**ЛАБОРАТОРНАЯ РАБОТА №2**

**«ПРОГРАММИРОВАНИЕ ОПЕРАЦИЙ НАД СТРУКТУРАМИ**

**И БИНАРНЫМИ ФАЙЛАМИ»**

**Цель работы**: Исследование особенностей обработки бинарных файлов, хранящих структурные типы данных.

1. **Вариант задания**

Описать структуру с именем STUDENT, содержащую следующие поля:

1. порядковый номер;
2. номер;
3. фамилия и имя;
4. год рождения;
5. год поступления в университет;
6. структура OCENKI, содержащая четыре поля: физика, математика, программирование, история.

Написать программу, выполняющую следующие действия с помощью функций:

1. ввод с клавиатуры данных в файл, состоящий из структур типа
2. STUDENT;
3. чтение данных из этого файла и вывод их на экран;
4. корректировку данных в файле по номеру записи;
5. вывод на дисплей фамилий и года рождения студентов, не получивших
6. ни одной тройки, а если таких студентов нет, вывести соответствующее сообщение
7. отсортировать записи по номеру.

**2. Алгоритм программы**



Рисунок 1 – Основное тело программы



Рисунок 2 - Алгоритм создания/добавления данных в файл



Рисунок 3 – Вывод записей на экран



Рисунок 4 – Сортировка записей по номеру



Рисунок 5 ­­­­­­­­­­­­­– Поиск и редактирование записи



Рисунок 6 – Поиск и вывод студентов без трое на экран

**3. Код программы на языке C**

#pragma warning(disable : 4996) ;

#include <stdio.h>

#include <string.h>

#include <conio.h>

#include <windows.h>

#include <stdlib.h>

#define clear() system("cls")

typedef struct {

int id;

char name[40];

char surname[40];

int birth\_date;

int stud\_date;

struct {

int fiz;

int math;

int it;

int history;

} marks;

} STudent;

const int SizeStudent = sizeof(STudent);

int create\_file(FILE\* f); //запись в файл

int sort\_file(FILE\* f); //сортировка файла

int print\_file(FILE\* f); //вывод файла

int getNewSelector(int selector, char c);

void MenuSelect(int selector, FILE\* f);

void printINFO(STudent d);

int correctInfo(FILE\* f);

int serchRed(FILE\* f);

char Menu[][30] = { "Запись в файл","Сортировка файла","Вывод файла","Редактировать запись","Найти студентов без троек","Выход"};

int main() {

char c; SetConsoleCP(1251); // Задаем таблицу символов для консоли.

SetConsoleOutputCP(1251);

//Открытие существующего файла для чтения и записи в конец

FILE\* f = fopen("data.dat", "rb+");

if (!f) {

//Создание нового файла для обновления

f = fopen("data.dat", "wb+");

if (!f) {

puts("Не могу открыть (создать) файл\n");

return 1;

}

}

//вывод меню и запуск соответствующих функций

int selector = 1;

while (1) {

clear();

puts("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

for (int i = 0; i < 6; i++)

{

if (i + 1 == selector) printf("--> %s\n", Menu[i]);

else printf("%s\n", Menu[i]);

}

puts("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

puts("Переключение стрелками , enter - выбрать\n");

c = getch();

selector = getNewSelector(selector, c, 6);

if (c == 13) {

if (selector == 6) return 0;

MenuSelect(selector, f);

}

}

}

int getNewSelector(int selector, char c, int selectMax)

{

switch (c)

{

case 72://вверх

if (selector > 1) selector--;

break;

case 80://вниз

if (selector < selectMax) selector++;

break;

}

return selector;

}

void MenuSelect(int selector, FILE\* f)

{

switch (selector) {

case 1:

create\_file(f);

break;

case 2:

sort\_file(f);

break;

case 3:

print\_file(f);

break;

case 4:

correctInfo(f);

break;

case 5:

serchRed(f);

break;

}

}

void printINFO(STudent d)

{

printf("Личный номер --> %d\n", d.id);

printf("Фамилия --> %s\n", d.surname);

printf("Имя --> %s\n", d.name);

printf("Год рождения --> %d\n", d.birth\_date);

printf("Год поступления --> %d\n", d.stud\_date);

puts("--------------------Оценки--------------------");

printf("\t Физика --> %d\n", d.marks.fiz);

printf("\t Математика --> %d\n", d.marks.math);

printf("\t Программирование --> %d\n", d.marks.it);

printf("\t История --> %d\n", d.marks.history);

puts("----------------------------------------------");

}

int create\_file(FILE\* f) {

STudent d; clear();

char Menu[][30] = { "Добавить запись","Выход" };

int selector = 1; int flag = 0;

fseek(f, 0, SEEK\_END); //указатель в конец файла

while (1) {

clear();

puts("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

for (int i = 0; i < 2; i++)

{

if (i + 1 == selector) printf("--> %s\n", Menu[i]);

else printf("%s\n", Menu[i]);

}

puts("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

if (flag)

{

puts("-----------Последняя введенная запись-----------");

printINFO(d);

puts("------------------------------------------------");

}

puts("Переключение стрелками , enter - выбрать\n");

char c = getch();

selector = getNewSelector(selector, c, 2);

if (c == 13)

switch (selector) {

case 1:

{

puts("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

printf("Личный номер -->"); scanf("%d", &d.id);

printf("Фамилия -->"); scanf("%s", &d.surname);

printf("Имя -->"); scanf("%s", &d.name);

printf("Год рождения -->"); scanf("%d", &d.birth\_date);

printf("Год поступления -->"); scanf("%d", &d.stud\_date);

puts("----------------Оценки----------------");

printf("Физика -->"); scanf("%d", &d.marks.fiz);

printf("Математика -->"); scanf("%d", &d.marks.math);

printf("Программирование -->"); scanf("%d", &d.marks.it);

printf("История -->"); scanf("%d", &d.marks.history);

puts("--------------------------------------");

fwrite(&d, SizeStudent, 1, f); flag = 1;

}

break;

case 2:

return 1;

}

}

}

int sort\_file(FILE\* f) {

long i, j; clear();

STudent d1, d2;

fseek(f, 0, SEEK\_END); //указатель в конец

long len = ftell(f) / SizeStudent;//определяем количество записей

rewind(f); //указатель в начало

//пузырьковая сортировка

for (i = len - 1; i >= 1; i--)

for (j = 0; j <= i - 1; j++) {

fseek(f, j \* SizeStudent, SEEK\_SET); //указатель на j-ую запись

fread(&d1, SizeStudent, 1, f); //читаем запись j в d1

fread(&d2, SizeStudent, 1, f); //читаем след. запись в d2

if (d1.id > d2.id) {

//указатель на 2 поз. назад

fseek(f, (-2) \* SizeStudent, SEEK\_CUR);

//обмен значений

fwrite(&d2, SizeStudent, 1, f); //сначала записываем d2

fwrite(&d1, SizeStudent, 1, f); // затем записываем d1

}

}

puts("Сортировка успешна завершена");

getch();

return 0;

}

int print\_file(FILE\* f) {

STudent d; clear();

int n; int Posid = 1;

rewind(f); //указатель в начало файла

puts("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Весь список студентов \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

do {

n = fread(&d, SizeStudent, 1, f);//чтение структуры из файла

if (n < 1) break; //если n<1, то конец файла

printf("Порядковый номер --> %d\n", Posid);

printINFO(d);

puts("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

Posid++;

} while (1);

puts("Нажмите любую клавишу");

getch();

return 0;

}

int correctInfo(FILE\* f)

{

char Menu[][40] = { "Редактировать","Сохранить","Найти по номеру","Выйти в главное меню"};

int selector = 1; int n; int MinSelector = 1; int MaxSelection = 4;

STudent d; d.id = NULL; int Position = 0;

while (1)

{

int k; int Posid = 1; clear();

if (d.id != NULL)

{

puts("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Текущая запись\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

printINFO(d);

puts("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

}

puts("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

for (int i = MinSelector-1; i < MaxSelection; i++)

{

if (i + 1 == selector) printf("--> %s\n", Menu[i]); else printf("%s\n", Menu[i]);

}

puts("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

puts("Переключение стрелками , enter - выбрать\n");

char c = getch();

selector = getNewSelector(selector, c, MaxSelection);

if (c == 13 )

switch(selector)

{

case 1:

{

puts("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

printf("Личный номер -->"); scanf("%d", &d.id);

printf("Фамилия -->"); scanf("%s", &d.surname);

printf("Имя -->"); scanf("%s", &d.name);

printf("Год рождения -->"); scanf("%d", &d.birth\_date);

printf("Год поступления -->"); scanf("%d", &d.stud\_date);

puts("----------------Оценки----------------");

printf("Физика -->"); scanf("%d", &d.marks.fiz);

printf("Математика -->"); scanf("%d", &d.marks.math);

printf("Программирование -->"); scanf("%d", &d.marks.it);

printf("История -->"); scanf("%d", &d.marks.history);

puts("--------------------------------------");

break;

}

case 2:

{

fseek(f, (Position - 1) \* SizeStudent, SEEK\_SET);

fwrite(&d, SizeStudent, 1, f);

}

break;

case 3: {

int SerchID; printf("Введите идентификационный номер -->"); scanf("%d", &SerchID);

int flag = 0; Posid = 1; fseek(f, 0, SEEK\_SET);

do {

n = fread(&d, SizeStudent, 1, f);//чтение структуры из файла

if (n < 1) break; //если n<1, то конец файла

if (d.id == SerchID)

{

Position = Posid;

flag = 1;

break;

}

Posid++;

} while (1);

if (!flag) { printf("Не найден студент с таким идентификатором .... Нажмите любую кнопку"); getch(); }

break; }

case 4: return 0;

}

}

}

int serchRed(FILE\* f)

{

STudent d; clear();

int n; int Posid = 1; int flag = 0;

rewind(f); //указатель в начало файла

puts("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Список студентов без троек\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

do {

n = fread(&d, SizeStudent, 1, f);//чтение структуры из файла

if (n < 1) break; //если n<1, то конец файла

if ((d.marks.fiz > 3) && (d.marks.math > 3) && (d.marks.history > 3) && (d.marks.it > 3))

{

flag = 1;

printf("Порядковый номер --> %d\n", Posid);

printINFO(d);

Posid++;

}

puts("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

} while (1);

if (!flag) puts("Все печально .... Студентов без троек нет (");

puts("Нажмите любую клавишу");

getch();

return 0;

}

**4. Результаты тестирования и отладки программы**

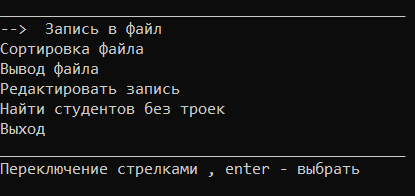


Рисунок 7 – Меню выбора действий

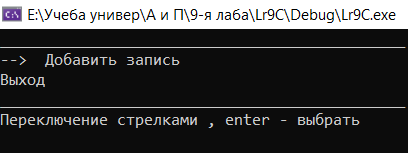


Рисунок 8 – Меню добавления записи

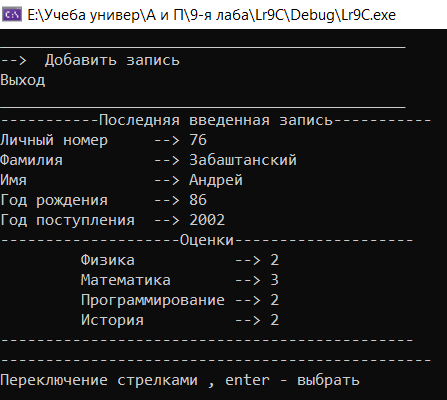


Рисунок 9 – Форма после добавления записи через нее

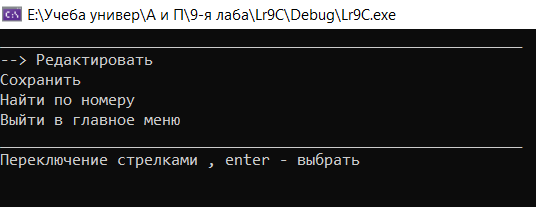


Рисунок 10 – Меню поиска и редактирования записей

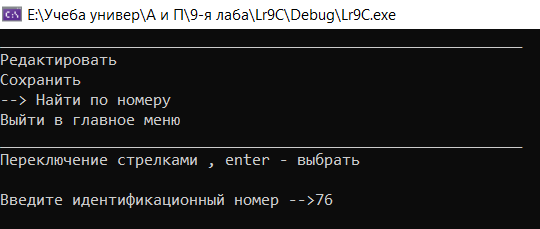


Рисунок 11 – Реализация поиска по номеру

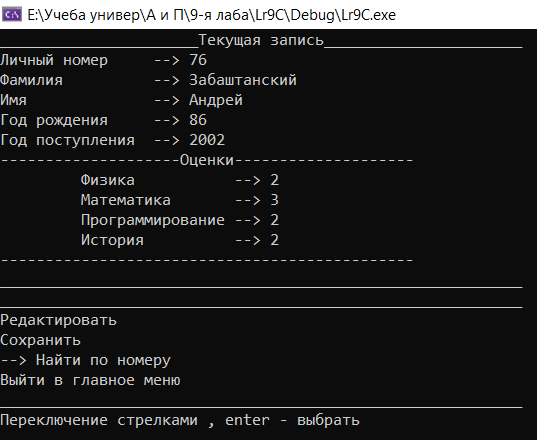


Рисунок 12 – Меню после после того как запись с таким номером найдена

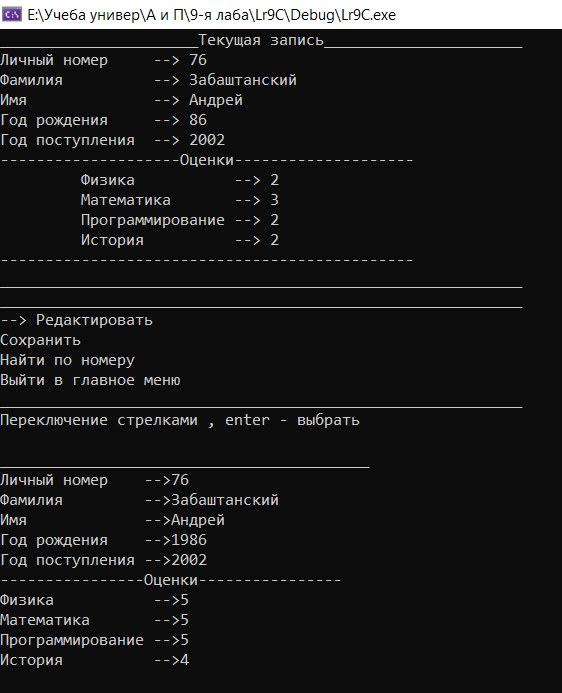


Рисунок 13 – Редактирование записи

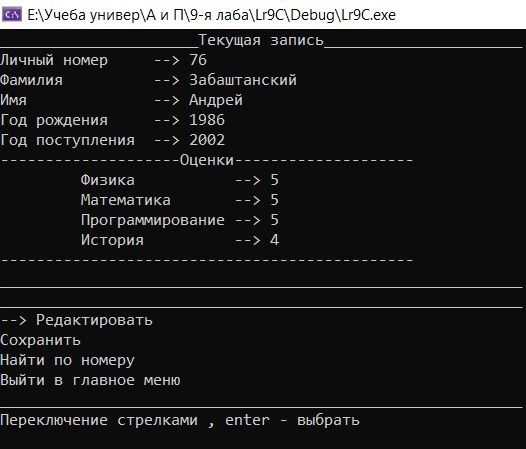


Рисунок 14 – Меню с уже отредактированной записью.

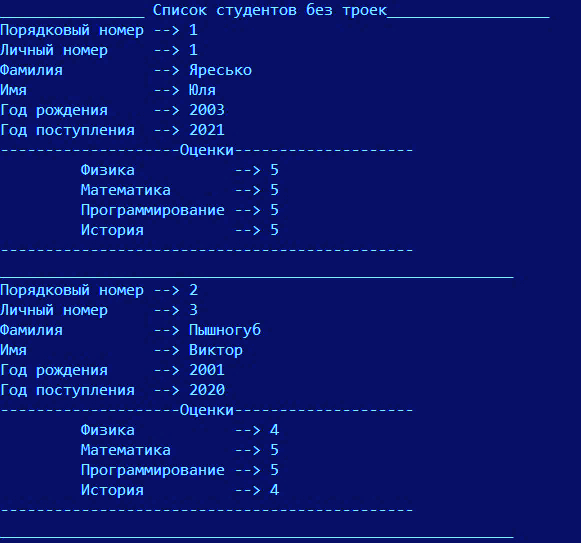


Рисунок 15 – Вывод всех студентов без троек

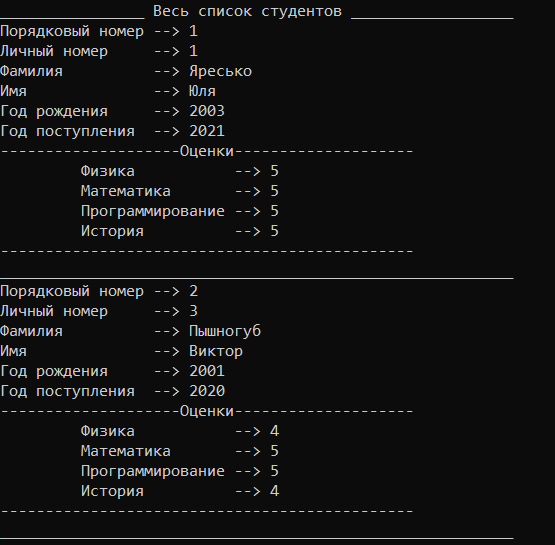


Рисунок 17 – Вывод всего файла на экран

**Вывод**

В ходе выполнения данной лабораторной работы были изучены основные операции над бинарными файлами, принцип работы с потоком ввода-вывода для бинарных файлов, обработки бинарной информации полученной из файла в языке С, с учетом связи указателей. Получены практические навыки работы с файловыми указателями, реализации алгоритмов обработки данных на прямую из бинарного файла в языке С. Реализован принцип минималистического графического меню для работы в программе.