课程名称 计算机操作系统

实验成绩

指导教师 曹勇



实 验 报 告

院系 信息工程学院

班级 物联网工程

学号 1601141019

姓名 晏沈威

日期 2018/10/29 地点：505

**­** 实验四、LRU算法模拟

一、实验目的和要求

用高级语言模拟页面置换算法LRU，加深对LRU算法的认识。

二、实验原理

其基本原理为：如果某一个页面被访问了，它很可能还要被访问；相反，如果它长时间不被访问，再最近未来是不大可能被访问的。

三、实验环境

1、pc

2、vc++

四、程序源代码：

#define MAXSIZE 20

#include "iostream"

using namespace std;

int main()

{

int input=0;

//用于输入作业号

int worknum=0;

//输入的作业个数

int storesize=0;

//系统分配的存储区块数

int interrupt=0;

//缺页中断次数

int stack[MAXSIZE];

//栈，LRU算法的主要数据结构

int workstep[MAXSIZE];

//记录作业走向

/\*

初始化

\*/

for(int i=0;i<MAXSIZE;i++)

{

stack[i]=0;

workstep[i]=0;

}

cout<<"请输入存储区块数：";

cin>>storesize;

cout<<"请输入作业的页面走向（0结束）：\n";

for(int j=0;j<MAXSIZE;j++)

{

cout<<"页面号"<<j+1; cin>>input;

workstep[j]=input;

if(input==0)

{

cout<<"输入结束！\n";

break;

}

worknum++;

}

if(workstep[0]==0)

{

cout<<"未输入任何作业，系统将退出！\n";

return 0;

}

cout<<"置换情况如下：\n";

for(int k=0;k<worknum;k++)

{

/\*

在栈中找相等的页号或空位置

\*/

for(int l=0;l<storesize;l++)

{

/\*

是否有相等的页号

\*/

if(stack[l]==workstep[k])

{

cout<<"内存中有"<<workstep[k]<<"号页面，无须中断！\n";

goto step1;

}

/\*

找栈中是否有空位置

\*/

if(stack[l]==0)

{

stack[l]=workstep[k];

cout<<"发生中断，但内存中有空闲区，"<<workstep[k]<<"号页面直接调入！\n";

interrupt++;

goto step1;

}

}

/\*

上述情况都不成立则调出栈顶，将调入页面插入栈顶

\*/

cout<<"发生中断，将"<<stack[0]<<"号页面调出，"<<workstep[k]<<"号装入！\n";

interrupt++;

/\*

新掉入的页面放栈顶

\*/

step1:

for(int m=0;m<storesize;m++)

{

stack[m]=stack[m+1];

}

stack[storesize-1]=workstep[k];

}

cout<<"作 业"<<worknum<<"个 ，"<<"中 断"<<interrupt<<"次 ，"<<"缺 页 率 ："<<float(interrupt)/float(worknum)\*100<<"%\n";

}

实验结果：



六、实验总结

通过本实验我对模拟实现页式地址重定位有更深刻的认识，并且了解的模拟实现页式地址重定位的具体思想和运算过程