互斥
19. ①P 操作 ②v 操作
20. 只有一个
21. 同步
22. ①P 操作 ②v 操作
23. 答: PV 操作是指在信号量上进行的 P 操作和 V 操作。假定信号量为 s , 则 $P(s)$ 和 $v(s)$ 的定义如下:

```

procedure p(Var s:semaphore);
begin
    s.value:=s.value-1;
    if s.value<0 then sleep(s);
end;
procedure V(Var s:semaphore);
begin
    s.value:=s.value+1;
    if s.value<=0 then wakeup (s);
end;

```

其中, $sleep(s)$ 表示将调用 $P(s)$ 过程的进程置成“等待信号量 s ”的状态, 且将其排入等待队列。 $wakeup(s)$ 表示释放一个“等待信号量 s ”的进程, 该进程从等待队列退出, 并加入就绪队列中。

信号量 S 的物理意义如下: $s.value > 0$ 时, S 表示可使用的资源数或表示可同时使用资源的进程数。 $s.value = 0$ 时, 表示无资源可供使用和表示不允许进程再进入临界区。 $s.value < 0$ 时, $|s.value|$ 表示等待使用资源的进程个数或表示等待进入临界区的进程个数。
24. 参考伪代码如下:

变量:

waiting: 表示等待的顾客数量。

信号量:

mutex: 用于对*waiting*的互斥访问

customers: 有等待复印的顾客

operator: 有等候顾客的操作员

```
int waiting = 0;
```

```
semaphore mutex, customers, barbers ;
```

```
mutex.value : =1;
```

```
customers.value : =0;
```

```
barbers.value : =0;
```

```
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
    }
}
```

```
process cusotmeri() //顾客进程
```

```
{
    p(mutex);
    if(waiting<5)
    {
        waiting++;
        v(customers);
        v(mutex);

        p(operator);

        复印;

        p(mutex);
        waiting--;
        v(mutex);
    }
    else
    {
        v( mutex);
        离开复印室;
    }
}
```

```
main( )
```

```
{
    cobegin
    {
        operator();
        cusotmeri();
    }
}
```