

第一章操 作系统引论

思考与练习题

- 1. 什么是操作系统？它的主要功能是什么？
- 2. 什么是多道程序设计技术？多道程序设计技术的主要特点是什么？
- 3. 批处理系统是怎样的一种操作系统？它的特点是什么？
- 4. 什么是分时系统？什么是实时系统？试从交互性，及时性，独立性，多路性，可靠性等几个方面比较分时系统和实施系统。
- 5. 实时系统分为哪俩种类型？
- 6. 操作系统主要特征是什么？
- 7. 操作系统也用户的接口有几种？它们各自用在什么场合？
- 8. “操作系统是控制硬件的软件”这一说法确切吗？为什么？
- 9. 设内存中有三道程序，A,B,C 它们按 A~B~C的先后顺序执行，它们进行“计算”和“I/o 操作”的时间如表 1-2所示，假设三道程序使用相同的 I/O 设备。

表 1-2 三道程序的操作时间			
操作 程序	计算	I/o 操作	计算
A	20	30	10
B	30	50	20
C	10	20	10

- (1) 试画出单道运行时三道程序的时间关系图，并计算完成三道程序要花多少时间。
- (2) 试画出多道运行时三道程序的时间关系图，并计算完成三道程序要花多少时间。
- 10. 将下列左右两列词连接起来形成意义最恰当的 5 对。

DOS

OS/2

UNIX

Linux

Windows NT

网络操作系统

自由软件

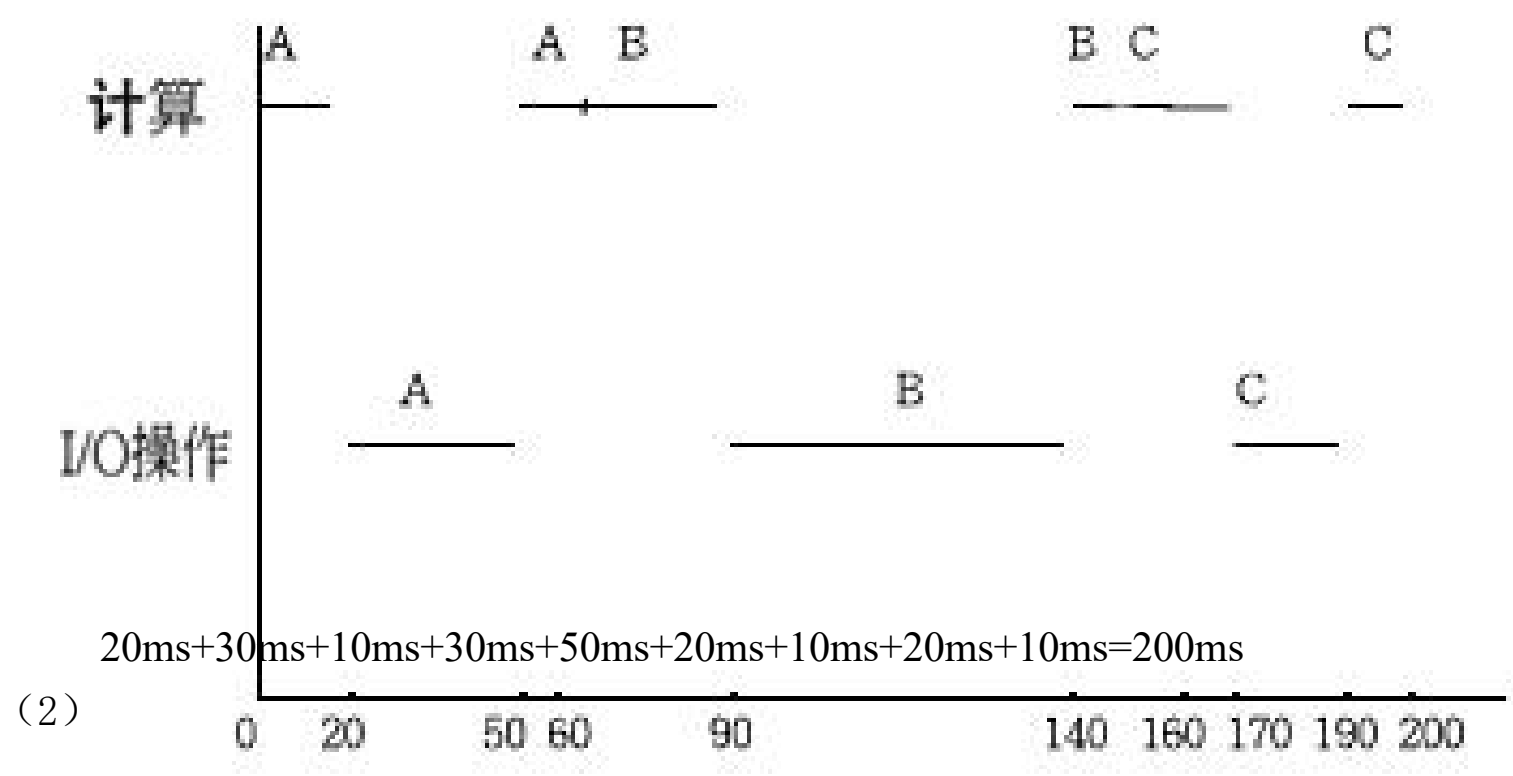
多任务

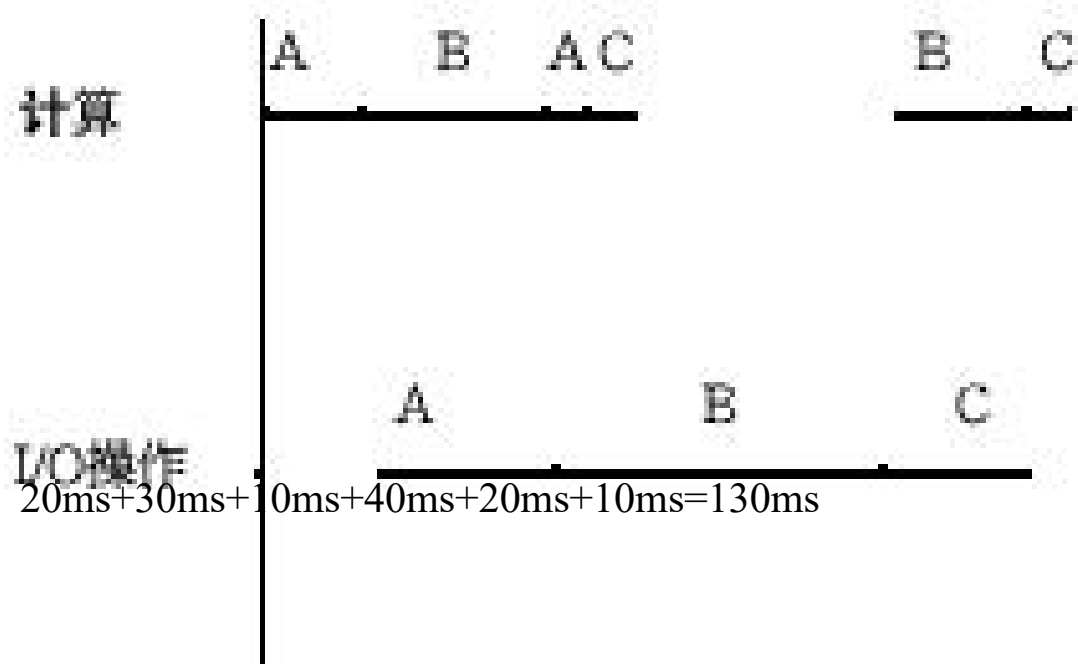
单任务

为开发操作系统而设计 C语言
- 11. 选择一个现代操作系统，查找和阅读相关的技术资料，写一篇关于操作系统如何进行内存管理、存储管理、设备管理和文件管理的文章。

答案

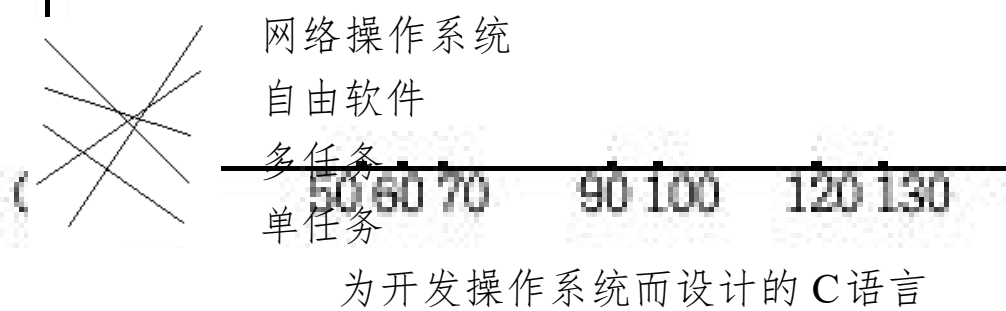
- 1. 答：操作系统是控制和管理计算机的软、硬件资源，合理地组织计算机的工作流程，以方便用户使用的程序集合。
- 2. 答：把多个独立的程序同时放入内存，使她们共享系统中的资源。
 - 1) 多道，即计算机内存中同时放多道相互独立的程序。
 - 2) 宏观上并行，是指共识进入系统的多道程序都处于运行过程。
 - 3) 微观上串行，是指在单道处理机环境下，内存中的多道程序轮流地占有 CPU 交替执行。
- 3. 答：批处理操作系统是一种基本的操作系统类型。在该系统中用户的作业被成批地输入到计算机中，然后在操作系统的控制下，用户的作业自动的执行。
特点是：资源利用率高。系统吞吐量大。平均周转时间长。无交互能力。
- 4. 答：分时系统：允许多个终端用户同时使用计算机，在这样的系统中，用户感觉不到其他用户的存在，好像独占计算机一样。实时系统：对外输入出信息，实时系统能够在规定的 时间内处理完毕并作出反应。
 - 1) 多路性：分时系统是为多个终端用户提供服务，实时系统的多路性主要表现在经常对多路的现场信息进行采集以及多多个对象或多个执行机构进行控制。
 - 2) 独立性：每个终端向实时系统提出服务请求时，是彼此独立的工作、互不干扰。
 - 3) 及时性：实时信息处理系统与分时系统对及时性的要求类似，都以人们能够接受的等待时间来确定。实时控制系统对一时性的要求更高，是以控制对象所要求的开始截止时间或完成截止时间来确定的。
- 5. 答：(1) 实时控制系统 (2) 实时信息处理系统。
- 6. 答：1) 并发性 2) 共享性 3) 虚拟性 4) 不确定性。
- 7. 答：两种，命令接口 ，程序接口。
命令接口：分为联机命令接口，脱机命令接口，图形用户命令接口。方便用户直接控制自己的作业而提供的接口。
程序接口：又称系统调用，是为了用户在程序一级访问操作系统功能而设置的。
- 8. 答：不正确，因为操作系统不仅仅是控制硬件，同时它还控制计算机的软件。
- 9. (1)





10.

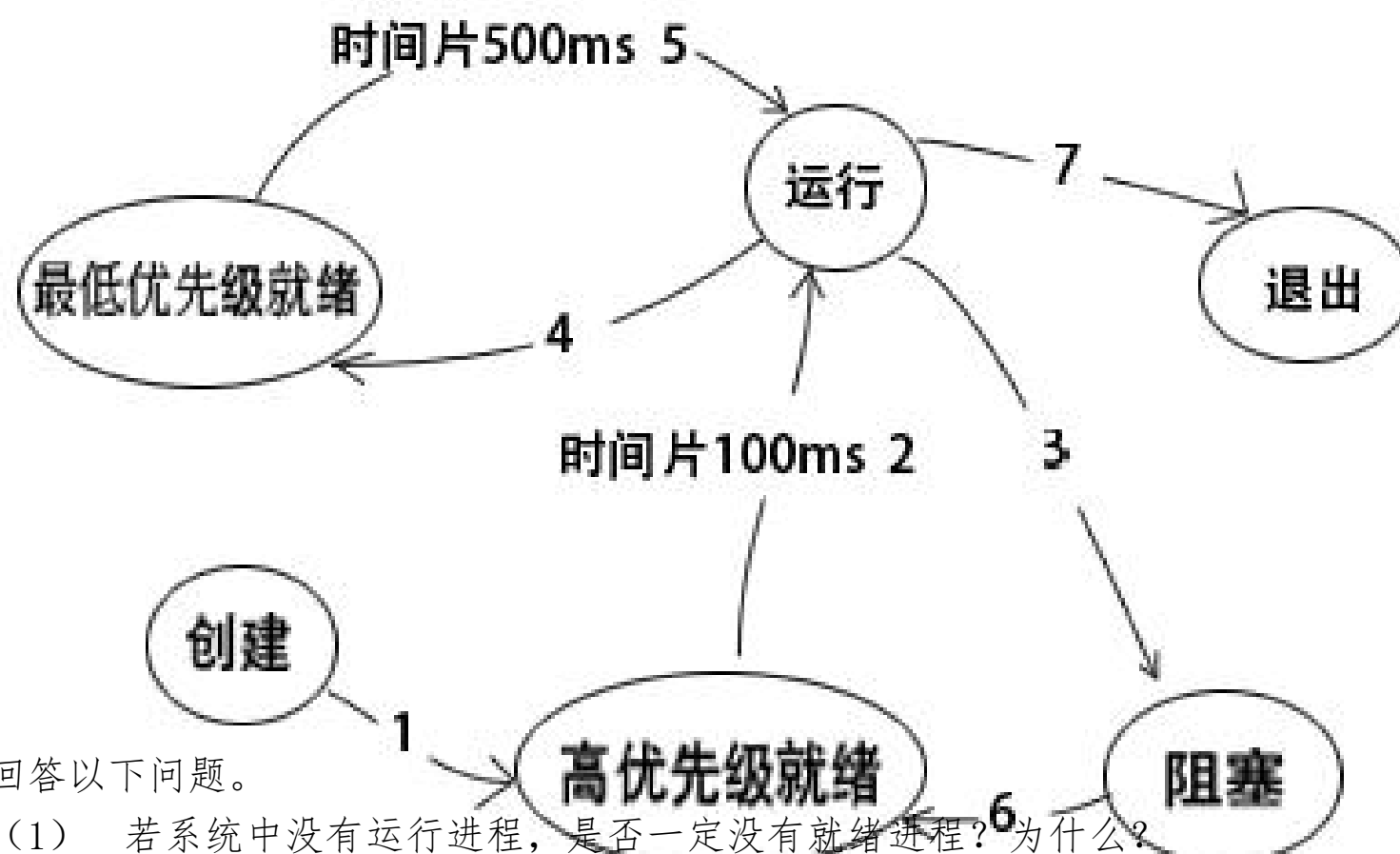
- DOS
- OS/2
- UNIX
- Linux
- WindowsNT



第二章 进程与线程

思考与练习题

1. 操作系统中为什么要引入进程的概念？为了实现并发进程之间的合作和协调，以及保证系统的安全，操作系统在进程管理方面要做哪些工作？
2. 试描述当前正在运行的进程状态改变时，操作系统进行进程切换的步骤。
3. 现代操作系统一般都提供多任务的环境，是回答以下问题。
 - (1) 为支持多进程的并发执行，系统必须建立哪些关于进程的数据结构？
 - (2) 为支持进程的状态变迁，系统至少应该供哪些进程控制原语？
 - (3) 当进程的状态变迁时，相应的数据结构发生变化吗？
4. 什么是进程控制块？从进程管理、中断处理、进程通信、文件管理、设备管理及存储管理的角度设计进程控制块应该包含的内容。
5. 假设系统就绪队列中有 10 个进程，这 10 个进程轮换执行，每隔 300ms 轮换一次，CPU 在进程切换时所花费的时间是 10ms，试问系统化在进程切换上的开销占系统整个时间的比例是多少？
6. 试述线程的特点及其与进程之间的关系。
7. 根据图 2-18，回答以下问题。
 - (1) 进程发生状态变迁 1、3、4、6、7 的原因。
 - (2) 系统中常常由于某一进程的状态变迁引起另一进程也产生状态变迁，这种变迁称为因果变迁。下述变迁是否为因果变迁：3~2, 4~5, 7~2, 3~6 是说明原因。
 - (3) 根据此进程状态转换图，说明该系统 CPU 调度的策略和效果。



8. 回答以下问题。
 - (1) 若系统中没有运行进程，是否一定没有就绪进程？为什么？
 - (2) 若系统中既没有运行进程，也没有就绪进程，系统中是否就没有阻塞进程？解释。
 - (3) 如果系统采用优先级调度策略，运行的进程是否一定是系统中优先级最高的进

程？为什么？

9. 假如有以下程序段，回答下面的问题。

S1: $a=3-x$;

S2: $b=2*a$;

S3: $c=5+a$;

(1) 并发程序执行的 **Bernstein** 条件是什么？

(2) 是画图表示它们执行时的先后次序。

(3) 利用 **Bernstein** 条件证明，S1、S2 和 S3 哪两个可以并发执行，哪两个不能。

答案

1. 答：①为了从变化角度动态地分析研究可以并发执行的程序，真实的反应系统的独立性、并发性、动态性和相互制约，操作系统中不得不引入进程的概念。

②为了防止操作系统及其关键的数据结构受到用户程序破坏，将处理机分为核心态和用户态。对进程进行创建、撤销以及在某些进程状态之间的转换控制。

2. 答：①运行状态→就绪状态：此进程根据自身的情况插入到就绪队列的适当位置，系统收回处理及转入进程调度程序重新进行调度。

②运行状态→阻塞状态：一个进程从运行状态道阻塞状态后。系统会调用进程调度程序重新选择一个进程投入运行。

3.

(1) 答：为支持多进程的并发执行，系统必须建立的数据结构式 **PCB** 不同状态进程的 **PCB**用链表组织起来，形成就绪队列、阻塞队列。

(2) 答：阻塞原句、唤醒原句、挂起原句、激活原句

(3) 答：创建原句：建立进程的 **PCB** 并将进程投入就绪队列。

撤销原句：删除进程的 **PCB** 并将进程在其队列中摘除。

阻塞原句：将进程 **PCB**中进程的状态从运行状态改为阻塞状态，并将进程投入阻塞队列。

唤醒原句：将进程 **PCB**中进程的状态从阻塞状态改为就绪状态，并将进程从则色队列摘下，投入到就绪队列中。

4. 答：进程控制块（**PCB**）是为了描述进程的动态变化而设置的一个与进程相联系的数据结构，用于记录系统管理进程所需信息。**PCB**是进程存在的唯一标识，操作系统通过 **PCB**得知进程的寻在。

为了进程管理，进程控制块包括以下几方面。

(1) 进程的描述信息，包括进程标识符、进程名等。

(2) 进程的当前状况。

(3) 当前队列链接指针。

(4) 进程的家族关系。

为了中断处理，进程控制块的内容应该包括处理机状态信息和各种寄存器的内容，如通用寄存器、指令计数器、程序状态字（**PSW**）寄存器及栈指针等。

为了内存管理的需要，进程控制块的内容应该包括进程使用的信号量、消息队列指针等。

为了设备管理，进程控制块的内容应该包括进程占有资源的情况。

5. 答：就绪队列中有 10 个进程，这 10 个进程轮换执行，每隔进程的运行时间是 300ms，切换另一个进程所花费的总时间是 10ms，隐刺系统化在进程切换上的时间开销占系统整个时间的比例是： $10/(300+10)=3.2\%$ 。

6. 答：线程是进程内的一个相对独立的运行单元，是操作系统调度和分派的单位。线程只拥有一点必不可少的资源（一组寄存器和栈），但可以和同属于一个进程的其他线程共享进程拥有的资源。

线程是进程的一部分，是进程内的一个实体；一个进程可以有多个线程，但至少必须有一个线程。

7.

- (1) 答：1 表示新进程创建后，进入高优先级就绪队列；3 表示进程因请求 I/O 活等待某件事儿阻塞；4 表示进程运行的时间片到；6 表示进程 I/O 完成或等待的时间到达；7 表示进程运行完成而退出。

- (2) 答：3→2 是因果变迁，当一个进程从运行态变为阻塞态时，此时 CPU 空闲，系统首先到高优先级队列中选择一个进程投入运行。

4→5 是因果变迁，当一个进程运行完毕时，此时 CPU 空闲，系统首先到高优先级队列中选择进程，但如果高优先级队列为空，则从低优先级队列中选择一个进程投入运行。

7→2 是因果变迁，当一个进程运行完毕时，CPU 空闲，系统首先到高优先级队列中选择一个进程投入运行。

3→6 不是因果变迁。一个进程阻塞时由于自身的原因而发生的，和另一个进程等待的时间到达没有因果关系。

- (3) 答：当进程调度时，首先从高优先级就绪队列选择一个进程，赋予它的时间片为 100ms。如果高优先级就绪队列为空，则从低优先级就绪队列选择进程，但赋予该进程的时间片为 500ms。

这种策略一方面照顾了短进程，一个进程如果在 100ms 运行完毕它将退出系统，更主要的是照顾了 I/O 量大的进程，进程因 I/O 进入阻塞队列，当 I/O 完成后它就进入了高优先级就绪队列，在高优先级就绪队列等待的进程总是优于低优先级就绪队列的进程。而对于计算量较大的进程，它的计算如果在 100ms 的时间内不能完成，它将进入低优先级就绪队列，在这个队列的进程被选中的机会要少，只有当高优先级就绪队列为空，才从低优先级就绪队列选择进程，但对于计算量大的进程，系统给予的适当照顾时间片增大为 500ms。

8.

- (1) 答：是。若系统中没有运行进程，系统会马上选择一个就绪进程队列中的进程投入运行。只有在就绪队列为空时，CPU 才会空闲。

- (2) 答：不一定。当系统中所有进程分别等待各自希望发生的事件时，它们都处于阻塞状态，此时系统中既没有运行进程，也没有就绪进程。这种情况出现时，如果各个进程没有相互等待关系，只要等待的时间发生了，进程就会从等待状态转化为就绪状态。但如果处于阻塞状态的进程相互等待彼此占有的资源，系统就有可能发生死锁。

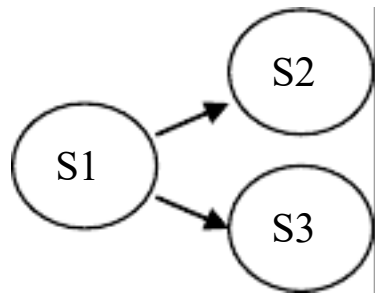
- (3) 答：不一定。因为高优先级的进程有可能处于等待状态，进程调度程序只能从就绪队列中挑选一个进程投入运行。被选中进程的优先级在就绪队列中是最高的，但在整个系统中它不一定是最高的，等待队列中进程的优先级有可能高于就绪队列中所有进程的优先级。

9.

(1) 答：P1 和 P2 并发执行的条件是，当且仅当

$$R(P1) \cap W(P2) \cup R(P2) \cap W(P1) \cup W(P1) \cap W(P2) = \{\}$$

(2)



(3) 答：R(S1)={x}, W(S2)={a}, R(S2)={a}, W(S2)={b}, R(S3)={a}, W(S3)={c}

所以 $W(S1) \cap R(S2) = \{a\}$, 因此 S1 和 S2 不能并发执行。

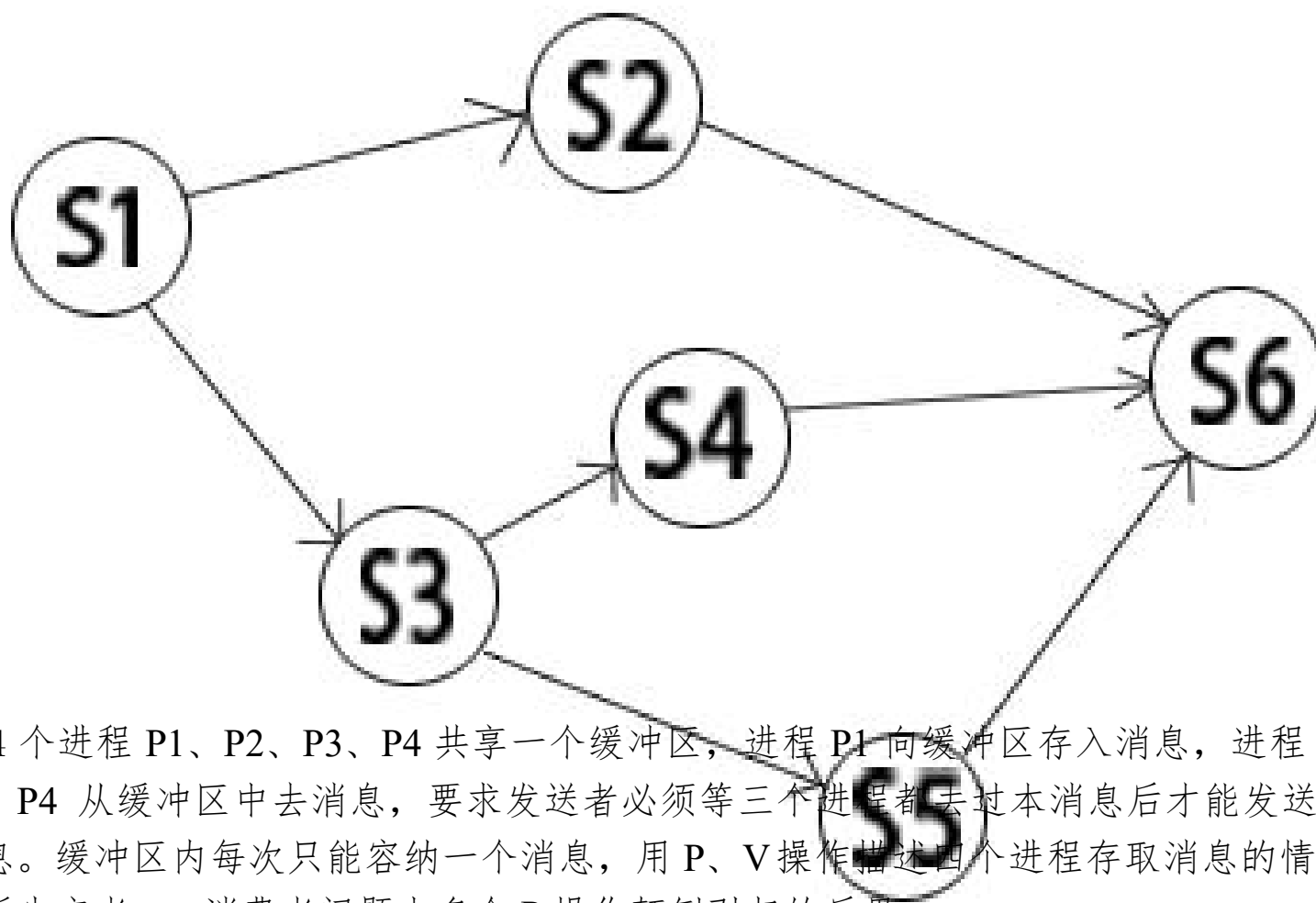
$W(S1) \cap R(S3) = \{a\}$, 因此 S1 和 S3 也不能并发执行。

而 $R(S2) \cap W(S3) \cup R(S3) \cap W(S2) \cup W(S2) \cap W(S3) = \{\}$, 因此 S2 和 S3 可以并发执行。

第三章 进程同步与通信

思考与练习题

- 一下进程之间存在相互制约关系吗？若存在，是什么制约关系？为什么？
 - 几个同学去图书馆借同一本书。
 - 篮球比赛中两队同学争抢篮板球。
 - 果汁流水线生产中捣碎、消毒、灌装、装箱等各道工序。
 - 商品的入库出库。
 - 工人做工与农民种粮。
- 在操作系统中引入管程的目的是什么？条件变量的作用是什么？
- 说明 P、V 操作为什么要设计成原语。
- 设有一个售票大厅，可容纳 200 人购票。如果厅内不足 200 人则允许进入，超过则在厅外等候；售票员某时只能给一个购票者服务，购票者买完票后就离开。试问：
 - 购票者之间是同步关系还是互斥关系？
 - 用 P、V 操作描述购票者的工作过程。
- 进程之间的关系如图 3-16 所示，试用 P、V 操作描述它们之间的同步。



- 有 4 个进程 P1、P2、P3、P4 共享一个缓冲区，进程 P1 向缓冲区存入消息，进程 P2、P3、P4 从缓冲区中去消息，要求发送者必须等三个进程都去过本消息后才能发送下一条消息。缓冲区内每次只能容纳一个消息，用 P、V 操作描述四个进程存取消息的情况。
- 分析生产者——消费者问题中多个 P 操作颠倒引起的后果。
- 读者——写者问题中写者优先的实现。
- 写一个用信号量解决哲学家进餐问题不产生锁死的算法。
- 一个文件可有若干个不同的进程所共享，每个进程具有唯一的编号。假定文件可有满足下列限制的若干个不同的进程同时访问，并发访问该文件的哪些进程的编号的总和不得大于 n，设计一个协调对该文件访问的管程。
- 用管程解决读者——写者问题，并采用公平原则。

- (1) 答：存在互斥关系，因为同一本书只能借给一个同学。
- (2) 答：存在互斥关系，因为篮球只有一个，两队只能有一个队抢到球
- (3) 答：存在同步关系，因为最后一道工序的开始依赖于前一道工序的完成。
- (4) 答：存在同步关系，因为商品若没有入库就无法出库，若商品没有出库，装满了库房，也就无法再入库。
- (5) 答：工人与农民之间没有相互制约关系。

2. 答：引入管程的目的是为了实现进程之间的同步和互斥。优于使用信号量在解决同步和互斥问题时要设置多个信号量，并使用大量的 P、V 操作，其中 P 操作的排列次序不当，还会引起系统死锁，因此引入另外一种同步机制。

3. 答：用信号量 S 表示共享资源，其初值为 1 表示有一个资源。设有两个进程申请该资源，若其中一个进程先执行 P 操作。P 操作中的减 1 操作有 3 跳及其指令组成：去 S 送寄存器 R;R-1 送 S。若 P 操作不用原语实现，在执行了前述三条指令中的 2 条，即还未执行 R 送 S 时（此时 S 值仍为 1），进程被剥夺 CPU，另一个进程执行也要执行 P 操作，执行后 S 的值为 0，导致信号量的值错误。正确的结果是两个进程执行完 P 操作后，信号量 S 的值为-1，进程阻塞。

4.

(1) 答：购票者之间是互斥关系。

(2)

答： semaphore empty=200;
semaphore mutex=1;
void buyer()
{ P(empty);
P(mutex);
购票;
V(mutex);
V(empty);
}

5.

答：

```
semaphore a,b,c,d,e,f,g=0,0,0,0,0,0,0;
void P1()    void P2()    void P3()    void P4()    void P5()    void P6()
{ S1;        { P(a);        { P(b);        { P(c);        { P(d);        { P(e)
  V(a);      S2;          S3;          S4;          S5;          P(f)
  V(b);      V(e);        V(c);          V(f);        V(g);        P(g)
}            }            V(d);          }            }            S6;
}                                     }            }
```

```

semaphore S1=1;
semaphore S2,S3,S4=0,0,0;
int count =0;
semaphore mutex=1 ;
void P1()/*发送进程*/ void P2()/*接收进程*/ void P3()/*接受进程*/ void P4()/*接受进程*/
{ while(true)          { while(true)          { while(true)          { while(true)
{
P(S1);                  P(S2);                  P(S3);                  P(S4);
发送消息;              接收消息;              接收消息;              接收消息;
P(mutex);              P(mutex);              P(mutex);              P(mutex);
count=0;              count=count+1;          count=count+1;          count=count+1;
V(mutex);              if (count==3) V(S1);    if (count==3) V(S1);    if (count==3) V(S1);
V(S2);                  V(mutex)}          V(mutex)}          V(mutex)}
V(S3);                  }                  }                  }
V(s4);} }

```

7

```

答： semaphore mutex=1 ;
semaphore empty=n ;
semaphore full=0;
int i,j;
ITEM buffer[n];
ITEM data_p,data_c;
void producer()/*生产者进程*/          void consumer() /*消费者进程*/
{while(true)                             {while(true)
{produce an item in data_p;                {P(full);
P(mutex);                                P(mutex);
P(empty);                                data_c=buffer[j];
buffer[i]=data_p;                        j=(j+1)%n;
i=(i+1)%n;                               V(mutex);
V(mutex);                                V(empty);
V(full);}                                consume the item in data_c}
}

```

6.

```

答： semaphore Wmutex , Rmutex=1;
int Rcount=0;
semaphore mutex=1
void reader ( ) /*读者进程*/          void writer() /*写者进程*/
{while(true)                             {while(true)
{P(mutex);                                {P(mutex);

```

```

P(Rmutex);
If(Rcount==0) P(wmutex);
Rcount=Rcount+1 ;
V(Rmutex);
V(mutex);
...;
read; /* 执行读操作 */
...;
P(Rmutex);
Rcount=Rcount-1;
if (Rcount==0) V (wmutex );
V (Rmutex); }
}

```

```

P(wmutex);

write; /*执行写操作*/
...;
V(Wmutex);
V(mutex);
}}

```

7.

答： semaphore chopstick[5]={1,1,1,1,1};
semaphore mutex=1;
void philosopher ()/* 哲学家进餐 */
{while(true)
{P(mutex);
P(chopstick[i]);
P(chopstick[(i+1)%5]);
V(mutex);
...;
eat; /*进餐*/
...;
V(chopstick[i]);
V(chopstick[(i+1)%5]);
...;
think; /* 思考 */
...; }
}

调度与死锁

思考与练习题

- 某进程被唤醒后立刻投入运行，能说明该系统采用的是可剥夺调度算法吗？
- 在哲学家进餐问题中，如果将先拿起左边筷子的哲学家称为左撇子，先拿起右边筷子的哲学家称为右撇子。请说明在同时存在左、右撇子的情况下，任何的就坐安排都不能产生锁死。
- 系统中有 5 个资源被 4 个进程所共享，如果每个进程最多需要 2 个这种资源，试问系统是否会产生锁死？
- 计算机系统有 8 台磁带机，由 N 个进程竞争使用，每个进程最多需要 3 台。问：N 为多少时，系统没有死锁的危险？
- 系统有 5 个进程，它们的到达时间和服务时间如表 4-8 所示。新进程（没有运行过）与老进程（运行过的进程）的条件相同时，假定系统选新进程运行。

表 4-8 进程情况

进程名	到达时间	服务时间
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2

若按先来先服务（FCFS）、时间片轮法（时间片 $q=1$ ）、短进程优先（SPN）、最短剩余时间优先（SRT，时间片 $q=1$ ）、响应比高者优先（HRRN）及多级反馈队列（MFQ，第一个队列的时间片为 1，第 i ($i>1$) 个队列的时间片 $q=2^{(i-1)}$) 算法进行 CPU 调度，请给出各个进程的完成时间、周转时间、带权周转时间，及所有的进程的平均周转时间和平均带权周转时间。

- 设系统中有 5 个进程 P1、P2、P3、P4、P5，有 3 种类型的资源 A、B、C，其中 A 资源的数量是 17，B 资源的数量是 5，C 资源的数量是 20，T0 时刻系统状态如表 4-9 所示。

表 4-9 T0 时刻系统状态

进程	已分配资源数量			最大资源需求量			仍然需求资源数		
	A	B	C	A	B	C	A	B	C
P1	2	1	2	5	5	9	3	4	7
P2	4	0	2	5	3	6	1	3	4
P3	4	0	5	4	0	11	0	0	6
P4	2	0	4	4	2	5	2	2	1
P5	3	1	4	4	2	4	1	1	0

- (1) 计算每个进程还可能需要的资源，并填入表的“仍然需要资源数”的栏目。
- (2) T0 时刻系统是否处于安全状态？为什么？
- (3) 如果 T0 时刻进程 P2 又有新的资源请求 (0,3,4)，是否实施资源分配？为什么？
- (4) 如果 T0 时刻，若进程 P4 又有新的资源请求 (2, 0, 1)，是否实施资源分配？为什么？

(5) 在 (4) 的基础上, 若进程 **P1** 又有新的资源请求 $(0, 2, 0)$, 是否实施资源分配? 为什么?

答：不能。如果当前就绪队列为空，这样被唤醒的进程就是就绪队列中的唯一的一个进程，于是调度程序自然选中它投入运行。

2. 答：该题的关键是证明该情况不满足产生死锁的四个必要条件之一。在死锁的四个必要条件中，本体对于互斥条件、请求与保持条件、不可剥夺条件肯定是成立的，因此必须证明环路条件不成立。

对于本体，如果存在环路条件必须是左、右的哲学家都拿起了左（或右）边的筷子，而等待右（或左）边的筷子，而这种情况只能出现在所有哲学家都是左（或右）撇子的情况下，但由于本题有右（或左）撇子存在，因此不可能出现循环等待链，所以不可能产生死锁。

3. 答：由于资源数大于进程数，所以系统中总会有一个进程获得资源数大于等于 2，该进程已经满足了它的最大需求，当它运行完毕后会把它占有的资源归还给系统，此时其余 3 个进程也能满足最大需求而顺利运行完毕。因此系统不会产生死锁。

4. 答：当 $N < 4$ 时，系统没有死锁的危险。因为当 N 为 1 时，它最多需要 3 台磁带机，系统中共有 8 台，其资源数已足够一个进程使用，因此绝对不会产生死锁，当 N 为 2 时，两个进程最多需要 6 台磁带机，系统中共有 8 台，其资源数也足够两个进程使用，因此也不会产生死锁；当 N 为 3 时，无论如何分配，3 个进程中必有进程得到 3 台磁带机，该进程已经达到它的最大需求，当它运行完毕后可是放这 3 台磁带机，这就保证了其他两个进程也可顺利执行完毕。因此当 $N < 4$ 时，系统没有死锁的危险。

当 $N = 4$ 时，假设 4 个进程都得到两个资源，此时系统中已没有剩余资源，而 4 个进程都没有到达它们的最大需求，所以系统有可能产生死锁。同理，当 $N > 4$ 时，也有产生死锁的危险。

5.

- (1) 先来先服务 (FCFS)

平均周转时间 $T = (3+7+9+12+12) / 5 = 43/5 = 8.6$

平均带全周转时间 $W = (1+1.17+2.25+2.4+6) / 5 = 12.82/5 = 2.56$

- (2) 采用时间片轮转 (时间片 $q=1$)

平均周转时间 $T = (4+16+13+14+7) / 5 = 54/5 = 10.8$

平均带权周转时间 $W = (1.33+2.67+3.25+2.8+3) / 5 = 13.5/5 = 2.71$

- (3) 短进程优先 (SPN)

平均周转时间 $T = (3+7+11+14+3) / 5 = 38/5 = 7.6$

平均带权周转时间 $W = (1+1.17+2.75+2.8+1) / 5 = 38/5 = 7.6$

- (4) 采用最短剩余时间 (SRT, 时间片 $q=1$)

平均周转时间 $T = (3+18+4+9+2) / 5 = 36/5 = 7.2$

平均带权周转时间 $W = (1+3+1+1.8+1) / 5 = 7.8/5 = 1.56$

- (5) 采用响应比高者优先 (HRRN)

平均周转时间 $T = (3+7+9+14+7) / 5 = 40/5 = 8$

平均带全周转时间 $W = (1+1.17+2.25+2.8+3) / 5 = 10.72/5 = 2.14$

- (6) 采用多级反馈队列 (MFQ, 第 1 个队列的时间片为 1, 第 $i (i > 1)$ 个队列的时间片 $q = 2^{(i-1)}$)

平均周转时间 $T = (3+15+14+14+6) / 5 = 52/5 = 10.4$

平均带权周转时间 $W = (1+2.5+3.5+2.8+3) / 5 = 12.8/5 = 2.56$

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/048052026107006044>