Отчет

по лабораторной работе № 3

«Информационные структуры ОС. Таблица Дескрипторов файлов»

по дисциплине

ОПЕРАЦИОННЫЕ СИСТЕМЫ

Выполнил студент гр. ИС/б-22о

Горбенко К.Н.

Проверила:

Шалимова Е.М.

**Цель работы**: изучение информационных структур ОС, получение навыков обработки массивов данных и отладки программ циклической структуры.

* 1. Постановка задачи

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 также обращение к функции сортировки, глобальные переменные не использовать.

Задание для **варианта № 4**:

* тип ключа: тип файла;
* метод сортировки: включениями.
  1. Словесное описание алгоритма сортировки

Ввод массива array, N, I;

Если I < N, то п.3, иначе конец;

current = arrayi; j = I;

Если current < arrayi-1 и j >= 0, то п.5, иначе п.2;

arrayj = arrayj-1; j = j – 1; arrayj = current;

* 1. Описание используемых структур данных

В программе используются два типа структур. Первая представляет дату и содержит три целочисленных поля: день, месяц и год. Вторая представляет дескриптор файлов и содержит следующие поля: имя файла, тип файла, дата создания, дата последней модификации, количество обращений, размер файла.

* 1. Спецификации подпрограмм

В программе используются следующе подпрограммы:

FileDescriptor input\_one\_struct()

Функция используется для ввода одной структуры дескриптора файлов с клавиатуры. Она возвращает одну переменную типа дескриптора файлов.

void show\_stored\_records(FileDescriptor \* records, int numberOfRecords)

Подпрограмма используется для вывода всех записей из массива records.

void sort\_sequence(FileDescriptor \* records, int size, int sortedIndex)

Подпрограмма используется для сортировки массива структур records. Она принимает сам массив records, его размер size и номер sortedIndex, начиная с которого массив не отсортирован.

* 1. Результат работы программы

Введем одну структуру с типом файла txt и выведем на экран.

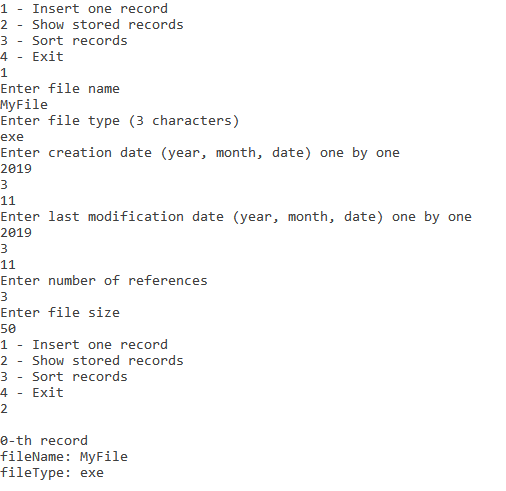


Рис. 1 – Ввод одной структуры и вывод ее на экран

Далее введем еще одну структуру с типом файла doc. Она должна оказаться в конце списка дескрипторов файлов:

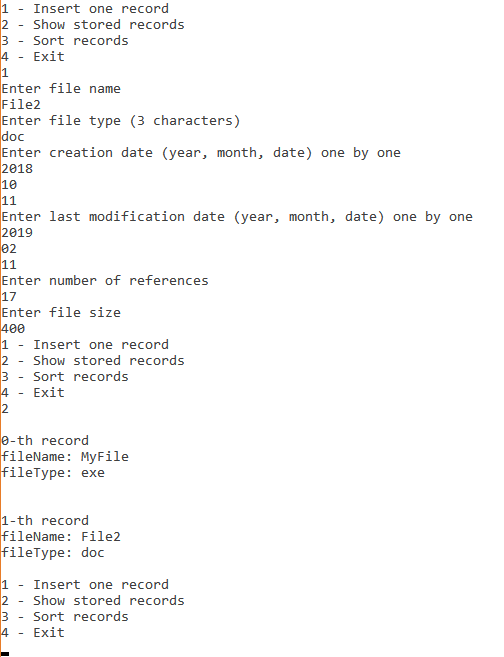


Рис. 2 – Ввод второй структуры и вывод ее на экран

Теперь выберем опцию сортировки списка структур. Структура с типом файла doc должна переместиться на первое место.

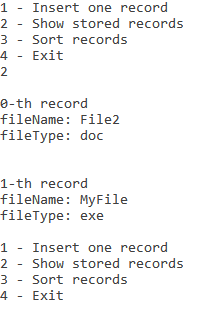


Рис. 3 – Вывод отсортированного списка структур

* 1. Текст программы

#include <pch.h>

#include <iostream>

#include <string.h>

#include <locale.h>

#include <clocale>

using namespace std;

struct Date {

int year;

int month;

int day;

};

struct FileDescriptor {

char fileName[8];

char fileType[3];

Date created;

Date lastModified;

int references;

int fileSize;

};

FileDescriptor input\_one\_struct() {

FileDescriptor data;

cout << "Enter file name" << endl;

cin >> data.fileName;

cout << "Enter file type (3 characters)" << endl;

cin >> data.fileType;

cout << "Enter creation date (year, month, date) one by one" << endl;

cin >> data.created.year;

cin >> data.created.month;

cin >> data.created.month;

cout << "Enter last modification date (year, month, date) one by one" << endl;

cin >> data.lastModified.year;

cin >> data.lastModified.month;

cin >> data.lastModified.day;

cout << "Enter number of references" << endl;

cin >> data.references;

cout << "Enter file size" << endl;

cin >> data.fileSize;

return data;

}

void show\_stored\_records(FileDescriptor \* records, int numberOfRecords) {

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

cout << endl << i << "-th record" << endl;

cout << "fileName: " << records[i].fileName << endl;

cout << "fileType: " << records[i].fileType << endl << endl;

}

}

void sort\_sequence(FileDescriptor \* records, int size, int sortedIndex)

{

for (int i = sortedIndex; i < size; i++)

{

FileDescriptor current = records[sortedIndex];

int j = i;

while (strcmp(current.fileType, records[j - 1].fileType) && j >= 1)

{

records[j] = records[j - 1];

j = j - 1;

records[j] = current;

}

}

}

int show\_menu() {

cout << "1 - Insert one record" << endl;

cout << "2 - Show stored records" << endl;

cout << "3 - Sort records" << endl;

cout << "4 - Exit" << endl;

int opted;

cin >> opted;

return opted;

}

int main() {

const int EXIT\_NUMBER = 4;

FileDescriptor \* data = new FileDescriptor[20];

int numberOfStructsStored = 0;

int sortedIndex = 1;

int userOption = show\_menu();

do {

switch (userOption) {

case 1:

data[numberOfStructsStored] = input\_one\_struct();

numberOfStructsStored += 1;

break;

case 2:

show\_stored\_records(data, numberOfStructsStored);

break;

case 3:

sort\_sequence(data, numberOfStructsStored, sortedIndex);

sortedIndex = numberOfStructsStored;

break;

}

userOption = show\_menu();

} while (userOption != EXIT\_NUMBER);

delete[] data;

}

* 1. Вывод

В ходе лабораторной работы били изучены структуры языка C. В отличие от массива, они позволяют содержать разнотипные данные. Для представления дескриптора файлов использовалась структура со следующими полями: имя файла, тип файла, дата создания файла, дата последнего изменения файла, количество обращений к файлу, размер файла.

Для сортировки структур использовался алгоритм сортировки вставками. Он предполагает возможность вставки структуры на нужную позицию в отсортированной части массива. Такой алгоритм дает возможность вводить новые записи в конец массива и вызывать сортировку только по вызову функции без лишних вычислительных затрат на полную сортировку.