

综 合 复 习 题

1. 三个进程 P1、P2、P3 互斥使用一个包含 N ($N > 0$) 个单元的缓冲区。P1 每次用 produce () 生成一个正整数并用 put () 送入缓冲区某一空单元中; P2 每次用 getodd () 从该缓冲区中取出一个奇数并用 countodd () 统计奇数个数; P3 每次用 geteven () 从该缓冲区中取出一个偶数并用 counteven () 统计偶数个数。请用信号量机制实现这三个进程的同步与互斥活动, 并说明所定义的信号量的含义。要求用伪代码描述。

参考答案:

定义资源信号量 empty、even、odd, 用于控制生产者与消费者之间的同步, 其中, empty 表示空缓冲区的数目, even 表示缓冲区中偶数的个数, odd 表示缓冲区中奇数的个数; 定义互斥信号量 mutex, 用于实现进程对缓冲区的互斥访问。伪代码描述如下:

semaphore empty=N,even=0,odd=0,mutex=1;

```
P1:
while(1)
{
    x=produce();
    wait(empty);
    wait(mutex);
    put(x);
    signal(mutex);
    if x%2==0
        signal(even);
    else
        signal(odd);
}
```

```
P2:
while(1)
{
    wait(odd);
    wait(mutex);
    getodd();
    countodd();
    signal(mutex);
    signal(empty);
}
```

```
P3:
while(1)
{
    wait(even);
    wait(mutex);
    geteven();
    counteven();
    signal(mutex);
    signal(empty);
}
```

2. 请求分页管理系统中, 假设某进程的页表内容如下表所示。

页号	页框号	有效位 (存在位)
0	101H	1
1	--	0
2	254H	1

页面大小为 4KB, 一次内存的访问时间是 100ns, 一次快表 (TLB) 的访问时间是 10ns, 处理一次缺页的平均时间为 10^8 ns (已含更新 TLB 和页表的时间), 进程的驻留集大小固定为 2, 采用最近最少使用置换算法 (LRU) 和局部淘汰策略。假设

① TLB 初始为空;

② 地址转换时先访问 TLB, 若 TLB 未命中, 再访问页表

(忽略访问页表之后的 TLB 更新时间);

③ 有效位为 0 表示页面不在内存, 产生缺页中断, 缺页中断处理后, 返回到产生缺页中断的指令处重新执行。设有虚地址访问序列

2362H、1565H、25A5H, 请问:

(1) 依次访问上述三个虚地址, 各需多少时间? 给出计算过程。

(2) 基于上述访问序列, 虚地址 1565H 的物理地址是多少? 请说明理由。

参考答案:

(1) 因为页大小为 4KB, 所以虚地址中的低 12 位表示页内地址, 剩余高位表示页号。则十六进制虚地址的低三位为页内地址, 最高位为页号。

2362H: 页号为 2, 页内地址为 362H。先访问快表 10ns, 未命中, 再访问内存中的页表 100ns, 页表项中的有效位指示该页在内存, 根据该页对应的页框号形成物理地址再次访问内存 100ns, 共计 $10\text{ns} + 100\text{ns} \times 2 = 210\text{ns}$ 。

1565H: 页号为 1, 页内地址为 565H。先访问快表 10ns, 未命中, 再访问内存中的页表 100ns, 页表项中的有效位指示该页不在内存, 处理缺页 10^8ns , 再次访问快表 10ns 命中, 根据该页对应的页框号形成物理地址再次访问内存 100ns, 共计 $10\text{ns} + 100\text{ns} + 10^8\text{ns} + 100\text{ns} + 10\text{ns} \approx 10^8\text{ns}$ 。

25A5H: 页号为 2, 页内地址为 5A5H。由于访问 2362H 时已将页 2 的表项写入 TLB, 因此访问快表 10ns, 命中, 根据该页对应的页框号形成物理地址访问内存 100ns, 共计 $10\text{ns} + 100\text{ns} = 110\text{ns}$ 。

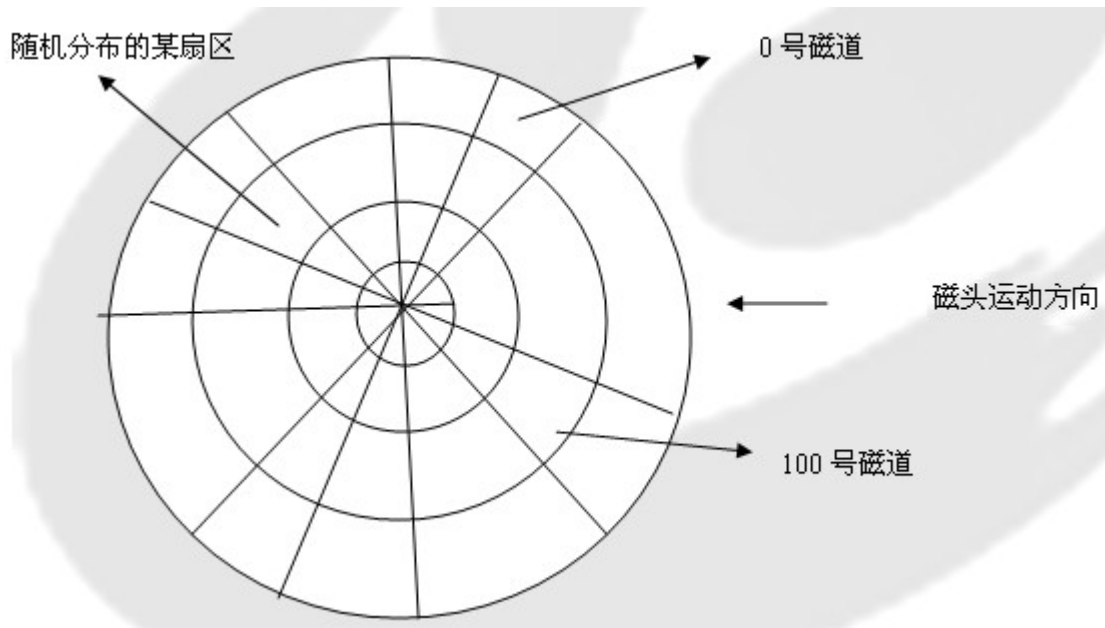
(2) 虚地址 1565H 的页号为 1, 页内地址为 565H。目前页 0、页 2 在内存, 访问页 1 时发生缺页, 根据 LRU 置换算法和局部淘汰策略, 将页 0 换出页 1 换入, 因此页 1 对应的页框号为 101H, 又块内地址为 565H, 则虚地址 1565H 的物理地址为 101565H。

3 假设计算机系统采用 CSCAN (循环扫描) 磁盘调度策略, 使用 2KB 的内存空间记录 16384 个磁盘块的空闲状态。

(1) 请说明在上述条件下如何进行磁盘块空闲状态管理。

(2) 设某单面磁盘旋转速度为每分钟 6000 转。每个磁道有 100 个扇区, 相邻磁道间的平均移动时间为 1ms。若在某时刻, 磁头位于 100 号磁道处, 并沿着磁道号大的方向移动 (如下图所示), 磁道号请求队列为 50、90、30、120, 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 4 个扇区点共需要多少时间? 要求给出计算过程。

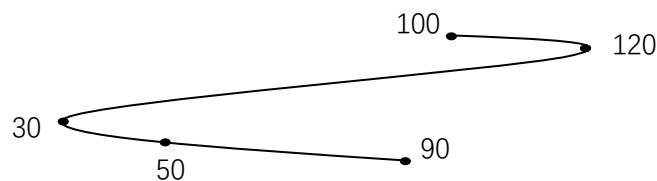
(3) 如果将磁盘替换为随机访问的 Flash 半导体存储器 (如 U 盘、SSD 等), 是否有比 CSCAN 更有效的磁盘调度策略? 若有, 给出磁盘调度策略的名称并说明理由; 若无, 说明理由。



参考答案:

(1) 可采用位示图法表示磁盘块的空闲状态, 一个磁盘块在位示图中用一个二进制位表示, 为 0 表示磁盘块空闲, 为 1 表示磁盘块已分配。16384 个磁盘块共占用 $16384\text{bit} = 16384/8\text{B} = 2048\text{B} = 2\text{KB}$, 正好可放在系统提供的内存中。

(2) 采用 CSCAN 调度算法, 磁道的访问次序为 120 30 50 90, 如下图所示:



因此访问过程中移动的磁道总数为 $(120-100) + (120-30) + (90-30) = 170$, 故总的寻道时间为 $170 \times 1\text{ms} = 170\text{ms}$;

由于每转需要 $1/6000$ 分钟 $= 10\text{ms}$, 则平均旋转延迟时间为 $10\text{ms}/2 = 5\text{ms}$, 总的旋转延迟时间为 $5\text{ms} \times 4 = 20\text{ms}$;

由于每个磁道有 100 个扇区, 则读取一个扇区需要 $10\text{ms}/100 = 0.1\text{ms}$, 总的读取扇区时间 (传输时间) 为 $0.1\text{ms} \times 4 = 0.4\text{ms}$;

综上, 磁盘访问总时间为 $170\text{ms} + 20\text{ms} + 0.4\text{ms} = 190.4\text{ms}$ 。

(3) 采用 FCFS (先来先服务) 调度策略更高效。因为 Flash 半导体存储器的物理结构不需要考虑寻道时间和旋转延迟时间, 可直接按 I/O 请求的先后顺序服务。

4 某银行提供 1 个服务窗口和 10 个顾客等待座位。顾客到达银行时，若有空座位，则到取号机领取一个号，等待叫号。取号机每次仅允许一个顾客使用。当营业员空闲时，通过叫号选取一位顾客，并为其服务。顾客和营业员的活动过程描述如下：

```
cobegin
{
    process 顾客 i
    {
        从取号机获得一个号码;
        等待叫号;
        获得服务;
    }
    process 营业员
    {
        while(true)
        {
            叫号;
            为顾客服务;
        }
    }
}
coend
```

请添加必要的信号量和 P、V（或 wait()、signal()）操作实现上述过程的互斥和同步。要求写出完整的过程，说明信号量的含义并赋初值。

参考答案：

```
semaphore seats=10; //表示空余座位数量的资源信号量，初值为 10
semaphore mutex=1; //互斥信号量，初值为 1，用于实现对取号机的互斥访问
semaphore custom=0; //表示顾客数量的资源信号量，初值为 0
cobegin
{
    process 顾客 i
    {
        P(seats);
        P(mutex);
        从取号机获得一个号码;
        V(mutex);
        V(custom);
        等待叫号;
        V(seats);
        获得服务;
    }
    process 营业员
    {
        while(TRUE)
        {
```

```
        P(custom);  
        叫号;  
        为顾客服务;  
    }  
}  
}  
coend
```

5. 某计算机主存按字节编址，逻辑地址和物理地址都是 32 位，页表项大小为 4 字节。请回答下列问题。

(1)若使用一级页表的分页存储管理方式，逻辑地址结构为：

页号 (20 位)	页内偏移量 (12 位)
-----------	--------------

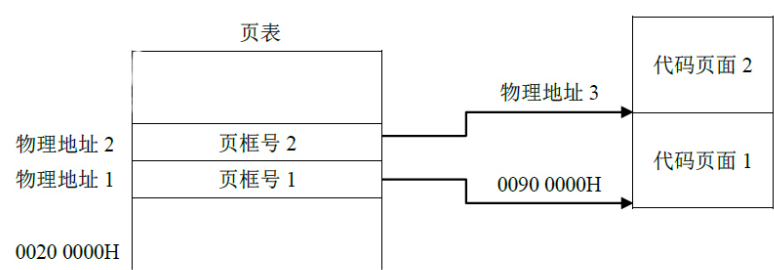
则页的大小是多少字节？页表最大占用多少字节？

(2)若使用二级页表的分存存储管理方式，逻辑地址结构为：

页目录号 (10 位)	页表索引 (10 位)	页内偏移量 (12 位)
-------------	-------------	--------------

设逻辑地址为 LA，请分别给出其对应的页目录号和页表索引。

(3)采用(1)中的分页存储管理方式，一个代码段起始逻辑地址为 0000 8000H，其长度为 8KB，被装载到从物理地址 0090 0000H 开始的连续主存空间中。页表从主存 0020 0000H 开始的物理地址处连续存放，如下图所示(地址大小自下向上递增)。请计算出该代码段对应的两个页表项物理地址、这两个页表项中的框号以及代码页面 2 的起始物理地址。



参考答案：

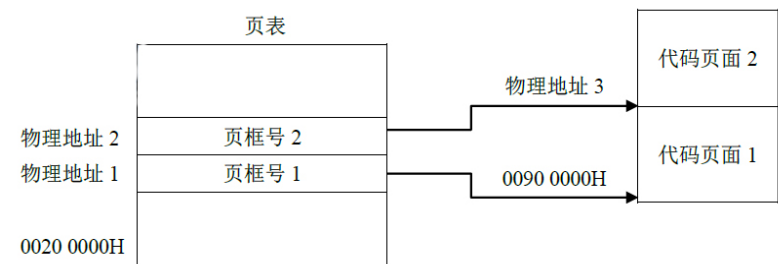
(1) 页的大小是 4K 字节，页表最大占用 4M 字节

(2) 页目录号=INT[INT[LA/4K]/1K]；页表索引= INT[LA/4K] mod 1K

页目录号：(((unsigned int)(LA))>>22)&0x3FF 或 ((unsigned int)(LA))>>22；

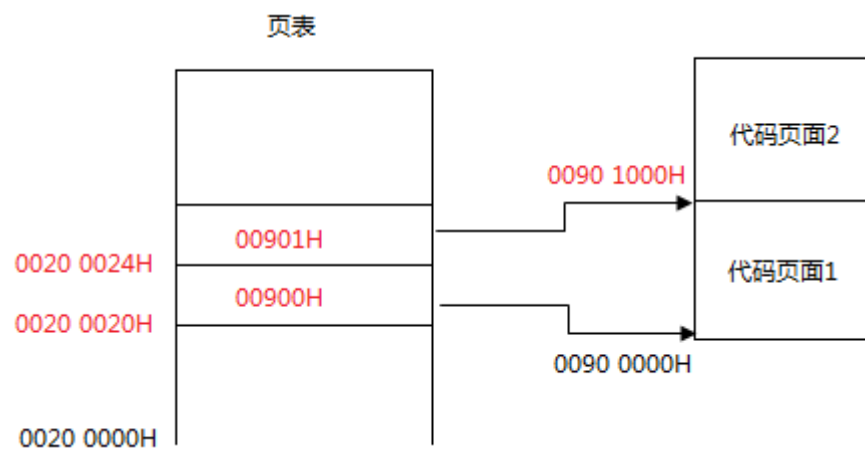
页表索引：(((unsigned int)(LA))>>12)&0x3FF 或 ((unsigned int)(LA))>>12。

(3)



物理地址 1： 0020 0020H 物理地址 2： 0020 0024H 物理地址 3： 0090 1000H

页框号 1： 00900H 页框号 2： 00901H



6. 46. (8 分) 设某计算机的逻辑地址空间和物理地址空间均为 64KB，按字节编址。若某进程最多需要 6 页 (Page) 数据存储空间，页的大小为 1KB，操作系统采用固定分配局部置换策略为此进程分配 4 个页框 (Page Fame)。在时刻 260 之前该进程访问情况如下表所示 (访问位即使用位)。

页号	页框号	装入时刻	访问位
0	7	130	1
1	4	230	1
2	2	200	1
3	9	160	1

当该进程执行到时刻 260 时，要访问逻辑地址为 17CAH 的数据，请问答下列问题：

- (1) 该逻辑地址对应的页号是多少？
- (2) 若采用先进先出 (FIFO) 置换算法，该逻辑地址对应的物理地址是多少？要求给出计算过程。
- (3) 若采用时钟 (CLOCK) 置换算法，该逻辑地址对应的物理地址是多少？要求给出计算过程。(设搜索下一页的指针沿顺时针方向移动，且当前指向 2 号页框，示意图如下。)



参考答案：

(1) 由于计算机的逻辑地址空间和物理地址空间均为 64KB=2¹⁶B，按字节编址，且页 (块) 的大小为 1KB=2¹⁰B，所以计算机的逻辑地址结构和物理地址结构均为：

页 (页框) 号 (6 位)	页 (块) 内偏移量 (10 位)
----------------	-------------------

17CA H=(0001 0111 1100 1010)₂，所以 17CAH 对应的页号是(000101)₂=5。

(2) 若采用先进先出 (FIFO) 置换算法，则置换装入时间最早的页，故 0 号页被置换，将 5 号页装入 7 号页框，所以 17CA H 对应的物理地址为 (0001 1111 1100 1010)₂=1FCA H。

(3) 若采用时钟 (CLOCK) 置换算法，则从当前指针指示页框开始查找，若其中页的访问位为 0，则置换该页，否则将访问位清零，并将指针指向下一个页框，继续查找。由于初始时内存中的 4 个页的访问位均为 1，因此，前 4 次查找并未找到合适的页，但查找时已将对应页的访问位清零，第 5 次查找时，指针重新指向 2 号页框，其中存放的 2 号页的访问位为 0，故置换该页，将 5 号页装入 2 号页框，所以 17CA H 对应的物理地址为 (0000 1011 1100 1010)₂=0BCA H。