

#include<stdio.h>

#include<stdlib.h>

#include<math.h>

#define NUM 10 //磁道号总数

int num[NUM]; //磁道号数组

int isTrue;

/\*从文件中读取磁道号，存入num[]数组\*/

int openFile()

{

isTrue=1;

FILE \*fp;

int i;

fp=fopen("num.txt","r");

if(fp==NULL)

{

printf("Can't open file");

isTrue=0;

}

for(i=0;!feof(fp);i++)

{

fscanf(fp,"%d",&num[i]);

}

fclose(fp);

for(i=0;i<NUM;i++) //检验所有磁道号是否合法，磁道号必须为大于零的整数

{

if(num[i]<0||num[i]==0)

{

printf("第%d个磁道号%d不合要求，请重新输入\n",i+1,num[i]);

isTrue=0;

}

}

if(isTrue)

{

printf("磁道号数组为:\n");

for(i=0;i<NUM;i++)

{

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

}

printf("\n");

}

printf("\n");

return isTrue;

}

/\*FCFS先来先服务算法\*/

void FCFS()

{

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* FCFS \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

int i,j;

float sum=0.0;

int k=0;

int move[NUM-1];

for(i=0,k=0;i<NUM-1,k<NUM-1;i++,k++) //寻道长度就等于磁道数组两两之差

{

move[k]=abs(num[i]-num[i+1]);

}

printf("下一个磁道号: 移动距离(磁道数):\n");

for(i=1,k=0;i<NUM,k<NUM-1;i++,k++)

{

sum+=move[k];

printf("\t%d",num[i]);

printf("\t\t%d\n",move[k]);

}

sum=(float)sum/(NUM-1);

printf("\nFCFS平均寻道长度为:%.1f\n",sum);

}

/\*SSTF最短寻道时间优先算法\*/

void SSTF()

{

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* SSTF \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

int i,j,t,p;

int l,r;

float sum=0.0;

int k=0;

int move[NUM-1],numsort[NUM],numSSTF[NUM];

for(i=0;i<NUM;i++)

numsort[i]=num[i];

for(i=0;i<NUM;i++) //磁道号从小到大排序

{

for(j=0;j<NUM-1;j++)

{

if(numsort[j]>numsort[j+1])

{

t=numsort[j];

numsort[j]=numsort[j+1];

numsort[j+1]=t;

}

}

}

for(i=0;i<NUM;i++) //找到第一个磁道号在排序数组里的位置为p

{

if(numsort[i]==num[0])

{

numSSTF[k]=num[0];

p=i;

l=i-1; //l为p前一个位置

r=i+1; //r为p后一个位置

}

}

for(k=1;k<NUM;k++) //以p为起点，比较前一个与后一个只差的绝对值大小，前面小则把l位置放入最终数组里，l--，往后同理

{

if(l>=0&&r<=NUM-1)

{

if(abs(numsort[l]-numsort[p])<=abs(numsort[p]-numsort[r]))

{

numSSTF[k]=numsort[l];

p--;

l--;

}

else

{

numSSTF[k]=numsort[r];

p++;

r++;

}

}

else if(l<0)

{

numSSTF[k]=numsort[r];

r++;

}

else if(r>NUM-1)

{

numSSTF[k]=numsort[l];

l--;

}

else

printf("error");

}

for(i=0,k=0;i<NUM-1,k<NUM-1;i++,k++)

{

move[k]=abs(numSSTF[i]-numSSTF[i+1]);

}

printf("下一个磁道号: 移动距离(磁道数):\n");

for(i=1,k=0;i<NUM,k<NUM-1;i++,k++)

{

sum+=move[k];

printf("\t%d",numSSTF[i]);

printf("\t\t%d\n",move[k]);

}

sum=(float)sum/(NUM-1);

printf("\nSSTF平均寻道长度为:%.1f\n",sum);

}

/\*SCAN扫描算法\*/

void SCAN() //与SSTF算法同理，只不过该算法直接指定一个方向向前或向后寻道

{

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* SCAN \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

int i,j,t,p,x;

int l,r;

float sum=0.0;

int k=0;

int move[NUM-1],numsort[NUM],numSCAN[NUM];

for(i=0;i<NUM;i++)

numsort[i]=num[i];

for(i=0;i<NUM;i++)

{

for(j=0;j<NUM-1;j++)

{

if(numsort[j]>numsort[j+1])

{

t=numsort[j];

numsort[j]=numsort[j+1];

numsort[j+1]=t;

}

}

}

printf("请选择: 1.磁道向内优先; 2.磁道向外优先\n");

printf("选择:");

scanf("%d",&x);

if(x==1)

{

for(i=0;i<NUM;i++)

{

if(numsort[i]==num[0])

{

numSCAN[k]=num[0];

p=i;

l=i-1;

r=i+1;

}

}

for(k=1;k<NUM;k++)

{

if(l>=0)

{

numSCAN[k]=numsort[l];

l--;

}

else if(r<=NUM-1)

{

numSCAN[k]=numsort[r];

r++;

}

else

printf("error");

}

for(i=0,k=0;i<NUM-1,k<NUM-1;i++,k++)

{

move[k]=abs(numSCAN[i]-numSCAN[i+1]);

}

printf("下一个磁道号: 移动距离(磁道数):\n");

for(i=1,k=0;i<NUM,k<NUM-1;i++,k++)

{

sum+=move[k];

printf("\t%d",numSCAN[i]);

printf("\t\t%d\n",move[k]);

}

sum=(float)sum/(NUM-1);

printf("\nSSTF平均寻道长度为:%.1f\n",sum);

}

else if(x==2)

{

for(i=0;i<NUM;i++)

{

if(numsort[i]==num[0])

{

numSCAN[k]=num[0];

p=i;

l=i-1;

r=i+1;

}

}

for(k=1;k<NUM;k++)

{

if(r<=NUM-1)

{

numSCAN[k]=numsort[r];

r++;

}

else if(l>=0)

{

numSCAN[k]=numsort[l];

l--;

}

else

printf("error");

}

for(i=0,k=0;i<NUM-1,k<NUM-1;i++,k++)

{

move[k]=abs(numSCAN[i]-numSCAN[i+1]);

}

printf("下一个磁道号: 移动距离(磁道数):\n");

for(i=1,k=0;i<NUM,k<NUM-1;i++,k++)

{

sum+=move[k];

printf("\t%d",numSCAN[i]);

printf("\t\t%d\n",move[k]);

}

sum=(float)sum/(NUM-1);

printf("\nSSTF平均寻道长度为:%.1f\n",sum);

}

}

int chooseMenu();

void selectMenu()

{

printf("请选择算法:\n");

printf("\t1.FCFS\n");

printf("\t2.SSTF\n");

printf("\t3.SCAN\n");

printf("\t0.退出\n");

chooseMenu();

}

int chooseMenu()

{

int i,j;

printf("选择:");

scanf("%d",&i);

switch(i)

{

case 0:

return 0;

case 1:

FCFS();

break;

case 2:

SSTF();

break;

case 3:

SCAN();

break;

default:

printf("输入有误，请重新输入\n");

chooseMenu();

break;

}

if(i==1||i==2||i==3)

{

printf("\n输入1返回。否则退出 :");

scanf("%d",&j);

if(j==1)

{

system("clear");

selectMenu();

}

else

return 0;

}

}

int main() {

openFile();

if(isTrue) {

system("clear");

selectMenu();

}

return 0;

}