FIFO-----页面置换

#include <stdio.h>

#include <time.h>

#include <stdlib.h>

#define max\_page 10

int Page[320] = {0};

int Page\_flu[320]={0};

int count=0;

double lack\_page=0;

int count\_page=max\_page;

int circle=1;

struct Memo{

int num;

int a;

struct Memo \*next;

};

int Judge\_Page(int value){

return value/10;

}

int scan\_queen(struct Memo \*hear,int value){

struct Memo \*move;

move=hear->next;

while(move!=NULL){

if(move->a==value){

return 1;

}

move=move->next;

}

return 0;

}

void print(struct Memo \*hear){

struct Memo \*move;

move=hear->next;

while(move!=NULL){

printf("%d",move->a);

move=move->next;

}

printf("\n");

}

void insert(struct Memo \*hear,int value,int ZL){

if(count\_page>=1){

struct Memo \*move;

move=hear->next;

while(move->a!=-1){

move=move->next;

}

move->a=value;

count\_page--;

printf("页面%d被调入----对应指令为：%d\n",value,ZL);

}

else{

struct Memo \*move;

move=hear->next;

while(move->num!=circle){

move=move->next;

}

printf("页面%d被调出，页面%d被调入----指令为：%d\n",move->a,value,ZL);

move->a=value;

circle++;

if(circle==max\_page+1){

circle=1;

}

}

print(hear);

}

void FIFO(struct Memo \*hear){

int i=0;

for(i=0;i<=319;i++){

if( scan\_queen(hear,Page\_flu[i])==0){

lack\_page++;

insert(hear,Page\_flu[i],Page[i]);

}

else{

printf("指令%d对应页面%d已在内存\n",Page[i],Page\_flu[i]);

}

}

}

void Pro\_Page(){

int m=0;

m=rand()%320;

Page[count]=m;

count++;

if(count==320){

return;

}

int m\_=0;

m\_=rand()%(m+1);

Page[count]=m\_;

count++;

if(count==320){

return;

}

Page[count]=m\_+1;

count++;

if(count==320){

return;

}

int m\_\_=0;

m\_\_=(m\_+2)+rand()%(319-(m\_+2)+1 );

Page[count]=m\_\_;

count++;

if(count==320){

return;

}

Pro\_Page();

}

void Flu(){

int i=0;

for(i=0;i<=319;i++){

Page\_flu[i]=Judge\_Page(Page[i]);

}

}

int main(){

struct Memo Stu[max\_page+1];

struct Memo \*hear;

hear=&Stu[0];

int i=0;

for(i=0;i<=max\_page;i++){

if(i==max\_page){

Stu[i].a=-1;

Stu[i].next=NULL;

Stu[i].num=i;

break;

}

Stu[i].next=&Stu[i+1];

Stu[i].a=-1;

Stu[i].num=i;

}

srand(time(0));

Pro\_Page();

Flu();

/\*

printf("页地址流：\n");

for(i=0;i<=319;i++){

printf("%d",Page[i]);

if(i%3==0&&i!=0){

printf("\n");

}

}

printf("\n");

\*/

FIFO(hear);

printf("缺页次数为：%0.01f\n",lack\_page);

printf("命中率为:%1f\n",1-lack\_page/320);

return 0;

}

结果：

