**操作系统原理复习**

**题型：一、选择题 二、简答题 三、综合应用题**

**综合应用题实例：**

**一、已知主存512K，OS占用低位40K，现有一作业序列如下：**

**J1要求 160K，J2要求 42K，J3要求 70K，J1完成,J4要求 130K，J5要求 140K，J3完成,J6要求 20K，J7要求 42K，J2完成，J8要求 62K。**

**试用最佳适应法为上述作业分配主存，画出主存分配情况和自由主存队列。（分配时，高地址处作为已分配区）（12’）**

**答：**

**主存分配情况 自由主存队列**

Flag:0

Size:30K

Next:40K

0

70K

∧

352K

0

60K

240K

OS:40K

空闲：60K

J5：140K

空闲：70K

J2：42K

空闲：30K

J4：130K

J3完成后

0

60K

∧

0

18K

240K

40K

OS:40K

空闲：18K

J7：42K

J5：140K

空闲：60K

J8：62K

J6：20K

J4：130K

J8分配后

**二、****一文件系统中，盘块大小为1KB=1024B(字节),盘块编号长4B,文件说明中可存放14个盘块编号。关于文件大小有如下统计结果：**

**文件大小≤4KB 占50%**

**4KB＜文件大小≤9KB 占20%**

**9KB＜文件大小≤256KB 占14%**

**256KB＜文件大小≤768KB 占8%**

**768KB＜文件大小≤64MB 占6%**

**64MB＜文件大小≤16GB 占2%**

**试为该文件系统设计文件的物理结构，使访问文件时具有尽可能小的平均访问磁盘次数，并计算其平均访问磁盘次数。（11’）**

**解：此文件系统应采用多级索引。先将文件大小转化为盘块个数**

**关于文件大小有如下统计结果：**

**文件大小≤4个盘块 占50%**

**4个盘块＜文件大小≤9个盘块 占20%**

**9个盘块＜文件大小≤256个盘块 占14%**

**256个盘块＜文件大小≤768个盘块 占8%**

**768个盘块＜文件大小≤64K个盘块 占6%**

**64K个盘块＜文件大小≤16M个盘块 占2%**

**考虑到一个索引块可索引1024/4=256个盘块。因此文件说明中可用编号a0-a8共9个标号索引9个盘块。用编号a9-a11共3个标号索引3个二级块,共3\*256=768个盘块。用编号a12可索引1个三级块,共1\*256\*256=64K个盘块。**

**用编号a13可索引1个四级块,共256\*256\*256=16M个盘块。**

**其平均访问磁盘次数=(50%+20%)\*1+(14%+8%)\*2+6%\*3+2%\*4=1.40**

**三、设某单面磁盘共有28000个扇区，采用CSCAN(循环扫描)磁盘调度策略，磁盘的旋转速度为每分钟6000转，每个磁道有100个扇区，相临磁道间的移动时间为1ms.若在某时刻，磁头位于100号磁道处，并沿着磁道号增大的方向移动,磁道号的请求队列为150，40，230，120，78，28，90，190，对请求队列中的每个磁道需读取1个随机分布的扇区，则读完这些扇区点共需要多少时间?需要给出计算过程。（8’）**

**解：每分钟6000转，转一圈需10ms,通过一扇区需要0.1ms;**

**访问时间=寻道时间+旋转时间平均5ms+通过扇区0.1ms;**

**按CSCAN则访问磁道的顺序为(当前100）-120-150-190-230-max-0-28-40-78-90 其中max=279**

**总的访问时间=（max-100+max+90)\*1ms+5.1ms\*8=588.8ms**

**四、设一系统中有三类资源，所有可用资源个数为（9，8，9）。某时刻系统中资源状态如下：Allocation Need 1)若P2提出请求Request（1,1,2），试问系统**

**P1: 2 1 1 3 2 4 能否将资源分配给它？为什么？**

**P2: 1 1 2 4 2 3 2)此后，P4提出请求Request（0,1,1），**

**P3: 1 2 1 2 1 1 试问系统能否将资源分配给它？为什么？（13’）**

**P4: 2 1 2 3 3 4**

**解：依题意可得Available（3，3，3）**

**1）若进程P2请求资源Req（1，1，2），按银行家算法判断如下：**

**1）判断Req(1,1,2)<=Need2(4,2,3),表示Req为合法请求；**

**2）判断Req(1,1,2)<=Available（3，3，3），表示Req为可满足的请求；**

**3）试探性分配**

**Available-=Req; 变为（2，2，1）**

**Alloc2+=Req; 变为（2，2，4）**

**Need2-=Req; 变为（3，1，1）**

**4）判断新状态的安全性**

**新状态是安全的，可找到安全序列P3,P2,P1,P4（具体过程在此略去）,因此可分配资源；**

**2）此后若进程P4请求资源Req（0，1，1），按银行家算法判断如下：**

**1）判断Req(0,1,1)<=Need4(3,3,4),表示Req为合法请求；**

**2）判断Req(0,1,1)<=Available（2，2，1），表示Req为可满足的请求；**

**3）试探性分配**

**Available-=Req; 变为（2，1，0）**

**Alloc4+=Req; 变为（2，2，3）**

**Need4-=Req; 变为（3，2，3）**

**4）判断新状态的安全性**

**此时可利用的资源不能满足任何进程的需求，新状态是不安全的，因此不可分配资源，进程P4阻塞，并取消试探性分配。**

五、一请求分页系统中，页面大小为2K，一作业共有7个页面，页面访问序列如下：

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 装入时刻 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 | 55 | 60 | 65 | 70 | 75 |
| 页号 | 2 | 1 | 0 | 3 | 2 | 1 | 3 | 5 | 2 | 3 | 6 | 2 | 1 | 4 |

时刻25时页面0，1，2，3分别装入到物理页框2，4，1，3中。(13分)

（1）若采用LRU页面置换算法，试计算缺页中断次数；假设时刻70时执行指令MOV AX，[2600]（AX为寄存器，2600为十进制），给出在执行过程中的地址变换过程。

（2）若采用FIFO页面置换算法，试计算缺页中断次数。

**解：**1）页块数为4，页面0 1 2 3 已装入内存，采用LRU算法，下面给出缺页中断时软件栈的变化情况（栈底打X号的为被淘汰的页面）：  
 5 6 1 4

5

3

1

2

0 X

6

3

2

5

1 X

1

2

6

3

5 X

4

1

2

6

3 X

共产生缺页中断4次。

LA=2600=1\*2K+552 p=1 w=552 访问页表，产生缺页中断；

按上图 时刻70时访问页面1，缺页中断会淘汰页面5，而页面5是淘汰页面0而使用页框2的 所以页框号b=2

PA=2\*2K+552=4648

1. 采用FIFO算法，需写出队列的变化：

5 2 6 1 4

2X5

1

0

3

5

1X2

0

3

5

2

0X6

3

5

2

6

3X1

5X4

2

6

1

共产生缺页中断5次。

**六、一座小桥横跨南北两岸，分南段、中段和北段三部分，南段较窄仅允许单向通行（即当有北向行人依次通过时，南向行人无法通过；同样，当有南向行人依次通过时，北向行人无法通过）；中段宽敞，允许多人通过或歇息；北段较短，仅允许两人同时通过，整座桥梁最多承重30人，试用信号量及PV操作实现南、北两岸行人过桥的同步，并列出信号量的含义和初值。**

**北向行人 南段 北段 南向行人**

**解：**semaphore empty=30;//桥梁最多承重30人

semaphore north=2;//北段仅允许两人通行

semaphore south=1;//南段单向通行

int ncount=0,scount=0; //通过南段时北向、南向行人个数

semaphore nmutex=1,smuter=1;//互斥访问ncount和scount

Cobegin {GO\_Northi();GO\_Southj();}coend

GO\_Northi()

{

P(empty);

P(nmutex);

if（ncount==0）P(south);

ncount++;

V(nmutex)；

过南段;

P(nmutex);

ncount--;

if（ncount==0）V(south);

V(nmutex)；

过中段或歇息;

P(north);

过北段;

V(north);

到达北岸;

V(empty);

}

GO\_Southj()

{

P(empty);

P(north);

过北段;

V(north);

过中段或歇息;

P(smutex);

if（scount==0）P(south);

scount++;

V(smutex)；

过南段;

P(smutex);

scount--;

if（scount==0）V(south);

V(smutex)；

到达南岸;

V(empty);

}

**七、有一仓库，可存放A和B两种产品，每次入库时只能存入A或B一种产品，每次出库时只能取出A或B一种产品。现要求**

**(1)-30<A产品数量-B产品数量<40**

**(2) A产品数量+B产品数量<200**

**试用P、V操作描述产品的入库过程和出库过程。（14分）**

**解：Main(){**

**Semaphore empty=199; //A+B<200**

**Semaphore fullA=0, fullB=0;**

**Semaphore mutex=1;**

**Semaphore AB=39; //A-B<40**

**Semaphore BA=29; //B-A<30**

**Cobegin**

**InLib();**

**OutLib();**

**Coend**

**}**

**入库过程 InLib() 出库过程OutLib()**

**while(有产品入库) while(有产品须出库)**

**{ {**

**if(产品为A) if(产品为A)**

**{ P(empty); { P(fullA)**

**P(AB) P(BA)**

**P(mutex) P(mutex)**

**A产品入库 A产品出库**

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

**V(BA) V(AB)**

**V(fullA); V(empty)**

**}else{ }else{**

**P(empty); P(fullB)**

**P(BA) P(AB)**

**P(mutex) P(mutex)**

**B产品入库 B产品出库**

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

**V(AB) V(BA)**

**V(fullB); V(empty)**

**} }**

**} }**