**广州大学学生实验报告**

**开课学院及实验室：**计算机科学与工程实验室 **2019年10月31日**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **学院** | **计算机科学与网络工程学院** | **年级/专业/班** | **计科173** | **姓名** | 谢绍波 | **学号** | 1706100109 |
| **实验课程名称** | **操作系统实验** | | | | | **成绩** |  |
| **实验项目名称** | 实验五 磁盘管理实验 | | | | | **指导老师** | 张汛涞 |

**一、实验目的及要求**

**二、实验设备与平台**

1. 实验设备：计算机；

2. 平台：Ubuntu虚拟机

**三、实验内容与步骤**

一、实验目的

掌握几种基本的磁盘调度算法。

二、实验内容和要求

1、采用模拟先来先服务法（First-Come, First-Served，FCFS），最短寻道时间优先法（Shortest Seek Time First， SSTF），电梯法三种磁盘调度算法，对一组请求访问磁道序列输出为每种调度算法的磁头移动轨迹和移动的总磁道数。

2、请求访问磁道序列的数据放在文件里，数据间以空格间隔。

三、实验程序

#include<stdio.h>

#include<algorithm>

#include<fstream>

#include<vector>

#include<iostream>

using namespace std;

int init\_pos;

int len;

int count\_xsb;

int \*data;

vector<int>move\_xsb;

void read\_xsb();

void FCFS\_xsb();

void SSTF\_xsb();

void SCAN\_xsb();

void path\_xsb();

int main(){

    read\_xsb();

}

void read\_xsb(){

    fstream input("all\_data.txt",ios::in);

    if(!input){

        cerr << "File open error!\n";

    }else{

        int n;

        input >> n;

        while(n--){

            input >> init\_pos;

            input >> len;

            data = new int[len];

            for(int i = 0; i < len; i++){

                input>>data[i];

            }

            FCFS\_xsb();

            SSTF\_xsb();

            SCAN\_xsb();

        }

    }

}

void FCFS\_xsb(){

    puts("------FCFS------\n\n");

    count\_xsb=0;

    move\_xsb.clear();

    int step=init\_pos;

    for(int i = 0; i < len; i++){

        move\_xsb.push\_back(data[i]);

        count\_xsb += abs(step-data[i]);

        step = data[i];

    }

    path\_xsb();

}

void SSTF\_xsb(){

    puts("------SSTF------\n\n");

    count\_xsb = 0;

    move\_xsb.clear();

    int step=init\_pos;

    sort(data,data+len);

    int pos=init\_pos;

    int left;

    int right;

    for(int i = 0; i < len; i++){

        if(init\_pos>=data[i]){

            left=i-1;

            right=i;

            break;

        }

    }

    int flag=0;

    while(flag<len){

        if(right>=len||left>=0&&abs(data[right]-pos)>abs(data[left]-pos)){

            count\_xsb += abs(data[left]-pos);

            move\_xsb.push\_back(data[left]);

            pos = data[left];

            left--;

        }else{

            count\_xsb += abs(data[left]-pos);

            move\_xsb.push\_back(data[right]);

            pos = data[right];

            right++;

        }

        flag++;

    }

    path\_xsb();

}

void SCAN\_xsb(){

    puts("------SCAN------\n\n");

    count\_xsb = 0;

    move\_xsb.clear();

    int step=init\_pos;

    sort(data,data+len);

    int pos=init\_pos;

    int left;

    int right;

    for(int i = 0; i < len; i++){

        if(init\_pos>=data[i]){

            left=i-1;

            right=i;

            break;

        }

    }

    while(right<len){

        count\_xsb += abs(data[right]-pos);

        move\_xsb.push\_back(data[right]);

        pos = right;

        right++;

    }

    while(left>=0){

        count\_xsb += abs(data[left]-pos);

        move\_xsb.push\_back(data[left]);

        pos = left;

        left--;

    }

    path\_xsb();

}

void path\_xsb(){

    for(auto val:move\_xsb){

        cout << val << "-->";

    }

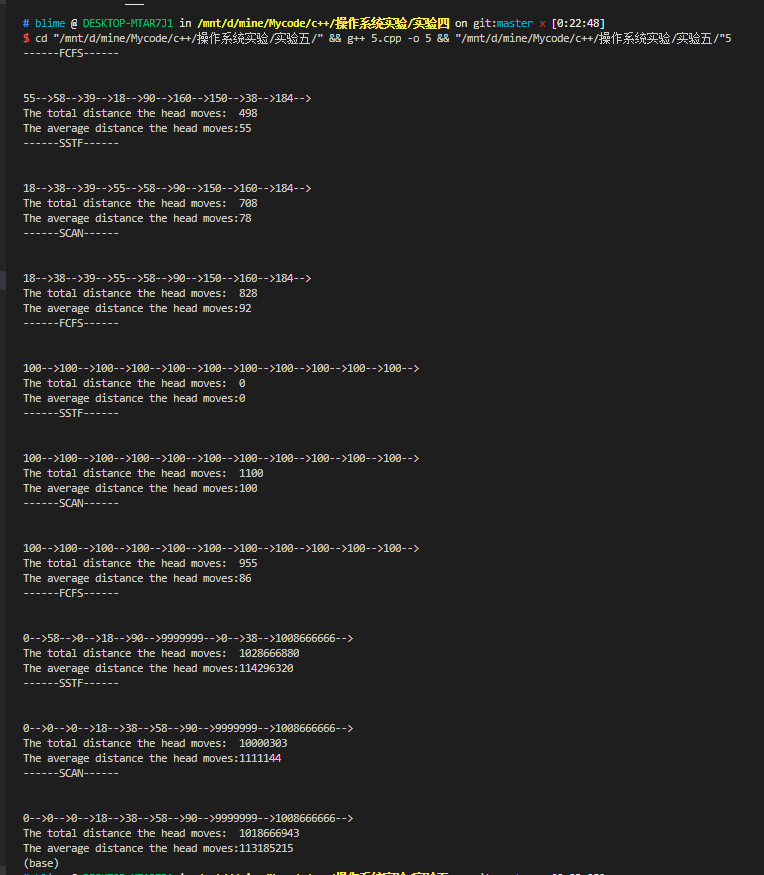
    cout<<endl;

    cout<<"The total distance the head moves:  "<<count\_xsb<<endl;

    cout<<"The average distance the head moves:"<<count\_xsb/len<<endl;

}

四、运行结果



五、思考和分析

**思考和分析：**

FCFS，就是简单的，读入，计算移动总/平均距离，输出

SSTF，就是先对数据进行排序，然后找到初始位置在排列后数据的位置，分成左右两部分，反复求左右与初始位置的距离，更新初始位置，递归求解。

SCAN，类似于SSTF，只是递归的条件改为不用比较，直接递归完一部分，再到另一部分。

心得：

经过本次实验，学习理解了几种基本的磁盘调度算法。并对每种磁盘调度算法都有了对比的理解，收获满满，受益匪浅。