**第2章 习题答案**

**2-9.**

1. x<=3 运行顺序为 Px，P3，P5，P6，P9

T=(x+(x+3)+(x+3+5)+(x+3+5+6)+(x+3+5+6+9))/5=x+9.6

1. 3<x<=5 运行顺序为 P3，Px，P5，P6，P9

T=(3+(3+x)+(3+x+5)+(3+x+5+6)+(3+x+5+6+9))/5=0.8x+10.2

1. 5<x<=6 T=0.6x+11.2
2. 6<x<=9 T=0.4x+12.4
3. 9<x T=0.2x+14.2

**2-12**．计算采用FCFS、SJN、RHN的平均周转时间和平均带权周转时间：

作业号 提交时刻 估计运行 开始运行时刻 完成时刻

时间 FCFS SJN RHN FCFS SJN RHN

1 8.00 2.0 8.0 8.0 8.0 10.0 10.0 10.0

2 9.00 1.2 10.0 10.8 10.5 11.2 12.0 11.7

3 9.50 0.5 11.2 10.0 10.0 11.7 10.5 10.5

4 10.20 0.3 11.7 10.5 11.7 12.0 10.8 12.0

2.05/3.307 1.65/1.875 1.875/2.8125

1. **FCFS 作业运行顺序：1，2，3，4**

各作业的周转时间Ti和平均周转时间T：

T1=10.0-8.00=2.0 T2=11.2-9.00=2.2

T3=11.7-9.5=2.2 T4=12.0-10.2=1.8

T=(T1+T2+T3+T4)/4=(2.0+2.2+2.2+1.8)/4=8.2/4=2.05

各个作业的平均带权周转时间W计算如下：

W=(2/2+2.2/1.2+2.2/0.5+1.8/0.3)/4=(1+1.83+4.4+6)/4=3.307

**2) SJN 作业运行顺序：1，3，4，2**

T1=10.0-8.00=2.0 T2=12-9.00=3

T3=10.5-9.5=1.0 T4=10.8-10.2=0.6

T=(T1+T2+T3+T4)/4=(2.0+3.0+1.0+0.6)/4=6.6/4=1.65

各个作业的平均带权周转时间W计算如下：

W=(2/2+3/1.2+1/0.5+0.6/0.3)/4=1.875

**3) HRN 作业运行顺序：1，3，2，4**

先选择作业**1**  从8.00-------10.00。当作业1完成时，究竟选谁运行，只有通过计算,选择响应比高者运行：

作业2的响应比=((10-9.0) +1.2)/1.2=1.83

作业3的响应比=((10-9.5)+0.5) /0.5=2.0

作业4还未到，只能选作业**3**运行。

作业3运行到10.5结束，再计算剩余的作业2和4：

作业2的响应比=（（10.5-9.0）+1.2）/1.2=2.25

作业4的响应比=((10.5-10.2)+0.3) /0.3=2 选作业**2**运行。

作业2到11.7完成。最后运行作业**4**。运行到12.0,全部结束。

各个作业的周转时间计算如下：

t1=2 t2=11.7-9=2.7 t3=10.5-9.5=1 t4=12-10.2=1.8

各个作业的平均周转时间计算如下：

T==(2+2.7+1+1.8)/4=1.875

各个作业的平均带权周转时间计算如下：

W=(2/2+2.7/1.2+1/0.5+1.8/0.3)/4=2.8125

**2-13．** 已知作业A,B,C,D,E需要的运行时间分别为10,6,2,4,8分钟，优先级分别为3,5,2,1,4。

**（1）轮转法（假定时间片=2分钟）**

**作业完成的顺序为C,D,B,E,A**

开始作业轮转一周需10分钟，

作业C的周转时间：Tc=10分钟 （6分）

C完成后，剩下四个作业，轮转一周需8分钟,

作业D的周转时间：Td=10+8×(4-2)/2=18分钟（16分）

D完成后，剩下三个作业，轮转一周需6分钟,

作业B的周转时间：Tb=18+6×(6-2-2)/2=24分钟（22分）

B完成后，剩下两个作业，轮转一周需4分钟,

作业E的周转时间：Te=24+4=28分钟（28分）

E完成后，只剩下作业A,

作业A的周转时间：Ta=28+2=30分钟（30分）

平均周转时间： T=(10+18+24+28+30)/5=22分（20.4分）

**（2）优先级调度法**

**作业完成顺序为：B,E,A,C,D**

Tb=6分，Te=6+8=14分，Ta=14+10=24分，Tc=24+2=26分，

Td=26+4=30分。

平均周转时间： T=(6+14+24+26+30)/5=20分

**第3章 答案**

**3-7.** 系统中有n+1个进程。其中A1、A2、**…**、An分别通过缓冲区向进程B发送消息。相互之间的制约关系为：发送进程A1、A2、**…**、An要互斥地向BUF中送消息，当接收进程B还未将消息接收完之前，任何一个发送不能再送。同样，B不能重复接收同一个消息。 为此，应设置两个信号量s1和s2。设系统只有容纳一个消息的缓冲区，用信号量s1表示，其初值为1，它用来制约发送进程。信号量s2用来制约接收进程，其初值为0。

BUF

现可用PV操作描述如下：

**int s1=1; s2=0;**

**进程A1、…、An执行的过程为：** **进程B执行的过程为：**

begin begin

准备消息 P(S2)

P(s1) 从缓冲区BUF取消息

将消息送入BUF V(s1)

V(s2) 消耗消息

end end

**若缓冲区容量为m个，这个问题就变为生产者和消费者问题。**

**3-8.**  首先这里的IN和OUT分别表示读写指针，而不是信号量。在系统初启时，环行缓冲区为空，此时IN和OUT都初始化为0。当并发进程通过环行缓冲区通信时，写进程不断地写，读进程不断地读，使得读写指针不断变化。写进程的速度太快，缓冲区会满；读进程的速度太快，缓冲区会空。

**已知循环缓冲区的容量为100。则**

**当（IN+1）%100 = OUT时，说明缓冲区已满。**

**当IN = OUT时，说明缓冲区已空。**

初始化时，IN=OUT=0。一段时间以后：

B1 B2 B3 B4

OUT IN

**3-9.** 为描述阅览室，用一个登记表来记录使用情况。表中共有100项。每当有读者进入阅览室时，为了正确地登记，各读者应互斥使用。为此设两个信号量。mutex:互斥信号量，用来制约各读者互斥地进行登记，其初值为1；empty资源信号量，用来制约各读者能同时进入阅览室的数量，初值为100。

为了正确地描述读者的动作，我们可以**将读者看成进程**。

**int empty=100, mutex=1;**

**读者进程：**

**......**

**p(empty);**

**p(mutex);**

找到一个登记项登记

**v(mutex);**

进入阅览室阅读

准备退出

**p(mutex);**

找到自己的登记项擦除

**v(mutex);**

**v(empty);**

**......**

**3-14.**

证明：假定会死锁，则根据死锁定义，N个进程之间相互等待，至少需要N个单位资源，又系统M个资源已分完，故所有进程需求总和大于或等于M+N，这与题中的所有进程需求总和小于M+N矛盾，故假设不成立。因此，在这种情况下不会死锁。

**3-15．**

**M1:**

**……**

**V(s12);**

**V(s13);**

**V(s14);**

**M2:**

**P(s12);**

**……**

**V(s26);**

**……**

**M3:**

**P(s13);**

**……**

**V(s36);**

**V(s38);**

**M4:**

**P(s14);**

**……**

**V(s47);**

**……**

**附加：m个同类资源，n个进程，每个进程对资源的最大需求量：**

**当m>n时，每个进程最多可以请求个该类资源**

**当m=n时，每个进程最多可以请求1个该类资源**

**当m<n时，每个进程最多可以请求1个该类资源**

（当m>n时，每个进程最多可以请求(m+n-1)/n个该类资源）

**3-16.** 叉子是临界资源，在一段时间内只允许一个哲学家使用。一个信号量表示一把叉子，五个信号量构成信号量数组，这些信号量的初值为1。

**int mutex=1; （int mutex=4;）**

**int fork[0]=fork[1]=…=fork[4]=1;**

**第i个哲学家所执行的程序：**

**do{**

**P(mutex);**

**P(fork[i]);**

**P(fork[(i+1)mod5]);**

**V(mutex);**

**吃饭**

**V(fork[i]);**

**V(fork[(i+1)mod5]);**

**} while(1)；**

**3-17.**

**（1）公平竞争（无写者时，读者仍遵循多个读者可以同时读）**

**rmutex互斥共享readcount; rwmutex读写互斥，写写互斥；**

**读写进程在z上排队。**

**int rmutex=1,rwmutex=1,readcount=0;**

**reader:**

**begin**

**p(z); //读写进程在z上排队。**

**p(rmutex);**

**if(readcount=0) then p(rwmutex);**

**end if**

**++readcount;**

**v(rmutex);**

**v(z); //无写者时，多个读者可以同时读.**

**read data;**

**p(rmutex);**

写

z

读

写

写

读

读

读

写

**--readcount;**

**if(readcount=0 then v(rwmutex);**

**end if;**

**v(rmutex);**

**…**

**end**

**writer:**

**begin**

**p(z); //读写进程在z上排队。**

**p(rwmutex);**

**write data;**

**v(rwmutex);**

**v(z);**

**…**

**end**

**（2）写者优先**

int readcount,writecount;

semaphore rmutex=1,wmutex=1,rwmutex=1,z=1,x=1;

reader：

**//当来了一个写进程时，通过p(x)禁止其后读进程读，直到写进程写完为止。**

**while(1){**

**p(z); //其他读进程在z上排队**

**p(x); //一个读进程与一个写进程在x上竞争**

**p(rmutex); //读进程互斥访问readcount**

**++readcount;**

**if(readcount==1) p(rwmutex);**

**v(rmutex);**

**v(x);**

**v(z);**

rwmutex

x

z

读

读

读

读

写

读

写

写

read data; **//临界区**

**p(rmutex);**

**--readcount;**

**if(readcount==0) v(rwmutex);**

**v(rmutex);**

**}**

Writer：

**while(1){**

**p(wmutex); //写进程互斥访问writecount**

**++writecount;**

**if(writecount==1) p(x); //一个写进程与一个读进程在x上竞争**

**v(wmutex);**

**p(rwmutex); //其他写进程在rwmutex上排队**

write data; **//临界区**

**v(rwmutex);**

**p(wmutex);**

**--writecount;**

**if(writecount==0) v(x); //写进程都写完时，通过v(x)允许读进程读**

**v(wmutex);**

**}**

**附加题：**

**读者写者，规定仅允许5个进程同时读，怎样修改程序？**

**解：增加一个资源信号量s，初值为5。**

**int s=5;**

**Reader：**

**begin**

**P(rmutex);**

**readcount=readcount+1;**

**if(readcount==1)then P(rwmutex);**

**V(rmutex);**

**P(s);**

**read\_file();**

**V(s);**

**P(rmutex);**

**readcount=readcount-1;**

**if(readcount==0)then V(rwmutex);**

**V(rmutex);**

**end**

**writer:**

**begin**

**p(rwmutex);**

**write data;**

**v(rwmutex);**

**…**

**end**

**3-18．**

**方法一： int s1=0, s2=n;**

**顾客进程：**

**P(s2);**

**V(s1);**

**坐椅子等理发**

**理发师进程：**

**P(s1);**

**给顾客理发**

**V(s2)**

**方法二：int customers=0, barber=0, mutex=1, waiting=0;**

**顾客进程： 理发师进程：**

**P(mutex); P(customers);**

**if(waiting<n) V(barber);**

**{ P(mutex);**

**waiting++; waiting--;**

**V(mutex); V(mutex);**

**V(customers); 理发；**

**P(barber);**

**理发**

**}**

**else V(mutex);**

**3-19．（2）和（4）会发生死锁。**

**3-20．**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **P1/剩余** | **P2/剩余** | **P3/剩余** | **系统剩余** |
| **1** | **3/5** |  |  | **7** |
| **2** |  | **2/4** |  | **5** |
| **3** |  |  | **4（不安全）** |  |
| **4** | **(3+2)/3** |  |  | **3** |
| **5** |  | **2（不安全）** |  |  |
| **6** | **(3+2+3)/0** |  |  | **0(8)** |
| **7** |  |  | **4/3** | **4** |
| **8** |  | **(2+2)/2** |  | **2** |
| **9** |  |  |  |  |

（1） P1占有5个资源，剩余3个资源请求。运行态。

P2占有2个资源，剩余4个资源请求。阻塞态。

P3占有0个资源，剩余7个资源请求。阻塞态。

系统剩余3个资源。

（2）P1的请求最先满足。进程完成序列：P1，P2，P3。

**3-21．**

（1）

**最大需求矩阵**： **分配矩阵**： **剩余请求矩阵**：

0 0 0 0

0 7 5 0

1 0 0 2

0 0 2 0

0 6 4 2

0 0 1 2

1 0 0 0

1 3 5 4

0 6 3 2

0 0 1 4

0 0 1 2

1 7 5 0

2 3 5 6

0 6 5 2

0 6 5 6

Max = Allocation = Need =

**剩余资源向量**：Available=(1 5 0 2)

（2）当前系统是安全的。

判断系统是否安全，只要检查系统剩余资源向量能否对各进程的剩余请求向量找到一个进程完成序列，当按照这个序列为各进程分配资源时，各进程都能成功完成。若能找到，则系统是安全的，否则，为不安全。

先找到p0, 因为p0已满足最大资源请求，它可以完成，释放其占有的资源，使系统剩余资源向量为（1 5 1 4）

之后，系统剩余资源向量（1 5 1 4），可满足进程p2, 使p2 可以完成，释放其占有的资源，使系统剩余资源向量为（2 8 6 8）。

之后无论选哪一个进程都可成功完成。

故找到的进程完成序列可为：p0,p2,p4,p3,p1; 或 p0,p2,p3,p1,p4 等，故系统是安全的。

（3）因系统剩余可用向量为（1502），p2的剩余请求向量为（1002），即（1502）>（1002）。故，当p2提出（1001）请求时，能满足。进程完成序列：p0,p2,p4,p3,p1

**第4 章 答案**

**4-14** 内存有如下顺序排列的空闲块：10K，40K，20K，18K，7K，9K，12K和15K，有如下的请求序列：12K，10K，9K。

**（1）若采用首次适应法**：

* **12K的请求**：将分配**40K**的空闲块， 40K变为剩余的（40-12）K=28K，空闲队列变为：10K，28K，20K，18K，7K，9K，12K和15K；
* **10K的请求**：将分配**10K**的空闲块，空闲队列变为：28K，20K，18K，7K，9K，12K和15K；
* **9K的请求**：将分配**28K**的空闲块，空闲队列变为：（28-9）=18K，20K，18K，7K，9K，12K和15K；

**（2）若采用最佳适应法**：

* **12K的请求**：将分配**12K**的空闲块，空闲队列变为：10K，40K，20K，18K，7K，9K和15K；
* **10K的请求**：将分配**10K**的空闲块，空闲队列变为：40K，20K，18K，7K，9K，12K和15K；
* **9K的请求**：将分配**9K**的空闲块，空闲队列变为： 40K，20K，18K，7K， 12K和15K；

**（3）若采用最坏适应法**：

* **12K的请求**，将分配**40K**的空闲块，空闲队列变为：10K，28K，20K，18K，7K，9K和15K；
* **10K的请求**：将分配**28K**的空闲块，空闲队列变为： 20K，18K，7K，9K，12K和15K；
* **9K的请求**：将分配**20K**的空闲块，空闲队列变为：10K，18K，11K， 18K，7K， 12K和15K。

**4-15**  有如下图所示的页表中的虚地址与物理地址之间的关系，即该进程分得6个内存块。页的大小为4096。给出对应下面虚地址的物理地址：(1)20； (2) 5100； (3) 8300； (4) 47000.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 2 | 1 | 6 | 0 | 4 | 3 | x | x |

解：

（1）虚地址 20变为页号0 和页内偏移20

由页号查页表得0页对应内存块号为2 ，可计算得

物理地址=块号\*页的大小+页内偏移=2\*4096+20=8212

（2）虚地址 5100变为页号1 和页内偏移1004（5100/4096）

由页号查页表得1页对应内存块号为1 ，可计算得

物理地址=块号\*页的大小+页内偏移=1\*4096+1004=5100

（3）虚地址 8300变为页号2 和页内偏移108

由页号查页表得2页对应内存块号为6 ，可计算得

物理地址=块号\*页的大小+页内偏移=6\*4096+108=24684

（4）虚地址 47000变为页号11 和页内偏移1944

11＞7 页号越界

**4-16**一个作业在执行过程中，按如下顺序依次访问各页，作业分得四个主存块，问分别采用FIFO、LRU和OPT算法时，要产生多少次缺页中断?设进程开始运行时，主存没有页面。

页访问串顺序为：0 1 7 2 3 2 7 1 0 3 2 5 1 7

**（1）FIFO**

0 1 7 2 3 2 7 1 0 3 2 5 1 7

**0** 1 7 2 3 3 3 3 0 0 0 5 1 7

0 1 7 **2** 2 2 2 3 3 3 0 5 1

0 1 7 7 7 7 **2** 2 2 3 0 5

0 1 1 1 1 7 7 7 2 3 0

F F F F F S S S F S S F F F

采用FIFO淘汰算法，产生**9次**缺页中断。

**（2）LRU**

0 1 7 2 3 2 7 1 0 3 2 5 1 7

**0 1 7 2 3** 2 7 1 **0 3 2 5 1 7**

0 1 7 2 3 2 7 1 0 3 2 5 1

0 1 7 7 3 2 7 1 0 3 2 5

0 1 1 1 3 2 7 1 0 3 2

F F F F F S S S F F F F F F

采用LRU算法时，产生**11次**缺页中断。

**4-17** 考虑如图所示的段表，给出如下所示的逻辑地址所对应的物理地址。

|  |  |
| --- | --- |
| 段始址 | 段的长度 |
| 219 | 600 |
| 2300 | 14 |
| 92 | 100 |
| 1326 | 580 |
| 1954 | 96 |

（1）0，430 🡪 219+430=649

（2）1，10 🡪 2300+10=2310

（3）2，500 🡪 500＞100段内地址越界

（4）3，400 🡪 1326+400=1726

（5）4，112 🡪 112＞96 段内地址越界

**4-18**  一台计算机含有65536字节的存储空间，这一空间被分成许多长度为4096字节的页。有一程序，其代码段为32768字节，数据段为16386字节，栈段为15870字节。试问该机器的主存空间适合这个作业吗?如果每页改成512字节，适合吗?

**答**：

（1）

存储空间每块为4096个字节，共可分成16块。

程序代码段占32768/4096=8块，数据段占16386/4096=5块，栈段占15870/4096=4块，合计为8+5+4=17块，故该机器的主存空间不适合这个作业。

（2）

当存储空间每块为512个字节，共可分成128块。

程序代码段占32768/512=64块，数据段占16386/512=33块，栈段占15870/512=31块，合计为64+33+31=128块，故该机器的主存空间是适合这个作业的。

**4-19 逻辑地址中，用9位表示页号，用10位表示页内地址。**

**4-20 （1）缺页中断50次； 5000次**

**（2）缺页中断100次； 10000次**

**4-21 0.9×(0.75×1+0.25×8)+0.10×(8+5000+8)+8=512.075**

**4-23 8192/4=2048**

**64=7+11+11+11+11+13 5级页表**

1. **文件系统**

**9.** 文件存贮空间管理可采用成组自由块链表或位示图。若一磁盘有B个盘块，其中有F个自由块。**若盘块号用D位表示**。试给出使用自由块链表比使用位示图占用更少的空间的条件。当D为16时，给出满足条件的自由空间占整个空间的百分比。

解：一磁盘有B个盘块，用位图表示要使用B位

现有F个自由块，若表示一个盘块需用D位。则采用链表接连F个盘块，需要F个链指针，共占F\*D位。使用自由块链表比使用位示图占用更少的空间的条件是F\*D<B。

当D=16时，满足条件的自由空间占整个空间的百分比为

F/B<1/16=6.25%。

**10**. 文件系统的执行速度依赖于缓冲池中找到盘块的比率。假设盘块从缓冲池读出用1毫秒，从盘上读出用40毫秒。从缓冲池找到盘块的比率为n，请给出一个公式计算读盘块的平均时间，并画出n从0到1.0的函数图像。

解：读一个盘块的平均时间=n+40(1-n)=40-39n

40

1

0 1

n

**13.**

1574/256=6……38， 因此，要访问文件的第7个记录内的38B处。每个块放两个记录，所要访问的字节处在第4个逻辑块内，其对应的物理块号为4，应访问4号块内的第38个字节。要访问4次磁盘才能将该字节的内容读出。

**14．**共需要4次磁盘操作

**15**．1GB=230 ，16KB=214 ， 230/214 =216 ，每个磁盘块号需要2个字节表示，即2B。

（1）10KB<16KB, 所以，只占用1个磁盘块。

（2）1089KB/16KB=69 需一个索引块和69个数据块，共70个盘块。

（3）129MB/16KB=8256 ， 16KB/2B=8K（个索引项）<8256

所以，需2个一级索引表和一个1个二级索引表，8256个数据块。

共需8259个磁盘块。

**第6章 设备管理**

**6．下列工作各是在四层I/O软件的哪一层上实现的?**

(1) 对于读磁盘，计算柱面、磁头和扇区（设备驱动）

(2) 维持最近所用块而设的高速缓冲（独立于设备的软件层）

(3) 向设备寄存器写命令（设备驱动）

(4) 查看是否允许用户使用设备（独立于设备的软件层）

(5) 为了打印，把二进制整数转换成ASCII码（用户进程）

**13．假设移动头磁盘有200个磁道(从0号到199号)。目前正在处理143**号磁道上的请求，而刚刚处理结束的请求是125号，如果下面给出的是按到达时间的先后排成的等待服务队列：86，147，91，177，94，150，102，75，130。那么，用下列各种磁盘调度算法来满足这些请求所需的总磁头移动量是多少?

1. **FCFS**：125 143--86--147--91--177--94--150--102--75--130

满足这些请求所需的总磁头移动量=（143-86）+（147-86）+（147-91）+（177-91）+（177-94）+（150-94）+（150-102）+（102-75）+（130-75）=57+61+56+86+83+56+48+27+55=524

1. **SSTF**：125 143--147--150--130--102--94--91--86--75--177

满足这些请求所需的总磁头移动量

=(150-143)+(150-75)+(177-75)=7+75+102=182

(3)**SCAN**：125 143--147--150--177--**199**--130--102--94--91--86--75

满足这些请求所需的总磁头移动量= (199-143)+(199-75)=56+124=180

(5)**C-SCAN**：125 143--147--150--177--**199**-**-0-**-75--86--91-94—102 --130满足这些请求所需的总磁头移动量=(199-143)+（199-0）+（130-0）=56+199+130=385

(4)**C-LOOK**：125 143--147--150--177--75--86--91--94--102--130

满足这些请求所需的总磁头移动量

=(177-143)+(177-75)+(130-75)=33+102+55=190