**第一章**

一.填空题

1.处理机管理(或进程管理)、内存管理、设备管理、文件管理

2.命令接口、程序接口

3.并发、共享、虚拟、异步

4.独立性、交互性

5.并行、交替

二.选择题

1-5：ADABA 6-10：ACDBB 11-15：ADACC

三.判断题

1.错（2分），多道系统不一定是多用户系统。桌面型Windows（也可以说是 非服务器版本）就是多道系统，但不是多用户系统。（3分）

四.问答题

1.主要从分布性、健壮性、透明性、统一性等四个方面来区别。

2.能够充分利用CPU资源，提高系统的吞吐量。

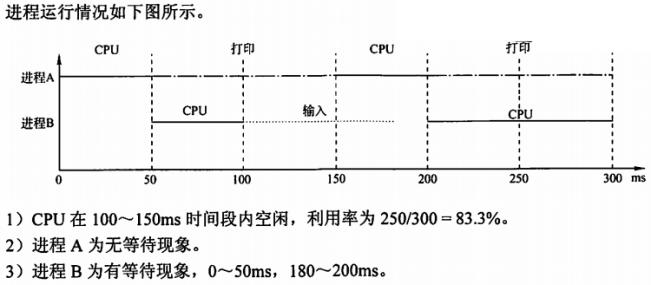
3.缓和CPU和I/O设备之间的速度不匹配；减少对CPU的中断频率；提高CPU和I/O设备之间的并行性；解决数据粒度不匹配的问题。

4.错，不一定

5.回答要点:经过一段时间的使用后，导致磁盘分散存储，空闲区域碎片化，通过碎片整理，将磁盘紧缩，把空闲区域挨在一起，在存储文件时将文件分配在连续的存续存储区域，加快读写速度。

6.现代操作系统设计目标包含用户态、内核态的双模式，进程运行在用户态，不能直接访问敏感资源，需要通过系统调用让操作系统内核代为访问:采用各种陷阱机制，限制进程的行为，同时又不影响性能:采用分时机制，让进程宏观上共享CPU:采用分页内存管理机制，让进程之间拥有独立的逻辑地址空间

三.综合题



**第二章**

一.填空题

1.进程

2.进程调度，静止就绪

3.减1，<0

4.阻塞，就绪

5.P操作，V操作

6.I/O请求、时间片到、申请资源不成功、系统进程无事可做、剥夺算法中高优先级进程的进入就绪状态。（选两个）

7.2

8.进程，线程（不能调换）

9.8

二.选择题

1-5：DBBAB 6-10：CCBDA 11-15：BABCD 16-20：DBDCC 21-25：AB

三.判断题

1.对（2分），因为信号量是共享变量，在用于同步互斥时需要检测并修改信号量，逐个时候，需要互斥操作。（3分）

四.问答题

1.设置信号量的初值为0，先执行的在其结尾加V操作，后执行的在其前加P操作。

2.链接方式和索引方式。

3.内核态、用户态；各种陷阱；系统调用；分页内存管理，独立的逻辑地址空间

4.信号量有值而条件变量无值：执行wait操作时，信号量会先减少值，小于0才会阻塞进程，而条件变量则直接阻塞进程；执行signal操作时，信号量会增加值，当小于等于0时唤醒一个进程，而条件变量直接唤醒一个阻塞的进程。

四.综合题

1.int empty1 = 1;

int empty2 = 1;

int full1 = 0;

int full2 = 0；

main( )

{

cobegin

PA( );

PB( );

PC( );

coend

}

PA( )

{

while(1)

{

从磁盘读一个记录;

p(empty1);

将记录存入缓冲区1；

v(full1);

}

}

PB( )

{

while(1)

{

p(full1);

从缓冲区1中取出记录;

v(empty1);

p(empty2);

将记录存入缓冲区2；

v(full2);

}

}

PC( )

{

while(1)

{

p(full2);

从缓冲区2中取出记录;

v(empty2);

打印记录;

}

}

2.设信号量empty，初值为2；

mutex，初值为1；

apple，初值为0；

orange，初值为0。

Parbegin

Father：begin

L1：P(empty)；

P(mutex)；

放苹果；

V(mutex)；

V(apple)；

GoTo L1；

End；

Mather：begin

L2：P(empty)；

P(mutex)；

放橘子；

V(mutex)；

V(orange)；

GoTo L2；

End；

Daughter：begin

L3：P(apple)；

P(mutex)；

放苹果；

V(mutex)；

V(empty)；

GoTo L3；

End；

Sun：begin

L4：P(orange)；

P(mutex)；

放橘子；

V(mutex)；

V(empty)；

GoTo L4；

End；

Parend.

3.该题是生产者/消费者问题的变形，可认为一个消费者（装配工人）同两个生产者（A、B车间）互斥使用两个缓冲区（F1、F2），可设mutex1，mutex2（初值为1）控制进程对F1、F2的互斥操作，另设empty1、empty2（初值均为10），full1，full2（初值均为0）。过程如下：

parbegin:

A车间：begin

L1：生产一个产品：

P (empty1);

P (mutex1);

放入F1;

V (mutex1);

V (full1);

Goto L1;

End;

B车间：begin

L2：生产一个产品：

P (empty2);

P (mutex2);

放入F2;

V (mutex2);

V (full2);

Goto L2;

End；

装配工人：begin

L3：P (full1);

P (full2);

P (mutex1);

P (mutex2);

取A和B;

V (mutex1);

V (mutex2);

V (empty1);

V (empty2);

Goto L3;

End;

Parend.

4.设置信号量 S1,S2,S3,初值都为0

第1组 第2组 第3组 第4组

起跑 P(S1) P(S2) P(S3)

100m 起跑 起跑 起跑

V（S1） 100M 100M 100M

V(S2) V(S3)

5.

int increaseAndGet(int \*a){

int t1,t2;

do {

t1=\*a;

t2=t1+1;

} while (!cas(a, t1, t2));

return t2;

}

6.a)readCount：0 mutex：1 writeBlock：1

b）第1个读者进程会阻塞在第5行，其他2个读者进程会阻塞在第2行。

c） 写者进程会阻塞在第2行

d） 会发生写者饥饿现象。当有读者进程在读取数据，有写者进程阻塞。如果不断有读者进程来到就会先于写者进入临界区读取数据，写者进程可能会无限期延迟。

7. 设置信号量 S 用于爸妈会小孩之间的同步，代表2000元钱，初值为0

设置信号量 F和M用于爸妈之间的前驱关系，F初值为1，M初值为0,mutex初值为1，用于存取互斥。

爸爸 妈妈 小孩

P(F) P(M) P(S)

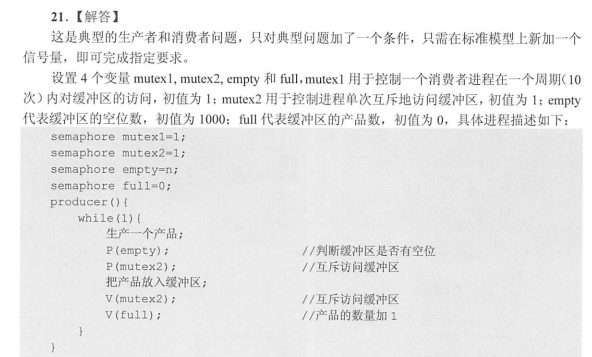
P(MUTEX) P(MUTEX) P(MUTEX)

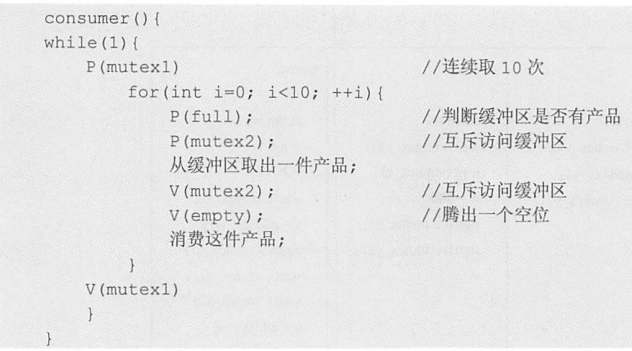
存钱 存钱 存钱

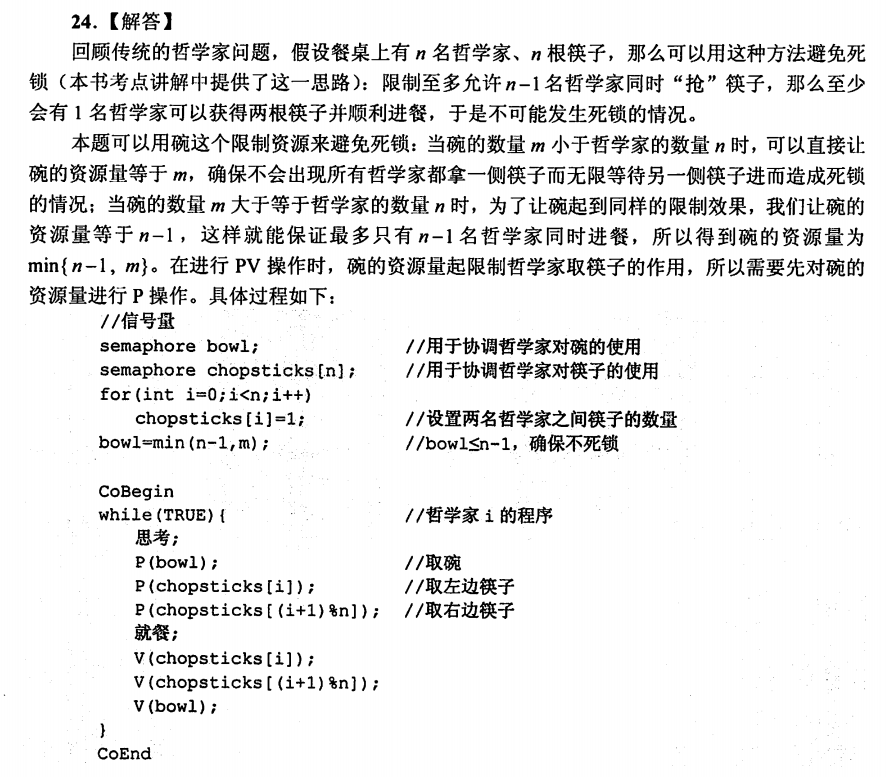
V(MUTEX) V(MUTEX) V(MUTEX)

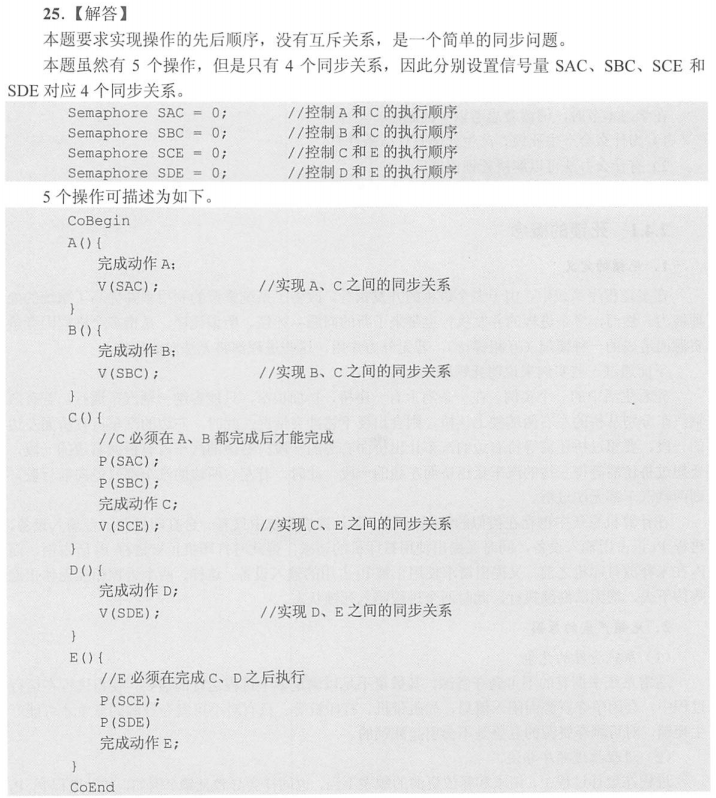
V(S) V(S)

V(M) V(F)

8.



9.

10.

11. (10分)

（1）P (empty) 和P (full)应放在 P (mutex) 之前，否则会出现死锁。(5分)

（2）把互斥信号量mutex换成2个,mutex-in和mutex-out.（2分）

（程序写对，3分）

**var mutex-in,mutex-out,empty,full:semaphore:=1,1,n,o ；**

**Buffer : array [0……n-1] of message ；**

**in, out : o……n-1:=0,0 ；**

**begin**

**parbegin**

**生产者: begin**

**消费者: begin**

**repeat**

**P (full) ；**

**P (mutex-out) ；**

**m := buffer[out] ；**

**Out : = (out+1) mod n ；**

**V (mutex-out) ；**

**V (empty) ；**

**Consume message m ；**

**until false**

**end**

**parend**

**end**

**repeat**

**Produce a new message m ；**

**P (empty) ；**

**P (mutex-in) ；**

**Buffer[in]=m ；**

**in :=(in+1) mod n ；**

**V (mutex-in) ；**

**V (full) ；**

**until false**

**end**

**第三章**

一.填空题

1.互斥

2.死锁的避免、死锁的预防、死锁的解除 （顺序不可交换）

3.循环等待链

4.PCB

5.内存

6.优先级

7.短进程优先

8.短作业优先、时间片轮换、响应比高者优先

9.预防、避免

10.请求和保持条件（或者说 逐次请求），循环等待连（可调换）

11.请求和保持条件（或者说 逐次请求），不可抢占（可调换）

二.选择题

1-5：CDDBD 6-10：CAAAA 11-15:BBBCC 16-20:DA

三.判断题

1.错（2分）。不一定，当各进程推进速度达到某种巧合，满足了死锁的4个必要条件时才会出现死锁。（3分）

四.问答题

1.不能

2.不是

3.将进程在运行中所需要的临界资源全部一次性分配给进程，等进程用完后再全部一起释放。

造成资源的利用率低。

4.对

5.对，安全状态存在安全序列，进程按这个序列执行肯定不会发生死锁。

6.互斥、非抢占、保持和请求、循环等待链

7.现运行进程阻塞、现运行进程结束、现运行进程时间片到、现运行进程挂起、现运行进程被抢占

8.略

五.综合题

1.FCFS: 作业号 提交时间 执行时间 开始时间 完成时间 周转时间 带权周转时间

1 10.0 2.0 10.0 12.0 2.0 1.0

2 10.2 1.0 12.0 13.0 2.8 2.8

3 10.4 0.5 13.0 13.5 3.1 6.2

4 10.5 0.3 13.5 13.8 3.3 11.0

平均周转时间 = (2.0+2.8+3.1+3.3) / 4 = 2.8 (小时)

平均带权周转时间 = (1.0+2.8+6.2+11) / 4 = 5.25

SJF: 作业号 提交时间 执行时间 开始时间 完成时间 周转时间 带权周转时间

1 10.0 2.0 10.0 12.0 2.0 1.0

4 10.5 0.3 12.0 12.3 1.8 6.0

3 10.4 0.5 12.3 12.8 2.4 4.8

2 10.2 1.0 12.8 13.8 3.6 3.6

平均周转时间 = (2.0+1.8+2.4+3.6) /4 = 2.45(小时)

平均带权周转时间 = (1+6+4.8+3.6) /4 = 3.85

2.（1）存在一个安全序列{P5，P4，P3，P2，P1}，故该状态是安全的。

（2）因请求资源数（0，3，4）> 剩余资源数（2，2，3），所以不能分配。

（3）存在一个安全序列{P4，P5，P3，P2，P1}，故该状态是安全的，可以将P4申请的资源分配给它。

（4）若分配，可能资源（0，1，2）已不能满足任何进程的资源需求，系统将进入不安全状态。此时系统不能将资源分配给P1。

3.(1) P1申请了R4,P2申请到R1,P3申请到R2,P4申请到R3

(2)一次性分配或按序分配。可以分别破坏逐次申请和循环等待连

(3) 设置信号量 R1、R2、R3、R4，初始值都为1，采用按序分配

P1: P2: P3: P4

P(R1) P(R1) P(R2) P(R3)

P(R4) P(R2) P(R3) P(R4)

… … … …

V(R4) V(R2) V(R3) V(R4)

V(R1) V(R1) V(R2) V(R3)

4.(1) 第1次，A响应比=1+70/90=1.78, B=1+60/60=2,C=1+30/24=2.25, 所以选 C 运行

9:24,第2次 A=1+94/90=2.04, B=1+84/60=2.4,所以选 B运行

10:24 运行A A=1+154/90=2.71， 11:54结束

A、B、C周转时间分别是244分钟、144分钟、54分钟

（2） C 2.25， B 2.4 A 2.71

5.a）互斥：一段桥面一个时刻只能允许一辆车占用；非抢占：一辆车占用了一段桥面不能被剥夺：占有并等待江右边车占用了右边的桥面，申请左边的桥面；循环等待：左边的车等待右边车释放右边的桥面，反之亦然。

b）互斥：将桥面修宽，双向2车道；非抢占：桥上安装一个吊车，将死锁的车子吊起来；占有并等待：一次只能一辆车上桥

6.a)13.25 b)8.5 c)13.25

7.1）需求矩阵为： 3 4 7

1 3 4

0 0 6

2 2 1

1 1 0

是安全状态，存在安全序列 （P4，P5，P3，P1，P2）(不唯一)

（2）能分配，存在安全序列（P4，P5，P1，P2，P3）(不唯一)

（3）如果基于（2）的基础上，不能分配，因为不存在安全序列。否则，能分配，存在不安全序列。

8. 1)方法1：最多允许4个哲学家同时坐在桌子周围

基于死锁避免策略

semaphore chopstick[5] = {1,1,1,1,1}; //初始化信号量

semaphore eating = 4; //至多只允许四个哲学家可以同时进餐

Pi(){ //i号哲学家的进程

while(1){

think;

P(eating); //请求进餐，若是第五个则挨饿

P(chopstick[i]); //取左边筷子

P(chopstick[(i+1)%5]) ; //取右边筷子

eat;

V(chopstick[(i+1)%5]) ; //放回右边筷子

V(chopstick[i]) ; //放回左边筷子

V(eating); //释放信号量给其他挨饿的哲学家

}

}

方法2：仅当一个哲学家左右两边的筷子都可用时才允许他拿筷子

基于死锁避免策略

semaphore chopstick[5] = {1,1,1,1,1}; //初始化信号量

semaphore mutex=l; //设置取筷子的信号量

Pi(){ //i号哲学家的进程

while(1){

P(mutex); //在取筷子前获得互斥量，一次只能由一个哲学家取筷子

P(chopstick[i]) ; //取左边筷子

P(chopstick[(i+1)%5]); //取右边筷子

V(mutex); //释放取筷子的信号量

eat;

V(chopstick[i]); //放回左边筷子

V(chopstick[(i+1)%5]); //放回右边筷子

think;

}

}

(也可以用AND信号量机制实现)

方法3：给所有哲学家编号，奇数号的哲学家必须首先拿左边的筷子，偶数号的哲学家则反之

基于死锁避免策略

semaphore chopstick[5] = {1,1,1,1,1}; //初始化信号量

Pi(){ //i号哲学家的进程

while(1){

think;

if(i%2==0){

P(chopstick[(i+1)%5]) ; //取右边筷子

P(chopstick[i]); //取左边筷子

eat;

V(chopstick[(i+1)%5]) ; //放回右边筷子

V(chopstick[i]) ; //放回左边筷子

}else{ //奇数哲学家，先左后右

P(chopstick[i]); //取左边筷子

P(chopstick[(i+1)%5]) ; //取右边筷子

V(mutex); //释放互斥量

eat;

V(chopstick[i]) ; //放回左边筷子

V(chopstick[(i+1)%5]) ; //放回右边筷子

}

}

}

2)如果系统进入不安全状态后，不一定会产生死锁。安全状态是指系统可以按照某种顺序分配资源而不会发生死锁。进入不安全状态可能会导致死锁的产生，但也可能不会，这取决于实际的资源请求顺序。

系统处于不安全状态未必死锁，但死锁时一定处于不安全状态。系统处于安全状态一定不会死锁。银行家算法通过防止系统进入不安全状态来避免死锁，因此银行家算法能够在一定程度上解决实际中的死锁问题。

9.（本题12分）：评分标准：每个作业平均3分。

作业名 到达时间 完成时间

-------------------------------------------------------------------------------------

A 10：00 10:40

B 10：20 11:10

C 10：30 12:20

D 10：40 11:30

平均周转时间=（40+50+110+50）/4=62.5

**第四章**

一.填空题

1.动态， 静态

2.分页

3.动态，静态

4.低地址、高地址

5.3

6.顺序存取，顺序存取，直接存取（或 随机存取）（第一个 和后面2个不能互换，后面2个可互换）

7.链接

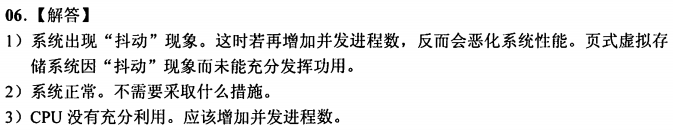
二.选择题

1-5：BCCAC 6-10：DACBB 11-15：CCDDC 16-20:BCBCA 21-25：BCC

三.简答题

1.重定位是指作业装入与其逻辑地址空间不同的物理空间所引起的地址变换过程。

特点是：（1）由硬件实现；（2）在程序运行过程中进行地址变换。

2.

3.由于驻留集大小任意，现要求两种算法的替换页面和缺页情况完全一样，就意味着要求 与LRU的置换选择一致。FIFO替换最早进入主存的页面，LRU替换上次访问以来最久未被访问的页面，这两个页面一致。就是说，最先进入主存的页面在此次缺页之前不能再被访问，这样页面也就同时是最久未被访问的页面。

例如，合法驻留集大小为4时，对访问串1，2，3，4，1，2，5，当5号页面调入主存时，应在1，2. 3，4页中选择一个替换，FIFO选择1，LRU选择3。原因在于1号页面虽然最先进入主存，但由于其进入主存后又被再次访问，所以它不是最久未被访问的页面。若去掉对1号页面的第二次访问，则FIFO与LRU的替换选择就会相同。同理，当5号页而调入主存后，若再访问新的6号页面，则2号页面会遇到同样的问题。因此，以此类推，访问串中的所有页而号都应不同，但要注意到，连续访问相同页面时不影响后面的替换选择，所以对访问串的要求是：不连续的页面号均不相同。

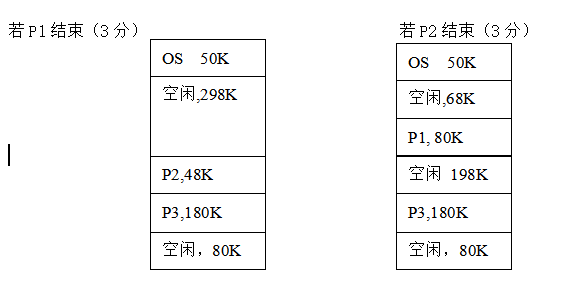
四.综合题

1.该系统虚拟地址空间为字节，页面长4K字节，页表项每项4字节，即每页可放页表项的个数为；最高层页表占1页，该页最多存放页表项个数为；每项指向一页，每页又存放页表项个数为；一次类推，最多可以采用的分页策略的层数为[64/10]=6。

2.（1）22位， （2）4K, (3) 112K,184K,48K,228K, (4)3904 +184K=188720

3.答：a）发生的错误为访问非法内存错误。是由第6行代码引起的。因为每个进程有独立的逻辑地址空间，但必须要向操作系统中请后才能实际访问，不断增加的指针会导致访问到并没有申请的内存单元。b）在Windows和Linux中，都是采用分页内存管理的，内存分配的单位是以页为单位，而一页是4k，所以发生访问非法内存的地方一定发生在访问一个无效页的第1个字节处，所以低12 位一定0.

4.



5.1）页框号为21。因为起始驻留集为空，而0页对应的页框为空闲链表中的第三个空闲页框（21），其对应的页框号为21。

2）页框号为32。理由： 因11>10，因此发生第三轮扫描，页号为1的页框在第二轮已处于

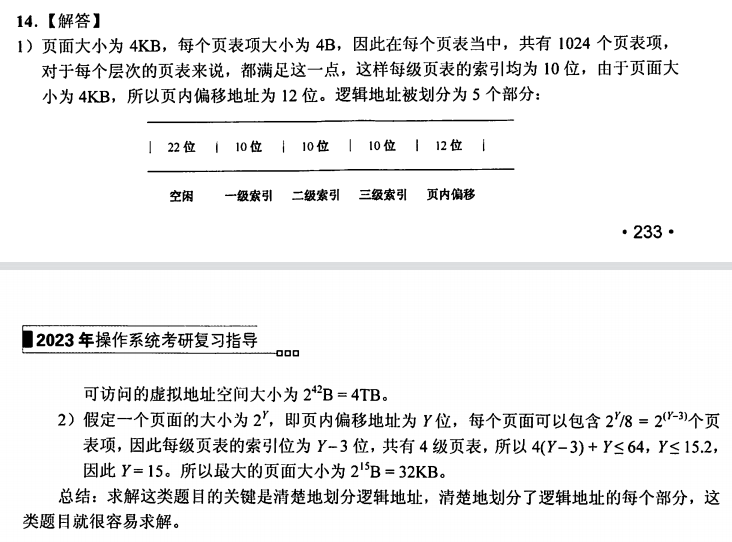
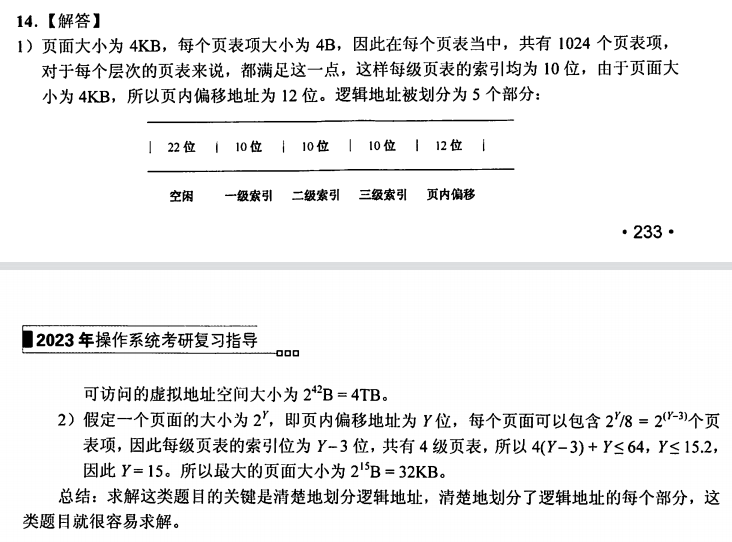
空闲页框链表中，此刻该页又被重新访问，因此应被重新放回驻留集中，其页框号为32。

3）页框号为41。理由：因为第2页从来没有被访问过，它不在驻留集中，因此从空闲页框

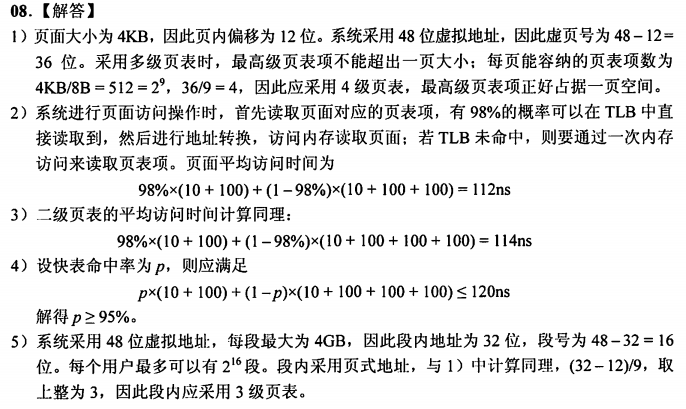
链表中取出链表头的页框41，页框号为41。

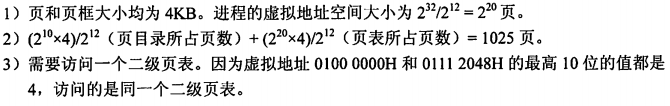
4）合适。理由：程序的时间局部性越好，从空闲页框链表中重新取回的机会越大，该策略

的优势越明显。

6.

7.



8.

**第五章**

一.填空题

1.状态位，修改位

2.离散

3.状态位，修改位

4.物理实体、逻辑上的对应体

5.请求调入， 置换 （可调换）

二.选择题

1-5:B

三.判断题

1.错（2分）。LRU总是选择内存中最近最久没有被访问的页面淘汰。（3分）

四.问答题

1.缺页中断在返回前要对返回地址进行处理，以便发生缺页中断的指令在缺页中断发生后可以重新执行。一般中断不能更改返回地址。

2.请求调页功能和页面置换功能。

页表机制，缺页中断机制，地址变换机构。

3.错，选择最近最长时间没有被访问的页面淘汰

4.系统颠簸是指系统频繁地忙于页面换出换入。产生的原因是系统物理内存不足，导致缺页中断发生频率过高。解决方法是增加系统内存，或者减少并发的进程数量。

五.综合题

1.最佳置换淘汰算法

页面 4 3 2 1 4 3 5 4 3 2 1 5

4 4 4 4 4 2 2

3 3 3 3 3 1

2 1 5 5 5

缺 缺 缺 缺 缺 缺 缺

缺页率： 7/12

LRU算法

页面 4 3 2 1 4 3 5 4 3 2 1 5

4 4 4 1 1 1 5 2 2 2

3 3 3 4 4 4 4 1 1

2 2 2 3 3 3 3 5

缺 缺 缺 缺 缺 缺 缺 缺 缺 缺

缺页率： 10/12

2.对于程序A：缺页次数为50次；

对于程序B：缺页次数为100 \* 50 = 5000次。

3.FIFO算法选择3号页面置换，由于修改过，需要写回外存；LRU算法选择1号页面置换；NRU选择2号页面置换。

4.采用FIFO算法，缺页次数为9次。

采用LRU算法，缺页次数为9次。

5.2362H: 页号P=2，访问快表10ns，因初始为空，访问页表100ns得到块号合成绝对地址后访问内存花100ns，共计10ns+100ns+100ns=210ns。

1565H: 页号P=1，访问快表10ns，落空，访问页表100ns落空，缺页中断处理108ns，合成物理地址后访问内存100ns，共计10ns+100ns+108ns +100ns。

25A5H: 页号P=2，访问快表10ns，命中，化10ns，根据绝对地址访问内存100ns，共计10ns+100ns=110ns。

（2）4分

访问虚地址1565H，产生缺页中断，必须淘汰一页，根据置换算法应淘汰0号页面，因此1565H 对应页的块号为101H，由此得到1565H的绝对地址为101565H。

6.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1\* | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 5\* | 6 | 6 | 6 | 6 | 6 | 6 |
|  |  | 2\* | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
|  |  |  | 3\* | 3 | 3 | 3 | 5\* | 5 | 5 | 5 | 5 | 3\* | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
|  |  |  |  | 4\* | 4 | 4 | 4 | 6\* | 6 | 6 | 6 | 6 | 7\* | 7 | 7 | 7 | 1\* | 1 | 1 | 1 |
|  | 1 | 2 | 3 | 1 | 2 | 1 | 5 | 6 | 2 | 1 | 2 | 3 | 7 | 6 | 3 | 2 | 1 | 2 | 3 | 6 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1\* | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 7\* | 7 | 7 | 7 | 1\* | 1 | 1 | 1 |
|  |  | 2\* | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
|  |  |  | 3\* | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
|  |  |  |  | 4\* | 4 | 4 | 5\* | 6\* | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 |
|  | 1 | 2 | 3 | 1 | 2 | 1 | 5 | 6 | 2 | 1 | 2 | 3 | 7 | 6 | 3 | 2 | 1 | 2 | 3 | 6 |

7.1，3，2

8.（共8分，每小题2分）（1）649；（2）发生越界中断；（3）发生缺段中断；（4）2064

**第六章**

一.填空题

1.DMA方式、通道控制方式 （顺序可交换）

2.输入井、输出井 （顺序不可交换）

3.控制器，通道

4.字节多路通道，数组多路通道

5.SPOOLing

6.DMA,通道

7.顺序，顺序

8.局部性、空间局部性、时间局部性

9.Spooling(或者 假脱机输入输出， 斯普林)

10.中断，DMA（不能调换）

二.选择题

1-5：CABBD 6-10:CB

三.判断题

1.错（2分），因为独占性的I/O设备不像 CPU和内存可以抢占，他们不能抢占，时间片轮转法其实是一种抢占方式。（3分）

四.简答题

1.不需要CPU全程参与的输入输出。

Spooling是在进程SP0，SP1控制下的“脱机”输入输出。

2.程序与具体设备无关，程序可以方便的在其它及其上执行。

3.写时复制是指当对一页共享内存执行写入操作时先将该页复制，再写入新的页，以达到既共享内存，又可写入的目的。实现实时复制，需要操作系统支持请求分页机制，并在缺页中断处理程序做相应的处理

4.轮询，中断，DMA，通道

5.先从系统设备表找到设备控制表,检查设备状态,空闲,则分配,否则进入设备等待队列:设备分配后找到跟设备相连的控制器的控制表,检查状态,忙则进入控制器等诗队列,空则分配：最后找到跟控制器相连的通道的控制表,检查状态,忙则进入控制器等待队列,空闲则分配,此时分配结束

**第七章**

一.填空题

1.文件

2.随机(或直接)，顺序

3.打开

4.基于索引节点的共享， 符号链接文件的共享

5.无结构字符流式，有结构的记录式，无结构字符流式

6.运行时动态

二.选择题

1-5：DCBAB 6-10：CCABD

三.判断题

1-5：√×

四.综合题

1.文件本身需要占用物理块数 128M/2k=64K,一个物理块可容纳2K/4=512个索引项，可以指向512个物理块，因此需要二次间接索引块（64K- 8 -512）/512=127个物理块，所以总共需要的物理块 是 64K+1+1+127 =65665

2.1057KB

因每个磁盘索引块和磁盘数据块大小均为256字节。所以4个直接地址索引指向的数据块大小为4×256字节。2个一级间接索引共包括2×(256/4)个直接地址索引，既其指向的数据块大小为2×(256/4)字节。1个二级间接地址索引所包含的直接地址索引数为(256/4)×(256/4)×256，即其所指向的数据块大小为(256/4)×(256/4)×256字节。  
  
即7个地址项所指向的数据块总大小为  
4×256+2×(256/4)×256+(256/4)×(256/4)×256=1082368字节=1057KB。

3.1GB磁盘共有1GB/16KB=65536块，索引块中每个索引项需要16位，因此需要2个字节存放索引块号，一个索引块最多有16KB/2=8K=8192个索引项，最大索引数据块的大小是8K\*16KB=128MB。故：

1091KB的文件需要1091KB/16KB=69个数据块，需要一个索引块，则占用的外存空间为69×16KB+16KB=1120KB

130MB的文件需要130MB/16KB=8320个数据块，超过一级索引能索引的最大块数8192，因此，需要一个二级索引块，2个一级索引块和8320个数据块，合计为=16KB+2\*16KB+8320\*16KB=133168KB。

**第八章**

一.填空题

1.块

2.寻道时间、旋转延迟时间、传输时间 （顺序可交换）

3.12.5

二.选择题

1-5：AAB

三.问答题

1.使平均寻道时间最短。