Отчет по КП № 6 по курсу “Фундаментальная информатика”

Студент группы М80-103Б-21 Трофимов Владислав Олегович, № по списку 20

Контакты e-mail:trofimov.vlad-1234@yandex.ru
Github: https://github.com/ant1hype1488/labs

Работа выполнена: «08» апреля 2022г.

Преподаватель: каф. 806 Севастьянов Виктор Сергеевич

Отчет сдан « » _____20__ г., итоговая оценка _____

Подпись преподавателя _____

1. **Тема:** Обработка последовательной файловой структуры на языке Си

1. **Цель работы:** Разработать последовательную структуру данных для представления простейшей базы данных на файлах в СП Си.
2. **Задание** (*вариант №* **16** : Найти фамилии лучших студенток курса(не имеющих отметок ниже четырех и по сумме баллов не уступающих другим студентам своей группы)

3. **Оборудование** (студента):
Процессор *Intel Core i5-8265U @ 8x 3.9GH* с ОП *7851* Мб, НМД *1024* Гб. Монитор *1920x1080*

4. **Программное обеспечение (**студента**):**
Операционная система семейства: *linux*, наименование: *ubuntu*, версия *18.10 cosmic*
интерпретатор команд: *bash* версия *4.4.19.*
Система программирования -- версия --**,** редактор текстов *emacs* версия *25.2.2*
Утилиты операционной системы --
Прикладные системы и программы --
Местонахождение и имена файлов программ и данных на домашнем компьютере --

6. Идея, метод, алгоритм

Программа имеет 4 функции:

1.Показать всех студентов.
Программа читает файл в бинарном виде и выводит всех студентов в таблице
2.Добавить студента

Программа отрывает файл с флагом **ab(**то есть дополняет к существующим записям) и вписывает в файл в бинарном виде студента.
3.**Показать лучших студентов.**
Сначала программа открывает файл и каждую запись вставляет в динамический массив .После программа пробегается по каждой записи этого массива, а также создается новый динамический массив, в котором будут хранятся пройденные группы студентов. Допустим, программа встречает какую-то группу. Если массив групп пуст, то программа добавляет ее в пустой массив групп и пробегается по всем студентам с условием, что он должен быть из соответствующей группы. Если массив групп не пуст – программа проверяет ,есть эта запись в массиве групп :если есть ,то запись пропускается, если нет- проводит те же действия, как если бы массив групп был пуст.
4.Пересоздать таблицу
Таблица студентов создается/пересоздается случайным образом, используя готовый массив фамилий студентов.

7. Сценарий выполнения работы

func.h

```
#ifndef KP6_FUNCS_H
#define KP6_FUNCS_H
typedef struct
{

    char family[20];
    char sign[10];
    char sex[5];
    int group;
    int grades[3];

} Students;

void addStudent();
void showStudents();
void randomStudents(int n);
void bestStudents();

#endif //KP6_FUNCS_H
```

showStudents.c

```
#include "funcs.h"
#include <stdio.h>
#include <stdlib.h>

void showStudents(){
    FILE *fptr;
    Students student;

    if ((fptr = fopen("data.bin","rb")) == NULL){
        printf("Error! opening file");

        // Program exits if the file pointer returns NULL.
        exit(1);
    }
    printf("Группа|\t\tФамилия                |\tИнициалы\t|\tПол\t|\tОценка1\t|\tОцен
ка2\t|\tОценка3\t|\n");

    while (fread(&student, sizeof(Students), 1, fptr)==1)
    {
        printf("------|----------------------------|---------------|-------|----------
-|-----------|-----------|\n");
        printf("%d    |\t\t%20s\t|%10s     |\t %s   |\t\t %d   |\t\t %d   |\t\t %d   |\n",
student.group,student.family,
student.sign,student.sex,student.grades[0],student.grades[1],student.grades[2]);
    }

    fclose(fptr);
}
```

randomStudents.c

```c
#include <stdio.h>
#include <stdlib.h>
#include "string.h"
#include "funcs.h"
#include <time.h>

void randomStudents(int n){
    FILE *fptr;
    Students student;
    int stime;
    long ltime;
    char sign1[3];
    char sign2[3];

    char names[][20] = {
            "Vlad",
            "Vladislav",
            "Trofimov",
            "Ilya",
            "Vlad Trofimov",
            "Murad Ahab",
            "Gudok Slatt",
            "Playboi Carti",
            "Lil Uzi Vert",
            "Vitalik Buterin",
            "Zamay",
            "Gnoyniy",
            "Sonya Marmevadoda",
            "Verhovenskiy",
            "Myshkin",
            "Kunteynir",
            "Putin",
            "Velikiy",
            "Moses",
            "Django"
    };
    if ((fptr = fopen("data.bin","wb")) == NULL){
        printf("Error! opening file");

        // Program exits if the file pointer returns NULL.
        exit(1);
    }
    ltime = time (NULL);
    stime = (unsigned int) ltime/2;
    srand(stime);
    for (int i = 0; i < n; ++i) {
        student.group = 100 + rand()%9;
        int nameNum = rand()%20;
        strcpy(student.family,names[nameNum]);
        sign1[0]= names[nameNum][rand()%4];
        sign1[1]='\0';
        sign2[0]= names[nameNum][rand()%4];
        sign2[1]='\0';
```

```
        strcpy(student.sign, strcat(sign1,sign2));
        if (rand()%2 == 1){
            strcpy(student.sex,"m");
        } else{
            strcpy(student.sex,"w");
        }

        student.grades[0] = 1+rand()%5;
        student.grades[1] = 1+rand()%5;
        student.grades[2] = 1+rand()%5;
        fwrite(&student, sizeof(Students), 1, fptr);

    }
    fclose(fptr);
}
```

addStudent.c

```c
#include "funcs.h"
#include <stdio.h>
#include <stdlib.h>
#include "string.h"
void addStudent(){

    FILE *fptr;
    Students student;
    scanf("%d",&student.group);
    char temp1[20];

    scanf("%s",temp1);
    strcpy(student.family,temp1);
    char temp2[20];

    scanf("%s",temp2);
    strcpy(student.sign,temp2);

    char temp3[20];
    scanf("%s",temp3);
    strcpy(student.sex,temp3);
    scanf("%d ",&student.grades[0]);
    scanf("%d",&student.grades[1]);
    scanf("%d",&student.grades[2]);
    if ((fptr = fopen("data.bin","ab")) == NULL){
        printf("Error! opening file");

        // Program exits if the file pointer returns NULL.
        exit(1);
    }
    fwrite(&student, sizeof(Students), 1, fptr);
    fclose(fptr);

}
```

bestStudents.c

```c
#include <stdio.h>
#include <stdlib.h>
#include "string.h"
#include "funcs.h"




void bestStudents(){
    FILE *fptr;

    Students student;
    int groupCount = 0;
    int max = 0;
    int p;
    int exist = 0;
    int *groups;

    int n = 0;
    Students *all = (Students*)malloc(n*sizeof(Students));
    if ((fptr = fopen("data.bin","rb")) == NULL){
        exit(1);
    }

    while (fread(&student, sizeof(Students), 1, fptr)==1)
    {
        n++;
        all = (Students *)realloc(all, sizeof(Students) * n);
        *(all+n-1) = student;
    }
    fclose(fptr);
    for (int i = 0; i < n; ++i) {
        if (groupCount == 0) {
            groupCount = 1;
            groups = (int*) malloc(sizeof(int)*groupCount);
            *(groups + groupCount-1) = all[i].group;
            for (int j = 0; j < n; ++j) {
                if (all[j].group == all[i].group){
                    if ((all[j].grades[0] + all[j].grades[1]+ all[j].grades[2] > max)
&&all[j].grades[0] >= 4 && all[j].grades[1] >= 4&& all[j].grades[2] >= 4) {
                        max = all[j].grades[0] + all[j].grades[1]+ all[j].grades[2];
                        p = j;
                    }
                }
            }
            if (max!=0) {
                printf("Лучший студент группы %d - %s\n", all[p].group, all[p].family);
                max = 0;
            }

        }
```

```
else{
            for (int j = 0; j < groupCount; ++j) {
                if (all[i].group ==  groups[j]){
                    exist = 1;
                }
            }
            if (exist == 0){

                groupCount ++;

                groups = (int *)realloc(groups, groupCount * sizeof(int));
                *(groups + groupCount -1) = all[i].group;
                for (int j = 0; j < n ; ++j) {
                    if ( all[j].group ==  all[i].group){
                        if ((all[j].grades[0] + all[j].grades[1]+ all[j].grades[2] >
max) &&all[j].grades[0] >= 4 && all[j].grades[1] >= 4&& all[j].grades[2] >= 4) {
                            max = all[i].grades[0] + all[i].grades[1]+
all[i].grades[2];
                            p = j;
                        }
                    }
                }
                if (max!=0) {
                    printf("лучший студент группы %d - %s\n", all[p].group,
all[p].family);
                    max = 0;

                }
            } else{
                exist = 0;

            }
        }
    }
    free(groups);
}
```

main.c

```c
#include <stdio.h>
#include "funcs.h"

int main()
{
    int n;
    int ans = 0;
    while (ans!=-1){
        printf("1.Все студенты\n2.Добавить студента\n3.Вывести лучших
студентов\n4.Пересоздать таблицу\n");
        scanf("%d",&ans);
        switch (ans) {
            case 1:
                showStudents();
                break;
            case 2:
                addStudent();
                break;
            case 3:

                bestStudents();
                break;
            case 4:
                printf("Кол-во студентов: ");
                scanf("%d",&n);
                randomStudents(n);
                break;
            default:
                ans = -1;
                break;
        }
    }

    return 0;
}
```

8. Распечатка протокола

```
1.Все студенты
2.Добавить студента
3.Вывести лучших студентов
4.Пересоздать таблицу
4
Кол-во студентов: 15
1.Все студенты
2.Добавить студента
3.Вывести лучших студентов
4.Пересоздать таблицу
1
Группа|     Фамилия                  |   Инициалы    |   Пол |   Оценка1 |   Оценка2 |   Оценка3 |
------|------------------------------|---------------|-------|-----------|-----------|-----------|
104   |                  Velikiy     |        iV     |    m  |        3  |        3  |        1  |
------|------------------------------|---------------|-------|-----------|-----------|-----------|
106   |                Kunteynir     |        nt     |    m  |        1  |        5  |        1  |
------|------------------------------|---------------|-------|-----------|-----------|-----------|
105   |                    Zamay     |        Za     |    m  |        2  |        2  |        5  |
------|------------------------------|---------------|-------|-----------|-----------|-----------|
101   |             Verhovenskiy     |        hV     |    w  |        4  |        1  |        2  |
------|------------------------------|---------------|-------|-----------|-----------|-----------|
101   |          Vitalik Buterin     |        tV     |    w  |        1  |        1  |        1  |
------|------------------------------|---------------|-------|-----------|-----------|-----------|
100   |                   Moses      |        Mo     |    w  |        1  |        5  |        2  |
------|------------------------------|---------------|-------|-----------|-----------|-----------|
107   |                    Ilya      |        aI     |    w  |        3  |        1  |        5  |
------|------------------------------|---------------|-------|-----------|-----------|-----------|
102   |                Vladislav     |        dl     |    m  |        3  |        3  |        5  |
------|------------------------------|---------------|-------|-----------|-----------|-----------|
101   |            Vlad Trofimov     |        VV     |    m  |        2  |        3  |        5  |
------|------------------------------|---------------|-------|-----------|-----------|-----------|
108   |                    Zamay     |        aa     |    m  |        4  |        3  |        2  |
------|------------------------------|---------------|-------|-----------|-----------|-----------|
105   |            Playboi Carti     |        yP     |    m  |        4  |        2  |        4  |
------|------------------------------|---------------|-------|-----------|-----------|-----------|
107   |                    Ilya      |        aI     |    w  |        3  |        1  |        5  |
------|------------------------------|---------------|-------|-----------|-----------|-----------|
102   |                Vladislav     |        dl     |    m  |        3  |        3  |        5  |
------|------------------------------|---------------|-------|-----------|-----------|-----------|
101   |            Vlad Trofimov     |        VV     |    m  |        2  |        3  |        5  |
------|------------------------------|---------------|-------|-----------|-----------|-----------|
108   |                    Zamay     |        aa     |    m  |        4  |        3  |        2  |
------|------------------------------|---------------|-------|-----------|-----------|-----------|
105   |            Playboi Carti     |        yP     |    m  |        4  |        2  |        4  |
------|------------------------------|---------------|-------|-----------|-----------|-----------|
106   |                  Velikiy     |        VV     |    m  |        3  |        5  |        3  |
------|------------------------------|---------------|-------|-----------|-----------|-----------|
108   |                    Vlad      |        dl     |    m  |        3  |        4  |        2  |
------|------------------------------|---------------|-------|-----------|-----------|-----------|
103   |             Verhovenskiy     |        er     |    w  |        3  |        5  |        4  |
------|------------------------------|---------------|-------|-----------|-----------|-----------|
100   |                  Velikiy     |        el     |    m  |        5  |        4  |        3  |
1.Все студенты
2.Добавить студента
3.Вывести лучших студентов
4.Пересоздать таблицу
2
103 Vladik vl m 5 5 5
1.Все студенты
2.Добавить студента
3.Вывести лучших студентов
4.Пересоздать таблицу
3
лучший студент группы 103 - Vladik
1.Все студенты
2.Добавить студента
3.Вывести лучших студентов
4.Пересоздать таблицу
```

9. Дневник отладки должен содержать дату и время сеансов отладки и основные события (ошибки в сценарии и программе, нестандартные ситуации) и краткие комментарии к ним. В дневнике отладки приводятся сведения об использовании других ЭВМ, существенном участии преподавателя и других лиц в написании и отладке программы.

№	Лаб. или дом.	Дата	Время	Событие	Действие по исправлению	Примечание

10. Замечания автора по существу работы

Работу считаю актуальной :она помогает поближе познакомиться с обработкой последовательных файловых структур

11. Выводы

Благодаря данный работе я научился **р**азработывать последовательную структуру данных для представления простейшей базы данных на файлах в СП Си.

Подпись студента _____