

week9

1

某系统采用页式虚拟存储管理，贮存每块为 128 个字节，现在要把一个 128×128 的二维数组置初值为“0”。在分页时把数组中的元素每一行放在一页中，假定系统只分给用户一页数据区。

1. 对如下数据段，执行完要产生多少次缺页中断？

```
1  var A: array[1...128] of array[1...128] of integer;
2      for j := 1 to 128
3          do for i := 1 to 128
4              do A[i,j] := 0;
```

2. 为减少缺页中断的次数，请改写上面的程序，使之仍能完成所要求的功能，并统计缺页次数。

1.1

由于系统只分给用户一页数据区，同时数组中的一列位于同一个页内，因此当数组按列访问时，每一次访问时，由于上一个访问的元素一定不是该行中的元素，因此系统中的页一定不是该元素所在的页，因此回访问新的页导致老的页被换出。

总的缺页中断次数等同于访问的元素的数量 $128 * 128 = 16348$

1.2

使元素按行访问，而不是按列访问。这样在内层循环时，除了第一个元素以外，不会发生缺页中断

```
1  var A: array[1...128] of array[1...128] of integer;
2      for i := 1 to 128
3          do for j := 1 to 128
4              do A[i,j] := 0;
```

此时缺页中断次数下降到了128次

2

假设有一个按需调页存储器，页表放在寄存器中。处理一个页错误，当有空的帧可用或被置换的帧没有被修改过时要 8ms，当被置换的帧被修改过时用 20ms。存储器存取时间为 100ns。假设被置换的页中有 70% 被修改过，有效存取时间不超过 200ns 时，最大可以接受的缺页率为多少？

有效存取时间 (EAT) = $(1 - \text{缺页率}) \times \text{存储器存取时间} + \text{缺页率} \times \text{页面错误处理时间}$

页面错误处理时间 = $0.7 \times 20ms + 0.3 \times 8ms$

最大缺页率大约为 6.10×10^{-6}

3

已知页面走向为1、2、1、3、1、2、4、2、1、3、4，且开始执行时主存中没有页面。若只给该作业分配 3 个物理块，当采用 FIFO 页面淘汰算法时缺页率为多少？假设现有一种淘汰算法，该算法淘汰页面的策略为当需要淘汰页面时，就把刚使用过的页面作为淘汰对象，试问就相同的页面走向，其缺页率为多少？

缺页率

- 1. **FIFO（先进先出）** 页面淘汰算法的缺页率约为 45.45%。
- 2. **刚使用过的页面淘汰算法**的缺页率约为 54.55%。

置换图

- 1. **FIFO 算法的置换图**（每一行代表一个时间步骤，每一列代表一个物理块）：

时间步骤	物理块 1	物理块 2	物理块 3
初始			
1	1		
2	1	2	
3	1	2	
4	1	2	3
5	1	2	3
6	1	2	3
7	2	3	4
8	2	3	4
9	3	4	1
10	3	4	1
11	3	4	1

- 2. **刚使用过的页面淘汰算法的置换图**（每一行代表一个时间步骤，每一列代表一个物理块）：

时间步骤	物理块 1	物理块 2	物理块 3
初始			
1	1		
2	1	2	
3	1	2	
4	1	2	3
5	1	2	3
6	1	2	3
7	1	2	4
8	1	2	4
9	1	2	4
10	1	2	3
11	1	2	4

4

在一个请求式分页系统中，目前系统的利用率如下：

CPU操作：20%

分页磁盘的I/O操作：97.7%

其它I/O设备：5%

下列方法是否可以提高 CPU 利用率，分别说出你的理由。

1. 安装一个更加快速的CPU；
2. 增加一个容量更加大的磁盘；
3. 增加更多的内存；
4. 增加页面的大小。

1. 安装一个更加快速的CPU错误.系统处于频繁的换入换出过程中,CPU处于空闲状态,利用率不高
2. 增加一个容量更加大的磁盘错误.系统已经处于实际的换入换出过程中,增加磁盘,增加磁盘的swap分区的容量无用
3. 增加更多的内存正确.因为增加内存可以使每个程序获得更多的页框,能减少缺页率,进而减少换入换出过程,可提高CPU的利用率
4. 增加页面大小正确.页面大小增大后缺页率降低,可减少换入换出过程,提高CPU利用率