Министерство науки и высшего образования РФ

Севастопольский государственный университет

Кафедра информатики и управления в технических системах

ОТЧЕТ

по лабораторной работе №3

ИНФОРМАЦИОННЫЕ СТРУКТУРЫ ОС. ТАБЛИЦА ДЕСКРИПТОРОВ ФАЙЛОВ

по дисциплине «Операционные системы»

Выполнил:

Студент группы ИВТ/б 22-о

Черняев Н.Г.

Проверил:

Корепанова Н.Л.

г. Севастополь 2019

**Цель работы:**

Изученить информационные структуры ОС, получить навыки обработки массивов данных и отладки программ циклической структуры.

**Вариант №4**

Задание:

Разработать функцию сортировки таблицы дескрипторов файлов по заданному ключу. Таблицу дескрипторов представить массивом структур. Дескриптор должен содержать имя файла (не более 8 символов), тип файла (не более 3 символов), дату создания (в формате чч.мм.гг), количество обращений (целое число), размер файла (целое число), время последней модификации(в формате час.мин). В главной программе предусмотреть ввод и вывод исходных данных и результатов, а так же обращение к функции сортировки, глобальные переменные не использовать.

Тип ключа: тип файла.

Метод сортировки: сортировка включениями.

**Текст программы:**

#include "pch.h"

#include <iostream>

using namespace std;

struct descriptor {

char file\_name[8];

char file\_type[3];

struct data {

int year;

int mon;

int day;

} d;

int file\_appeals;

int file\_size;

struct last\_mod {

int hours;

int min;

} lm;

};

void file\_output(const descriptor a[], int size) {

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

cout << "File name: " << a[i].file\_name << endl;

cout << "File type: " << a[i].file\_type << endl;

cout << "Data: " << a[i].d.day << "." << a[i].d.mon << "." << a[i].d.year << endl;

cout << "File appeals: " << a[i].file\_appeals << endl;

cout << "File size: " << a[i].file\_size << endl;

cout << "Last file modification: " << a[i].lm.hours << "." << a[i].lm.min << endl << endl;

};

};

void sorting( descriptor a[], int size) {

for (int i = 1; i < size; i++) {

int nmin = i;

descriptor tmp = a[i];

while ((tmp.file\_type[0] < a[nmin - 1].file\_type[0]) && (nmin > 0)) {

a[nmin] = a[nmin - 1];

nmin--;

}

a[nmin] = tmp;

};

};

int main()

{

int count;

descriptor files[200];

cout << "Enter the amount of files: " << endl;

cin >> count;

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

cout << "Enter file name " << endl; cin >> files[i].file\_name;

cout << "Enter file type " << endl; cin >> files[i].file\_type;

cout << "Enter data " << endl; cin >> files[i].d.day >> files[i].d.mon >> files[i].d.year;

cout << "Enter file appeals " << endl; cin >> files[i].file\_appeals;

cout << "Enter file size " << endl; cin >> files[i].file\_size;

cout << "Enter file last modification " << endl; cin >> files[i].lm.hours >> files[i].lm.min;

cout << endl;

};

file\_output(files, count);

cout << "SORTING \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* SORTING" << endl;

sorting(files, count);

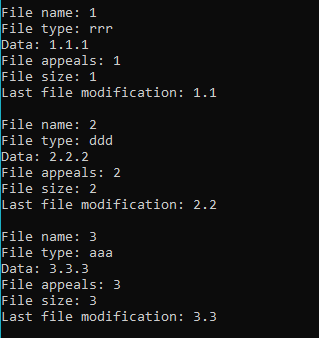
file\_output(files, count);

return 0;

}

**Тестовые примеры:**

Введем в программу следующее:



Тогда программа ,сортируя массив структур по типу файла, должна поставить 3-й файл на 1-ое место, а 1-ый файл на 3-ее место.

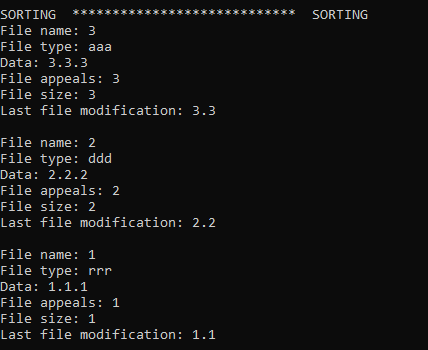


рис. 1 – результат работы программы

Вывод:

В данной лабораторной работе были изучены информационные структуры ОС, получены навыки обработки массивов данных и отладки программ циклической структуры.