

第四章 实验四

模拟先进先出(FIFO)页面置换算法

- **置换策略：**总是淘汰最先进入内存的页面，即选择在内存中驻留时间最久的页面予以淘汰。

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	2	4	4	4	0			0	0			7	7	7
		0	0	0		3	3	3	2	2	2			1	1			1	0	0
			1	1		1	0	0	0	3	3			3	2			2	2	1
缺页否	√	√	√	√		√	√	√	√	√	√			√	√			√	√	√

$$\text{缺页率} = 15/20 = 75\%$$

- 使用数组来模拟先进先出(FIFO)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]始终存放最先进入内存的页面。**页面置换过程：**①每次置换出 a[0] 页面；②将之后的页面 a[1...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=15
缺页中断率=75.00%
驻留内存的页号分别为： 1, 0, 7,
被淘汰的页号分别为： 7, 0, 1, 2, 3, 0, 4, 2, 3, 0, 1, 2,
```

- 使用 Microsoft Visual Studio C++ 6.0 或 CodeBlocks 编程：程序 4_4_FIFO_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*/
#include <stdio.h>
bool found(int x,int a[M]) /*以下函数判断第 i 个页面是否在内存*/
{ 填补程序 }
void main()
```

```
{  
    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*/  
    int count; /*count 为缺页总次数*/  
    bool flag; /*flag 标识该页是否在内存*/  
    { 填补程序 }  
}
```