**实验五 虚拟存储器管理**

**1 实验目的**

1. 了解虚拟存储技术的特点；

2. 掌握请求页式管理的页面置换算法。

**2 实验内容**

1. 通过随机数产生一个指令序列，共 320 条指令。其地址按下述原则生成：

（1）50%的指令是顺序执行的；

（2）50%的指令是均匀分布在前地址部分；

（3）50%的指令是均匀分布在后地址部分；

具体的实施方法是：

A. 在[0，319]的指令地址之间随机选取一起点 M；

B. 顺序执行一条指令，即执行地址为 M+1 的指令；

C. 在前地址[0，M+1]中随机选取一条指令并执行，该指令的地址为 M’；

D. 顺序执行一条指令，其地址为 M’+1；

E. 在后地址[M’+2，319]中随机选取一条指令并执行；

F. 重复 A—E，直到执行 320 次指令。

2. 指令序列变换成页地址流

设：

（1）页面大小为 1K；

（2）用户内存容量为 4 页到 32 页；

（3）用户虚存容量为 32K。

在用户虚存中，按每 K 存放 10 条指令排列虚存地址，即 320 条指令在虚存中的存放方式为：

第 0 条—第 9 条指令为第 0 页（对应虚存地址为[0，9]）；

第 10 条—第 19 条指令为第 1 页（对应虚存地址为[10，19]）；

……………………

第 310 条—第 319 条指令为第 31 页（对应虚存地址为[310，319]）；

按以上方式，用户指令可组成 32 页。

3. 计算并输出下述各种算法在不同内存容量下的命中率。

A. 先进先出（FIFO）页面置换算法

B. 最近最久未使用（LRU）页面置换算法--最近最少使用算法

C. 最少使用（LFR）页面置换算法

D. 最佳（Optimal）页面置换算法

**3. 实验准备**

1. 先进先出（FIFO）页面置换算法

该算法总是淘汰最新进入内存的页面，即选择在内存中驻留时间最久的页面予以淘汰。该算法实现简单，只需把一个进程已调入内存的页面，按先后次序链接成一个队列，并设置一个指针，称为替换指针，使它总是指向最老的页面。

2. 最近最久未使用（LRU）页面置换算法

最近最久未使用（LRU）页面置换算法，是根据页面调入内存后的使用情况进行决策的。由于无法预测各页面将来的使用情况，只能利用“最近的过去”作为“最近的将来”的近似，因此，LRU 置换算法是选择最近最久未使用的页面予以淘汰。该算法赋予每个页面一个访问字段，用来记录一个页面自上次被访问以来所经历的时间 t，当需淘汰一个页面时，选择现有页面中其 t 值最大的，即最近最久未使用的页面予以淘汰。

3. 最少使用（LFR）页面置换算法

在采用该算法时，应为在内存中的每个页面设置一个移位寄存器，用来记录页面被访问的频率。该置换算法选择在最近使其使用最少的页面作为淘汰页。

4. 最佳（Optimal）页面置换算法

该算法选择的被淘汰页面，将是以后永远不使用的，或许是在最长（未来）时间内不再被访问的页面。采用该算法，通常可保证获得最低的缺页率。但由于人们目前还无法预知一个进程在内存的若干个页面中，哪一个页面是未来最长时间内不再被访问的，因而该算法是无法实现的，但可以利用该算法去评价其他算法。

提示：A.命中率=1-页面失效次数/页地址流长度

B.本实验中，页地址流长度为 320，页面失效次数为每次访问相应指令时，该指令所对应的页不在内存的次数。

C.关于随机数产生方法，采用 VC 系统提供函数 RAND()和 RANDOMIZE()来产生