

## 第四章 实验五

### 模拟最近最久未使用(LRU)页面置换算法

- **置换策略：**选择最近最久未使用的页面予以淘汰，系统在每个页面设置一个访问字段，用以记录这个页面自上次被访问以来所经历的时间  $T$ ，当要淘汰一个页面时，选择  $T$  最大的页面。

顺序	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
内存块	7	7	7	2		2		4	4	4	0			1		1		1		
		0	0	0		0		0	0	3	3			3		0		0		
			1	1		3		3	2	2	2			2		2		7		
缺页否	√	√	√	√		√		√	√	√	√			√		√		√		

$$\text{缺页率} = 12/20 = 60\%$$

- 使用**数组**来模拟最近最久未使用(LRU)页面置换算法。  
int a[M]; /\*存放已装入内存的页号序列，M 为系统分配给作业的主存页面数\*/  
int b[N]; /\*存放作业页号序列，N 为要装入作业的页面总数\*/  
int c[N]; /\*存放被淘汰的页号序列\*/
- **算法思想：**a[0]始终存放最近最久未使用的页面，a[M-1]存放最近刚被使用的页面。  
(1) 当待访问页面 b[i]在内存数组 a 中下标 k 时，**处理过程：**①将页面 a[k+1]...a[M-1]全部前移一位；②将页面 b[i]放置在 a[M-1]位置。  
(2) 当待访问页面 b[i]不在内存，发生缺页且内存已满时，**置换过程：**①每次置换出 a[0]页面；②将页面 a[1]...a[M-1]全部前移一位；③将新页面插入到 a[M-1]位置。

- 输入作业页号序列：

输入：

```
请输入作业序号：
7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1
```

- 输出如下结果：

输出：

```
发生缺页的次数=12
缺页中断率=60.00%
驻留内存的页号分别为： 1, 0, 7,
被淘汰的页号分别为： 7, 1, 2, 3, 0, 4, 0, 3, 2,
```

- 使用 Microsoft Visual Studio C++ 6.0 或 CodeBlocks 编程：程序 4\_5\_LRU\_page.cpp。完善如下程序代码：

```
#define M 3 /*M 为系统分配给作业的主存页面数*/
#define N 20 /*N 为要装入作业的页面总数*/
int a[M]; /*存放已装入内存的页号序列*/
```

```
int b[N]; /*存放作业页号序列*/
int c[N]; /*存放被淘汰的页号序列*/

#include <stdio.h>
int LRU() /*分别调入页面判断是否需要置换*/
{ 填补程序 }
void main()
{
    int count; /*count 为缺页总次数*/
    { 填补程序 }
}
```