Министерство образования и науки Российской Федерации

Севастопольский государственный университет

Кафедра ИС

Отчет

по лабораторной работе № 9

«Программирование операций над

структурами и бинарными файлами»

Выполнил

ст. гр. И12д

Серегин А.В.

Проверил:

асс. Забаштанский А.К.

Севастополь

2015

1. Цель работы

Изучение способов описания структур данных на языке С. Исследование особенностей обработки бинарных файлов, хранящих структурные типы данных.

2. Варианты заданий

Вариант 20:

Описать структуру с именем ORDER, содержащую следующие поля:

-расчетный счет плательщика;

-расчетный счет получателя;

-перечисляемая сумма.

Написать программу, выполняющую следующие действия с помощью функций:

-ввод с клавиатуры данных в файл, состоящий из элемента типа ORDER; записи должны быть размещены в алфавитном порядке по расчетным счетам плательщиков;

-чтение данных из этого файла;

-вывод на экран информации о сумме, снятой с расчетного счета плательщика, введенного с клавиатуры;

-если такого расчетного счета нет, выдать на дисплей соответствующее сообщение.

3. Структурная схема

На рисунках ниже представлены структурные схемы функций программы.



Рисунок 1 – Функция PrintAll.



Рисунок 2 – Функция Search.



Рисунок 3 – Функция Add.



Рисунок 4 – Функция LoadFile.



Рисунок 5 – Функция main.



Рисунок 6 – Функция NewOrder.

4. Код программы

#include <iostream>

#include <fstream>

#include <iomanip>

#include <cstdio>

#include <conio.h>

using namespace std;

struct order

{

char id\_1[8]; //расчетны счет плательщика

char id\_2[8]; //расчетный счет получателя

unsigned int summ; //сумма перевода

};

size\_t size\_order = sizeof(order);

//создание новой записи о переводе

order NewOrder()

{

system("cls");

order new\_order;

cout << "Введите расчетный счет плательщика:" << endl;

cin >> new\_order.id\_1;

cout << "Введите расчетный счет получателя:" << endl;

cin >> new\_order.id\_2;

cout << "Введите сумму перевода:" << endl;

cin >> new\_order.summ;

return new\_order;

}

//добавление нового перевода в файл в алф. порядке

void Add(FILE\* f,const order& O)

{

order temp;

int pos = 0;

bool flag = false; //вставка в середине или конце

fseek(f, 0, 2);

unsigned long int file\_length = ftell(f) / size\_order;

for (pos = file\_length-1; pos >= 0; pos--)

{

fseek(f, pos\*size\_order, SEEK\_SET);

fread(&temp, size\_order, 1, f);

fseek(f, pos\*size\_order, SEEK\_SET);

if (strcmp(temp.id\_1, O.id\_1) > 0)

{

fwrite(&O, size\_order, 1, f);

fwrite(&temp, size\_order, 1, f);

}

else

{

fwrite(&temp, size\_order, 1, f);

fwrite(&O, size\_order, 1, f);

flag = true;

break;

}

}

if (!flag) //если не нашел место в ср. или конце

{

fseek(f, 0, SEEK\_SET);

fwrite(&O, size\_order, 1, f);

}

system("pause");

}

//вывод данных файла на экран

void PrintAll(FILE\* f)

{

system("cls");

fseek(f, 0, 0);

order temp;

cout.setf(ios::left);

cout << "Плательщик Получатель Сумма" << endl;

while (fread(&temp, size\_order, 1, f))

cout << setw(11) << temp.id\_1 << setw(11)

<< temp.id\_2 << temp.summ << endl;

system("pause");

}

//загрузка файла

void Loadfile(FILE\*\* f)

{

\*f = fopen("input", "r+b");

if (!(\*f))

\*f = fopen("input", "w+b");

}

//поиск расчетных счетов в файле по плательщику.

void Search(FILE \*f)

{

system("cls");

order temp;

char str[8];

cout << "Введите расчетный счет плательщика:" << endl;

cin >> str;

system("cls");

bool empty = true;

cout.setf(ios::left);

cout << "Плательщик Получатель Сумма" << endl;

fseek(f, 0, 0);

while (fread(&temp, size\_order, 1, f))

{

if (!strcmp(temp.id\_1, str))

{

cout << setw(11) << temp.id\_1 << setw(11)

<< temp.id\_2 << temp.summ << endl;

empty = false;

}

}

if (empty)

cout << "Расчетный счет не найден." << endl;

system("pause");

}

int main()

{

setlocale(LC\_ALL, "Russian");

FILE\* f;

Loadfile(&f);

char key; //клавиша

do

{

system("cls");

cout << "1.Add 2.PrintAll 3.Search Esc.Exit" << endl;

key = \_getch();

switch (key)

{

case '1': Add(f, NewOrder()); break;

case '2': PrintAll(f); break;

case '3': Search(f); break;

}

} while (key != 27);

fclose(f); //закрытие и сохранение файла

system("cls");

system("pause");

return 0;

}

5. Тестовые примеры

На рисунках ниже представлены тестовые примеры работы программы.

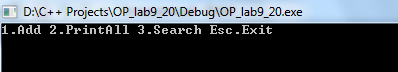


Рисунок 7 – Главное меню.

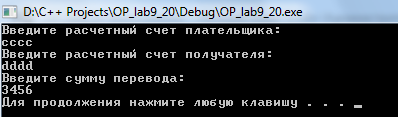


Рисунок 8 – Добавление записи.

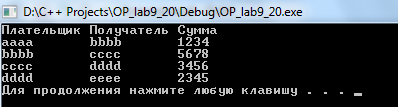


Рисунок 9 – Вывод всего списка.

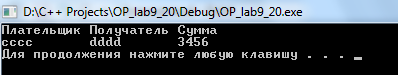


Рисунок 10 – Поиск записи “cccc”.

Выводы

С помощью структур данных можно легко хранить различную информацию в бинарных файлах и считывать сразу всю структуру из файла. Также запись можно легко объединить в массив, что позволяет быстро работать с данными всего списка, в отличии от работы напрямую с файлом.