

操作系统使用教程 第三章习题

计创18 181002222 连月菡

1.现代操作系统中为什么要引入“进程”的概念？它与程序有什么区别？

因为每个程序段都不能与环境隔绝,不能随心所欲地在处理机上运行,只用"程序"不能说明问题的本质,引入"进程"则可以把问题从简单、孤立、静止的描述,转而反映出系统的独立性、并发性、动态性、相互制约性。

进程和程序的区别:

a.动态性和静态性:进程是程序执行的过程,是动态的过程。程序是一组有序静态指令组成的集合,是一种静态概念。

b.进程包含进程控制块:进程实体都是由程序段和相应的数据段两部分组成的,这一特征和程序相近。

但进程的结构还要包含一个数据结构PCB,即进程控制块。

c.一对多关系:一个进程可以涉及一个或几个程序的执行,反之,同一程序可以对应一个或多个进程。即同一程序在某一指定的时刻,可以是几个不同进程的一部分。

2.试叙述进程的并发性和制约性。

a.并发性是指多道程序中多个进程同时向前推进的过程,每个进程总是与其它进程并发地执行的。

b.制约性是指一个进程的运行受到另一进程的制约。

3.进程的含义是什么？如何构成和描述进程？

进程是指程序在一个数据集合上运行的过程,是系统进行资源分配和调度运行的一个独立单位。

构成: 程序、数据、进程控制块

描述:根据应用的需要,对进程的描述会略有不同。最主要的是需要进程控制块PCB来描述进程的外部特性(名称、状态等)以及与其他进程的联系(通信关系)。

4.有三个并发进程，R负责从输入设备读入信息并传送给M，M将信息加工并传送给P，P将打印输出，写出下列条件下的并发程序：

a.双缓冲区，每个区大小为KB；

```
1  变量的初值如下：
2  mutex1=1, mutex2=1, empty1=KB, empty2=KB;
3  full1=0, full2=0, i=0, j=0, a=0, b=0;
4
5  var
6  mutex1, full1, empty1, mutex2, full2, empty2: semaphore;
7  i, j, a, b: integer;
8  Buffer1: array 0..KB-1 of item;
9  Buffer2: array 0..KB-1 of item;
10
11 procedure R
12 begin
```

```

13     while true do
14     begin
15         输入data1;
16         P(empty1);
17         P(mutex1);
18         Buffer1(i):=data1;
19         i:=(i+1)mod(KB);
20     V(mutex1);
21     V(full1);
22     end
23 end;
24
25 procedure M
26 begin
27     while true do
28     begin
29         P(full1);
30         P(mutex1);
31         data2:=Buffer1(j);
32         j:=(j+1)mod(KB);
33     V(mutex1);
34     V(empty1);
35     对data2加工;
36     P(empty2);
37     P(mutex2);
38     Buffer2(a):=data2;
39     a:=(a+1)mod(KB);
40     V(mutex2);
41     V(full2);
42     end
43 end;
44
45 procedure P:
46 begin
47     while true do
48     begin
49         P(full2);
50         P(mutex2)
51         data3:=Buffer(b);
52         b:=(b+1)mod(KB);
53     V(mutex2);
54     V(empty2);
55     打印 data3;
56     end
57 end;

```

b.单缓冲区，其大小为KB;

```

1  初值设定:
2  mutex=1, full=0, empty=KB, s=0, i=0, j=0, a=0, b=0;
3
4  var
5  mutex,full,empty,s:semaphore;
6  i,j,a,b: integer;
7  Buffer : array 0..KB-1 of item;

```

```

8
9  procedure R:
10 begin
11 while true do
12     begin
13         读入data1;
14         P(empty);
15         P(mutex);
16         buffer(i):=data1;
17         i:=(i+1) mod (KB);
18         V(mutex);
19         V(full)
20     end
21 end;
22
23 procedure M:
24 begin
25 while true do
26     begin
27         P(full);
28         P(mutex);
29         data2:=buffer(j);
30         j:=(j+1)mod(KB);
31         V(mutex);
32         加工信息 data2;
33         P(mutex);
34         buffer(a):=data2;
35         a:=(a+1)mod (KB);
36         V(mutex);
37         V(s);
38     end
39 end;
40
41 procedure P:
42 begin
43     while true do
44         begin
45             P(s);
46             P(mutex);
47             data3:=buffer(b);
48             b:=(b+1) mod(KB);
49             V(mutex);
50             V(empty);
51             打印 data3;
52         end
53     end;
54

```

5.在生产者与消费者问题的算法中，交换两个V操作的次序会有什么结果？交换两个P操作的次序呢？试说明理由。

交换两个V操作的次序不会发生死锁。V操作是释放资源,不存在死锁可能。

交换两个P操作的次序可能会发生死锁。例如,当无空缓冲区时,如果此时生产者先做互斥操作,即:
P(mutex),然后才做同步操作P(empty).由于此时empty=-1造成生产者被阻塞。当消费者行到互斥操作
P(mutex)时,由于生产者已执行过P(mutex)并未作释放,所以此时mutex=-1,造成消费者也被阻塞,生产者
等消费者释放空缓冲区,而消费者则等待生产者释放临界资源的使用权,所以两个进程都无法向前推进而
造成死锁。

6.设有三个进程A、B、C, 其中A与B构成一对生产者与消费者 (A为生产者, B为消费者), 共享一个由n个缓冲块组成的缓冲池;B与C也构成一对生产者和消费者(此时B为生产者,C为消费者),共享另一个由m个缓冲快组成的缓冲池。用P/V描述它们之间的同步关系。

```
1  初值设定:
2  mutex1=1, full=0, empty1=n, mutex2=1, full2=0, empty2=m,
3  i=0, j=0, a=0, b=0;
4
5  var
6  mutex1, full1, empty1, mutex2, full2, empty2, : semaphore;
7  i, j, a, b: integer;
8  Buffer1: array 0..n-1 of item;
9  Buffer2: array 0..m-1 of item;
10
11 procedure A;
12 begin
13     while true do
14     begin
15         Produce next product1;
16         P(empty1);
17         P(mutex1);
18         Buffer1(i):=product1;
19         i:=(i+1) mod(n);
20         V(mutex1);
21         V(full1);
22     end
23 end;
24
25 procedure B
26 begin
27     while true do
28     begin
29         P(full1);
30         P(mutex1);
31         product2:=buffer1(j);
32         j:=(j+1) mod(n);
33         V(mutex1);
34         V(empty1);
35         Consume product2;
36         Produce next product3;
37         P(empty2);
38         P(mutex2);
39         Buffer2(a):=product3;
40         a:=(a+1) mod(m);
41         V(mutex2);
42         V(full2);
```

```

43     end
44 end;
45
46 procedure C;
47 begin
48     while ture do
49     begin
50         P(full2);
51         P(mutex2);
52         product4:=buffer2(b);
53         b:=(b+1) mod(m);
54         V(mutex2);
55         V(empty2);
56         Consume product4;
57     end
58 end
59

```

7.思考题

【思考题】
用P、V操作解决司机与售票员的问题

司机进程:	售票员进程:
while (true) {	while (true) {
启动车辆	关门
正常驾驶	售票
到站停车	开门
}...	}...

```

1  初值设定:
2  S1=1, S2=0
3
4  S1:能否启动车辆 (S1=0表示司机不能启动车辆)
5  S2:能否开门 (S2=0表示售票员不能开门)
6
7  procedure Driver:
8  while(true)
9  {
10     P(S1)
11     启动车辆
12     正常行驶
13     到站停车
14     V(S2)
15 }
16
17 procedure Seller:

```

```
18 while(true)
19 {
20     关门
21     V(S1)
22     售票
23     P(S2)
24     开门
25 }
26
```