学号 E21514033 专业 软件工程 姓名 何铭春

实验日期 教师签字 成绩

实验报告

【实验名称】 页式地址重定位模拟

【实验目的】

1. 掌握页式地址重定位的基本概念和方法；

2. 加深理解页式地址重定位技术在多道程序设计中的作用和意义。

【实验原理】

1.设计页表结构，能够通过逻辑地址查询到物理块号。

2.通过输入逻辑地址，按照算法计算出物理地址。

3.在程序运行过程中有合理的提示。

【实验内容】

1.程序布局：

memory1.0.cpp 主函数文件，用于创建页表，和调用其他函数。

showPageStruct.h 显示页表函数

showProcess.h 显示当前进程信息

addProcess.h 添加新进程并分配内存函数

deleteProcess.h 删除进程并释放内存函数

search.h 通过逻辑地址查询物理地址函数。

2.数据结构：

页表结构

page:保存页号

block:保存对应块号

next:记录下一个页号，新进程创建时添加，进程删除时通过next循环删除。

class pageStruct

{

public:

int page;

int block;

int next;

};

进程表结构：

id自动添加，逐步自增

start：进程存储开始页号，通过开始页号的next能够删除进程。

size：进程所需内存大小

class processStart

{

public:

int id;

int start;

int size;

processStart()

{

id=-1;

start=-1;

size=-1;

}

processStart(const processStart& P)

{

id=P.id;

size=P.size;

start=P.start;

}

};

3.新增进程函数详细结构

//用于随机寻找一个物理块保存信息，并返回找到的物理块号。

int setMemoryBlock(int\* memoryBlock,int memoryCounter)

//第一层循环是进程所需块数，第二层循环用于找一个空闲页号，并调用setMemoryBlock函数设置物理块号。

int changeMemory(pageStruct\* ps,int memoryCounter,int tempCounter,int\* memoryBlock)

//添加进程函数，在用户输入进程大小后，查找到页表，找到第一个空闲页号。并使用这个页号调用进程新增函数。

int addProcess(vector<processStart >& v,pageStruct \*ps,int memoryCounter,int memorySize,int\* memoryBlock)

4.逻辑地址与物理地址转换函数

函数接收参数页表和预设物理块大小，要求用户输入逻辑地址进行计算。

int search(pageStruct\* ps,int memorySize)

{

int pageAddress;

//p用于保存除法产生的页表号

//d用于保存余数

int p;

int d;

int address;

cout<<"请输入逻辑地址：";

cin>>pageAddress;

p=int(pageAddress/(memorySize\*1024));

if(ps[p].block==-1)

cout<<"无效的逻辑地址"<<endl;

else

{

d=(pageAddress % (memorySize\*1024));

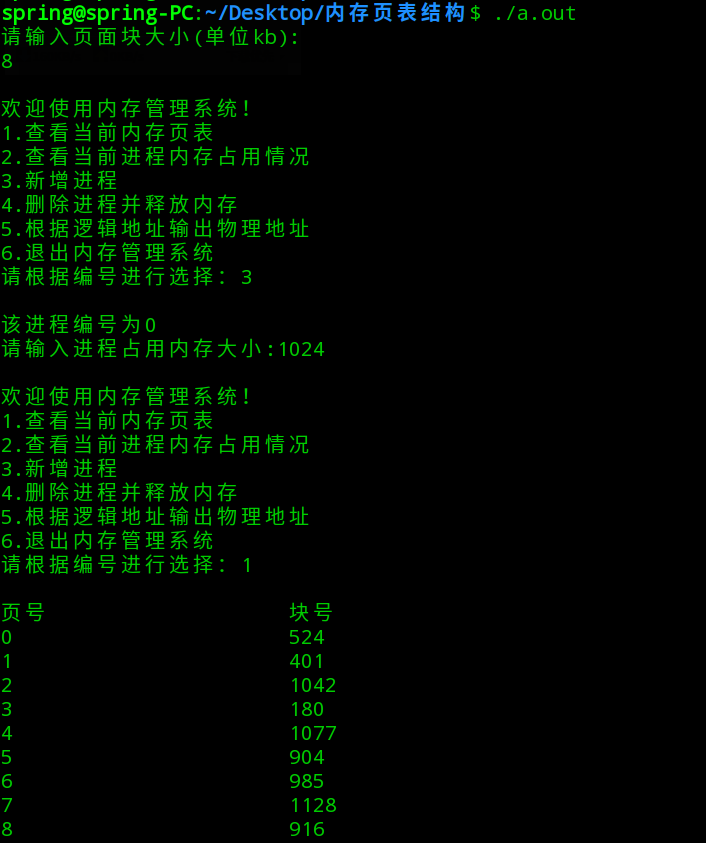
address=ps[p].block\*1024\*memorySize+d;

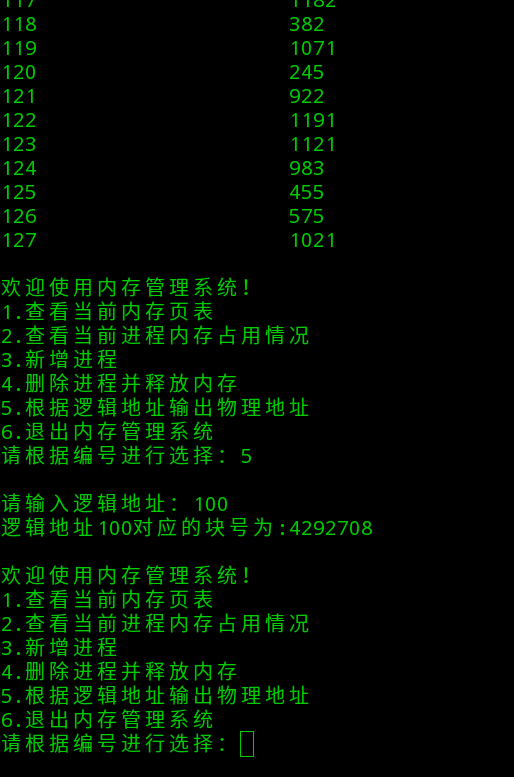
cout<<"逻辑地址"<<pageAddress<<"对应的块号为:"<<address<<endl;

}

}

实验截图：





【小结或讨论】

1.页表结构的实现不是很难，通过简单的随机数和数组结构就能实现。

2.为了实现进程的访问，我在页表结构中增加了一个next用于找出进程的所有空间，便于进程的访问和删除。