二、分页式存储管理

**2.1目的**

在第1部分实验基础上实现进程的分页式内存分配和地址转换过程，并进一步实现请求分页式存储分配和地址转换过程。页面置换算法实现先进先出（FIFO）、最近最久未使用（LRU）、OPT算法。

**2.2内容**

实现FIFO、LRU、OPT算法。

**2.3数据结构**

用结构体存储页表，为FIFO、LRU、OPT分别建立页表，每个算法本身用数组模拟栈。

**2.4算法设计及流程图**

**每次输入地址后，计算出页号，若页号越界，则给出错误提示。否则依次调用FIFO和LRU算法，这里值得注意的是，由于我们的FIFO算法先于LRU算法被调用，那么当在处理FIFO算法时，我们暂且不将位视图相应位置做变化，留到处理LRU算法再做处理。**

**对于FIFO、LRU算法的缺页，我们分两种情况考虑，第一种是模拟栈内还有空间，那么直接将其入栈。第二种是模拟栈内无空间，要发生置换。发生置换时把模拟栈最底的弹出，新来的加入栈顶。**

**对于FIFO、LRU算法的命中，二者不同之处就是对于LRU算法，当元素命中时，要将该元素移至栈顶。**

**对于OPT算法，每次置换时向后查找，与栈内未来最久不使用的进行置换，若无法找到最久不使用的元素，那么默认与栈底元素进行置换。**

**回填位视图时只要对FIFO页表或LRU页表进行扫描，找到状态为true的位置，将其在位视图中对应的位置置回原值即可。**

**对于三种算法的输出格式，只要开三个相对应的二维数组，然后每次按照格式对二维数组进行赋值即可。**









**2.5小结**

对三种算法理解的更透彻，通过对其过程的模拟，发现其中规律。