**代码+计算**

3.在银行家算法中，若出现下述资源分配情况：

|  |  |  |  |
| --- | --- | --- | --- |
| 进  程 | Allocation | Need | Available |
| A  B  C  D | A  B  C  D | A  B  C  D |
| P0  P1  P2  P3  P4 | 0  0  3  2  1  0  0  0  1  3  5  4  0  3  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 | 1  6  2  2 |

试问:（1)该状态是否安全?

(2)如果进程P2提出请求Request（1,2,2,2〉后,系统能否将资源分配给它?

3．解:(1)利用银行家算法对此时刻的资源分配情况进行分析,可得此时刻的安全性分析情况。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 进 程 | Work | Need | Allocation | Work+Allocation | Finish |
| A  B  C  D | A  B  C  D | A  B  C  D | A  B  C  D |
| P0  P3  P4  P1  P2 | 1  6  2  2  1  6  5  4  1  9  8  6  1  9  9  10  2  9  9  10 | 0  0  1  2  0  6  5  2  0  6  5  6  1  7  5  0  2  3  5  6 | 0  0  3  2  0  3  3  2  0  0  1  4  1  0  0  0  1  3  5  4 | 1  6  5  4  1  9  8  6  1  9  9  10  2  9  9  10  3  12  14  14 | true  true  true  true  true |

从上述分析中可以看出，此时存在一个安全序列{P0,P3,P4,P1,P2},故该状态是安全的。

（2）P2提出请求Request2(1,2,2,2),按银行家算法进行检查：

     Request2(1,2,2,2)≤Need2(2,3,5,6)

     Request2(1,2,2,2)≤Available(1,6,2,2)

     试分配并修改相应数据结构，资源分配情况如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 进  程 | Allocation | Need | Available |
| A  B  C  D | A  B  C  D | A  B  C  D |
| P0  P1  P2  P3  P4 | 0  0  3  2  1  0  0  0  2  5  7  6  0  3  3  2  0  0  1  4 | 0  0  1  2  1  7  5  0  1  1  3  4  0  6  5  2  0  6  5  6 | 0  4  0  0 |

再利用安全性算法检查系统是否安全,可用资源Available (0,4,0,0)已不能满足任何进程的需要,故系统进入不安全状态,此时系统不能将资源分配给P2。

2.己知某分页系统统,主存容量为64K,页面大小为1K,对一个4页大的作业;其0、1、2、3页分别被分配到主存的2、4、6、7块中。试将十进制的逻辑地址1023、2500、3500、4500转换成物理地址。

分析：在分页系统中进行地址转换时,地址交换机构将自动把逻辑地址转化为页号和页内地址,如果页号不小于页表长度,则产生越界中断;否则便以页号为索引去检索页表,从中得到对应的块号,并把块号和页内地址分别送入物理地址寄存器的块号和块内地址字段中,形成物理地址。

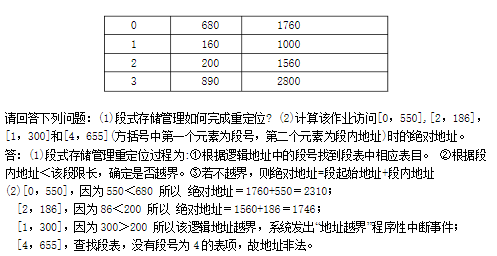
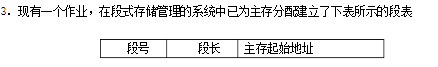
2．解答:对上述逻辑地址,可先计算出它们的页号和页内地址(**逻辑地址除以页面大小，得到的商为页号,余数为页内地址**),然后通过页表转换成对应的物理地址。

①逻辑地址1023:1023/1k,得到页号为0,页内地址为1023,查页表找到对应的物理块号为2,故物理地址为2×1K+1023=3071。

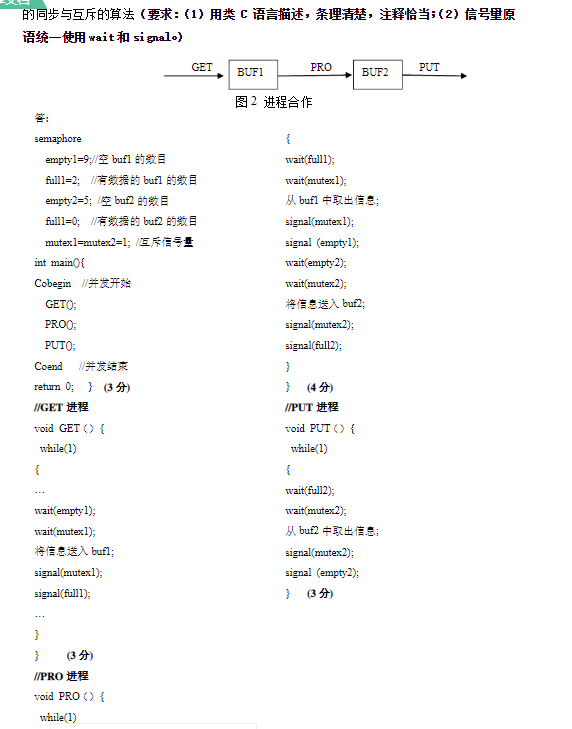
②逻辑地址2500：2500/1K，得到页号为2,页内地址为452,查页表找到对应的物理块号为6,故物理地址为6×1K+452=6596。

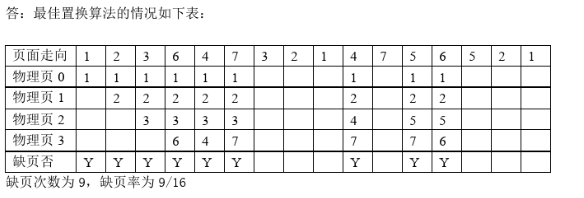
③逻辑地址3500:3500/1K,得到页号为3,:页内地址为428,查页表牛找到对应的物理块号为7,故物理地址为7×1K+428=7596。

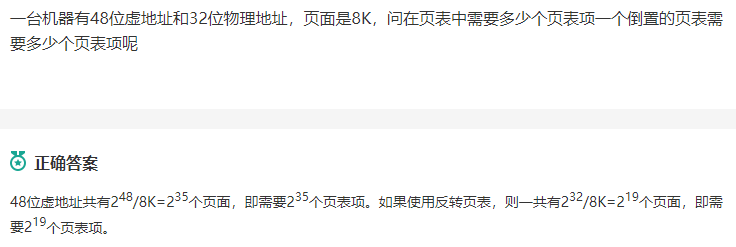
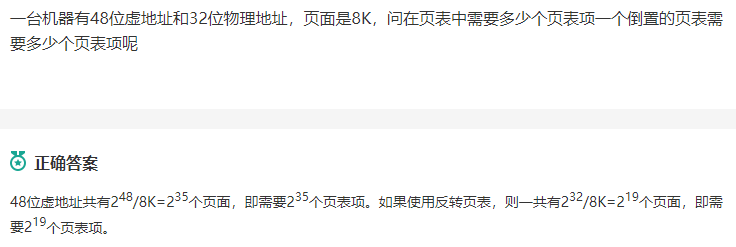
④逻辑地址4500:4500/1K,得到页号为4,页内地址为404,因页号不小于页表长度,故产生越界中断。

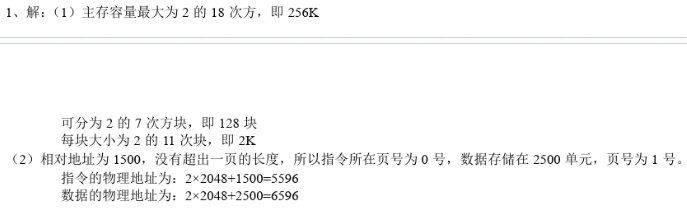
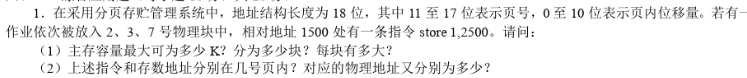
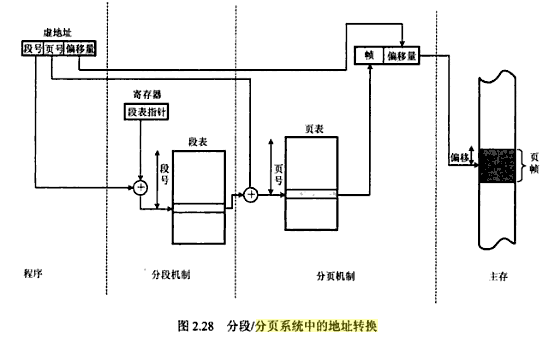


**\*\*\*所以段式存储有两种地址非法的情况：1.能在段表中查到逻辑地址的段号对应的表项，但是段偏移大于该段的长度，则地址越界。2.无法在段表中查到逻辑地址的段号对应的表项，则地址非法**







1. ****一台计算机有26位虚地址和16位物理地址，页面大小为1KB，问页表需要多少个页表项？

**2 (26 - 10) = 65536**

若采用“反转页表”，需要多少个页表项？

**2 (16 – 10) = 64**

1. 公共汽车上，司机循环执行的工作步骤是：启动汽车→正常行车→到站停车→开车门，乘务员循环执行的工作步骤是：等乘客上车→关车门→售票→等乘客下车。请用P、V原语实现司机、乘务员两个并发进程的同步。

答：Driver(司机) Attendant(乘务员)

开车门

L1: P(run) L2: 等乘客上车

启动汽车 关车门

正常行车 V(run) **允许开车**

到站停车 售票

开车门 P(stop)

V(stop) **允许售票**  等乘客下车

Goto L1 Goto L2

Main()

{

Semaphore run=0，stop=0；

Parbegin (Driver, Attendant);

}

1. 写出采用“Peterson算法”实现两个进程P0、P1互斥进入临界区的伪代码程序。

答：**P0**

**{**

**flag[0] = true;**

**turn = 1;**

**while ( flag[1] &&turn == 1 );**

***critical section***

**flag[0] = false;**

**}//0true turn1 判11 执行临界 0变flase**

**P1**

**{**

**flag[1] = true;**

**turn = 0;**

**while ( flag[0] && turn == 0 );**

***critical section***

**flag[1] = false;**

**}**

**main( )**

**{**

**boolean flag[2]={flase，false};**

**int turn=1;**

**parbegin(P0, P1);**

}

4. 若有一售票厅只能容纳300人,当少于300人时，可以进入；否则，需在外等候。若将每一个购票者作为一个进程，请用P、V操作编程，并写出信号量的初值。

4.解:购票者进程Pi （i=1，2，3，…）

P(S)

进入售票厅

购票

退出售票厅

V(S)

信号量的初值：S=300

5.　有一单向行驶的公路桥,每次只允许一辆汽车通过。当汽车到达桥头时,若桥上无车,便可上桥，否则,需等待,直到桥上的汽车下桥为止。若每一辆汽车为一个进程,请用P、V操作编程实现。

5.解：汽车进程Pi(i=1,2,3,…)

到达桥头

P(S)

上桥行驶

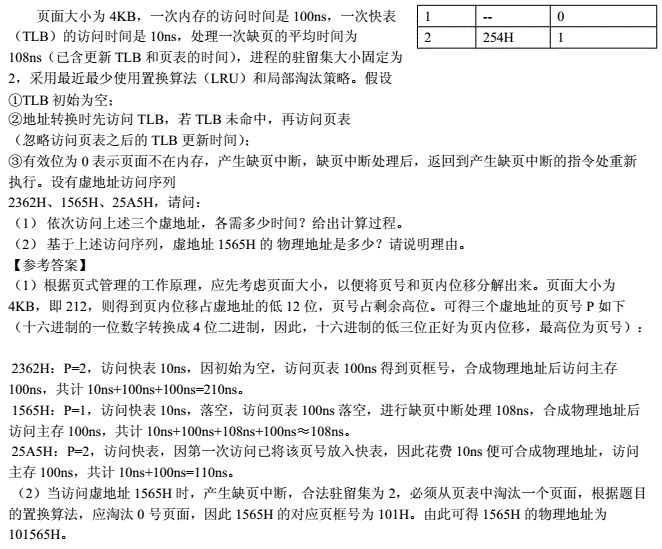
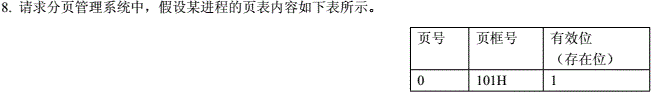
到达桥另一端

V(S)

信号量的初值：S=1

页表大小=表项数\*表项大小

页面大小=2页内偏移位数

****

**因为物理地址就是物理基地址+之前的那个偏移量，所以后面565不用变，直接在前面写上101即可了**

2．系统内存被划分成8块，每块4KB。某作业的虚拟地址空间共划分成16个页面。当前在内存的页与内存块的对应关系如下表所示，未列出的页表示不在内存。

|  |  |  |  |
| --- | --- | --- | --- |
| 页 号 | 块 号 | 页 号 | 块 号 |
| 0 | 2 | 4 | 4 |
| 1 | 1 | 5 | 3 |
| 2 | 6 | 9 | 5 |
| 3 | 0 | 11 | 7 |

试指出对应于下列虚拟地址的绝对地址：

（a）20 （b）4100 （c）8300

解：（a）虚拟地址20对应的页号是0，页内位移是20。用0去查页表，知道第0页现在存放在内存的第2块。由于每块的长度是4KB，所以第2块的起始地址为8192。因此，虚拟地址20所对应的绝对地址是：

8192+20=8212

（b）虚拟地址4100对应的页号是：

4100/4096=1（“/”是整除运算符）

对应的页内位移是：

4100%4096=4（“%”是求余运算符）

用1去查页表，知道第1页现在存放在内存的第1块。第1块的起始地址为4096。因此，虚拟地址4100所对应的绝对地址是：

4096+4=4100

（c）虚拟地址8300对应的页号是：

8300/4096=2（“/”是整除运算符）

对应的页内位移是：

8300%4096=108（“%”是求余运算符）

用2去查页表，知道第2页现在存放在内存的第6块。第6块的起始地址为

6×4K=24576

因此，虚拟地址8300所对应的绝对地址是

24576+108=24684

1．试说出图6-1（即教材中第2章的图2-2）所给出的监视程序A和计数程序B之间体现出一种什么关系，是“互斥”还是“同步”？为什么？



图6-1 对两个程序的描述

答：图6-1（即教材中第2章的图2-2）所给出的监视程序A和计数程序B之间体现出的是一种互斥关系，因为在监视程序A里，要对共享变量COUNT进行操作：

COUNT=COUNT+1;

在计数程序B里要对共享变量COUNT进行操作：

打印COUNT的值;

COUNT=0;

这两段程序是不能交叉进行的，不然就会出现与时间有关的错误。

1．在公共汽车上，司机和售票员的工作流程如图6-9（即教材上的图6-29）所示。为了确保行车安全，试用信号量及其P、V操作来协调司机和售票员的工作。

解：从日常生活知识知道，司机和售票员之间的工作有如下的制约关系存在。



图6-9 司机与售票员

（1）司机必须在得到售票员的“关门完毕”的信号后，才能启动汽车。这是一个司机要与售票员取得同步的问题。

（2）售票员必须在得到司机的“已经停车”的信号后，才能打开车门。这是一个售票员要与司机取得同步的问题。

因此，为了确保行车安全，需要设置两个同步信号量：

S1——初值为0，控制司机与售票员取得同步；

S2——初值为0，控制售票员与司机取得同步。

于是，在加入了信号量上的P、V操作后，图6-9应该变为图6-10。



图6-10 加入P、V操作后的司机与售票员

2．有一个阅览室共100个座位。用一张表来管理它，每个表目记录座号以及读者姓名。读者进入时要先在表上登记，退出时要注销登记。试用信号量及其P、V操作来描述各个读者“进入”和“注销”工作之间的同步关系。

解：分析题意，知道在管理读者“进入”和“注销” 阅览室的工作中，存在这样一些制约关系：

（1）100个座位是读者共同使用的资源，因此要用一个资源分配信号量来管理它；

（2）读者“进入”阅览室时，要申请座位。只有申请到座位才能进入，否则应该等待到座位的释放；

（3）没有读者时，不能做“注销”工作，必须等到有了读者才能做。

因此，可以设置两个信号量：

S1——初值为100，管理座位的分配；

S2——初值为0，控制“注销”与“进入”间取得同步。

“进入”与“注销”两个进程的流程如图6-11所示。



图6-11 “进入”与“注销”两个进程

在读者进入时，调用“进入”进程，通过P(S1)来申请座位。如果申请到，就可以办理阅览手续。如果100个座位都申请完毕，那么第101个读者就只有在关于S1的队列上等待，等到有人调用“注销”进程执行V(S1)。在有读者离去时，就调用“注销”进程。