Отчёт по лабораторной работе №12

по дисциплине
«Информатика»

Студент: Еремян Р.А.
Группа: ФН4-11Б
Преподаватель: Евдокимова К.В.

Москва
, 2021

Оглавление

Теоретическая часть **3**

Постановка задачи 3

Используемые конструкции языка 3

Практическая часть **4**

Входные и выходные данные 5

Реализованные функции 5

Примеры работы программы 11

Текст программы 16

Теоретическая часть

Постановка задачи

Написать консольное приложение на языке Си, в котором пользователю предоставляется меню с вариантами выбора действий:

1. Добавить в конец бинарного файла книгу/музыку/фильм;
2. Вывести книги/музыку/фильмы на экран;
3. Отсортировать книги/музыку/фильмы (по возрастанию по одному из полей);
4. Удалить книгу/музыку/фильм;
5. Выход.

Вариант 20 предусматривает работу с музыкой, сортировку по названию, удаление по автору.

Используемые конструкции языка

К базовым конструкциям языка С относятся:

- алфавит;
- константы;
- идентификаторы;
- ключевые слова;
- комментарии.

Алфавит неизменен для всех задач, решаемых на языке С.

Сформулированная задача не требует использование констант.

Обязательным идентификатором для любого решения задачи является число task – исходно введенное значение команды меню.

Для решения возможно использовать следующие ключевые слова: int, for, else, void, sizeof, char, do, if, while, size_t, struct.

Комментарии являются необязательной частью программы и будут опущены при решени

Практическая часть

Исходя из условия задачи очевидно, что необходимо осуществить в коде меню, с перечнем выполняемых функций: добавление структуры в файл, вывод всех структур в консоль, сортировка структур в бинарном файле, удаление структуры из файла, закрытие программы. Для первых четырех функций было принято решение работать со структурой Music.

В соответствии со схемой, приведенной выше, выделены функции добавления структуры в файл - add, вывода всех структур в консоль - print, сортировка структур в бинарном файле - sort, удаление структуры из файла - del.

Накладываемым ограничением в программе является условие, что введенное число целое, неотрицательное, лежащее в промежутке от 1 до 5. Ограничена и длина названия музыкальной композиции, длина имени исполнителя. Длина обоих параметров задается макросом nameLen. Размер бинарного файла также ограничен и задан макросом fileLen.

Исходов завершения программы два:

1. Исполнение соответствующих действий, пока не будет дана команда прекращения работы программы;

2. Если в ходе выполнения программы произойдет ошибка открытия / закрытия файла, то программа прекращает свою работу, выведется сообщение об ошибке в консоль.

Входные и выходные данные :

Входные данные:

option - номер соответствующей опции в меню.

Выходные данные:

Вывод информации об исполненных операциях и сообщения об успешной компиляции программы.

Реализованные функции :

main	Функция аргументов не принимает. - Вызывает функцию, которая печатает информацию о командах; - Запрашивает номер команды; - Вызывает функцию, соответствующую введенному числу; - Повторяет алгоритм или завершает программу. Функция возвращает 0.
input	Аргументами функции являются указатель на массив с элементами типа char, размером, заданным переменной типа size_t (unsigned int); - Считывает с клавиатуры строку; - В случае, если количество символов не превышает размер массива, то строка записывается в этот массив, функция возвращает 1; - В случае, если пользователь ввел пустую строку или строку, размер которой превышает максимально допустимый размер массива, то функция возвращает 0; Функция возвращает целочисленное значение.

closeFile	Аргументом функции является переменная fp - указатель на файл. - Функция закрывает ранее открытый файл при помощи функции стандартной библиотеки fclose. В случае возникновения ошибки возвращается значение EOF, в консоль выводится соответствующее сообщение, функция возвращает 0. В случае успешного закрытия потока функция возвращает значение 1. Функция возвращает целочисленное значение.
print	Аргументом функции является указатель на массив символов типа char. - Функция открывает поток(файл) в режиме чтения бинарного файла с помощью функции стандартной библиотеки fopen. Производится проверка открытия файла: если возвращаемое значение NULL, то произошла ошибка, и функция возвращает значение 0; Если открытие произошло успешно, то функция продолжает свою работу. - С помощью цикла while и функции стандартной библиотеки fread осуществляется чтение бинарного файла и вывод в консоль, в понятной пользователю форме, данных из файла. Цикл работает до тех пор, пока файл не будет полностью считан (пока fread не возвратит значение 1). Функция возвращает целочисленное значение.

add	Аргументом функции является указатель на символьный массив типа char. - Функция открывает поток(файл) в режиме записи в конец бинарного файла с помощью функции стандартной библиотеки fopen. Производится проверка открытия файла: если возвращаемое значение NULL, то произошла ошибка, и функция возвращает значение 0; Если открытие произошло успешно, то функция продолжает свою работу. - Функция ведет себя однотипно при вводе названия и автора композиции. У пользователя запрашивается название/автор музыкальной композиции. - При помощи цикла while и функции input происходит проверка введенного массива символов. Если функция возвращает значение 1, то введенный массив символов соответствует требованиям и является верным, функция продолжает свою работу, цикл завершается. Если функция возвращает 0, то цикл продолжается, пользователю снова предлагается ввести данные. Цикл продолжается до тех пор, пока введенные данные не пройдут проверку. - С помощью функции стандартной библиотеки strcpy из буфера, в котором содержатся введенные пользователем данные, прошедшие проверку, копируются в соответствующие поля структуры переменной msc структуры Music.

	- .Функция ведет себя однотипно при вводе продолжительности и оценки музыкальной композиции. У пользователя запрашивается продолжительность / оценка композиции. - При помощи цикла while происходит проверка введенного массива символов. Если введенные числа неотрицательны, то цикл завершается; в противном случае цикл продолжает свою работу, пока вводимые символы не будут в точности соответствовать требованиям. Введенные числа напрямую сохраняются в соответствующем поле структуры. - По окончании работы работы функции введенные данные перезаписываются в бинарный файл из полей структуры, поток закрывается с помощью функции closeFile. Функция возвращает значение передаваемое функцией closeFile (1 или 0), свидетельствующее об успешном/провальном закрытии файла. Функция возвращает целочисленное значение

del	Аргументом функции является строка - имя файла. В случае возникновения ошибок функция возвращает 0; В случае корректной работы возвращает 1. - Открывает файл на чтение и дополнение; - Считывает с помощью input имя исполнителя, по которому происходит удаление. - Считывает структуры из файла, если попадается структура с совпадающим полем “исполнитель”, на ее место записывает следующую структуру, а на место следующую, ту которая идет перед ней. (Таким образом структуры в файле смещаются на одну позицию вверх, а конечная структура дублируется); - Уменьшает файл на размер одной структуры, таким образом удаляя последнюю; - Повторяет пункты 3 и 4, пока все структуры с соответствующим полем “Исполнитель” не удалятся. - По окончании поток закрывается с помощью функции closeFile. Функция возвращает значение передаваемое функцией closeFile (1 или 0), свидетельствующее об успешном/провальном закрытии файла. Функция возвращает целочисленное значение.
sort	Аргументом функции является указатель на массив символов типа char. - Функция открывает поток(файл) в режиме чтения бинарного файла с помощью функции стандартной библиотеки fopen. Производится проверка открытия файла: если возвращаемое значение NULL, то произошла ошибка, и функция возвращает значение 0; Если открытие произошло успешно, то функция продолжает свою работу. - Считывается структура из файла. В процессе считывания сравнивается поле “Имя” структуры с этим полем предыдущей (функция strcmp). Если поля расположены не в алфавитном порядке, то меняет структуры местами. - Повторяет считывание и замены, пока в процессе считывания структур не будет требоваться менять их местами.

	- По окончании поток закрывается с помощью функции closeFile. Функция возвращает значение передаваемое функцией closeFile (1 или 0), свидетельствующее об успешном/провальном закрытии файла. Функция возвращает целочисленное значение.
menu	Функция аргументов не принимает. - Функция выводит в консоль перечень опций, которые реализованы в коде программы и доступны пользователю при активации соответствующего ключа(ввода в консоль номера пункта из предложенного перечня опций). Функция значений не возвращает
swap	Аргументом функции является переменная fp, которая является указателем на файл, переменные msc_1 и msc_2 структуры Music. - Функция меняет местами две переменные структуры с помощью функции fwrite. Функция значений не возвращает.

Примеры работы программы :

Входные данные	Выходные данные
1	
Battle of Hymns	Extreme — He-Man Woman Hater — 324 sec. — #4
Manowar	Motley Crue — Misunderstood — 246 sec. — #5 System of a Down — Toxicity — 231 sec. — #5
342	Manowar — Battle of Hymns — 342 sec. — #4
4	
2	
3	
2	Manowar — Battle of Hymns — 342 sec. — #4 Extreme — He-Man Woman Hater — 324 sec. — #4
	Motley Crue — Misunderstood — 246 sec. — #5
4	System of a Down — Toxicity — 231 sec. — #5
Motley Crue	
2	Manowar — Battle of Hymns — 342 sec. — #4 Extreme — He-Man Woman Hater — 324 sec. — #4
	System of a Down — Toxicity — 231 sec. — #5
5	The program has successfully compiled

6	Unexpected option.

```
        MENU:
*-------------------------*
1. Add song to the playlist.
2. Show the playlist.
3. Sort playlist by name.
4. Delete music by artist.
5. Exit.
1
Enter a name of a song: Battle Hymns
Enter an artists nickname: Manowar
Enter duration: 342
Rate the song: 4

        MENU:
*-------------------------*
1. Add song to the playlist.
2. Show the playlist.
3. Sort playlist by name.
4. Delete music by artist.
5. Exit.
2
Extreme --- He-Man Woman Hater --- 324 sec. --- #4
Motley Crue --- Misunderstood --- 246 sec. --- #5
System of a Down --- Toxicity --- 231 sec. --- #5
Manowar --- Battle Hymns --- 342 sec. --- #4

        MENU:
*-------------------------*
1. Add song to the playlist.
2. Show the playlist.
3. Sort playlist by name.
4. Delete music by artist.
5. Exit.
3

        MENU:
*-------------------------*
1. Add song to the playlist.
2. Show the playlist.
3. Sort playlist by name.
4. Delete music by artist.
5. Exit.
2
Manowar --- Battle Hymns --- 342 sec. --- #4
Extreme --- He-Man Woman Hater --- 324 sec. --- #4
Motley Crue --- Misunderstood --- 246 sec. --- #5
System of a Down --- Toxicity --- 231 sec. --- #5
```

Рис. 1.1 Пример корректной работы программы

```
        MENU:
*-------------------------*
1. Add song to the playlist.
2. Show the playlist.
3. Sort playlist by name.
4. Delete music by artist.
5. Exit.
4
Enter an author of a song to delete: Motley Crue

        MENU:
*-------------------------*
1. Add song to the playlist.
2. Show the playlist.
3. Sort playlist by name.
4. Delete music by artist.
5. Exit.
2
Manowar --- Battle Hymns --- 342 sec. --- #4
Extreme --- He-Man Woman Hater --- 324 sec. --- #4
System of a Down --- Toxicity --- 231 sec. --- #5

        MENU:
*-------------------------*
1. Add song to the playlist.
2. Show the playlist.
3. Sort playlist by name.
4. Delete music by artist.
5. Exit.
5

The program has succesessfully compilled.
```

Рис. 1.2 Пример корректной работы программы

```
        MENU:
*---------------------------*
1. Add song to the playlist.
2. Show the playlist.
3. Sort playlist by name.
4. Delete music by artist.
5. Exit.
6
Unexpected option
```

Рис. 2.1 Пример работы программы в случае некорректного ввода опции

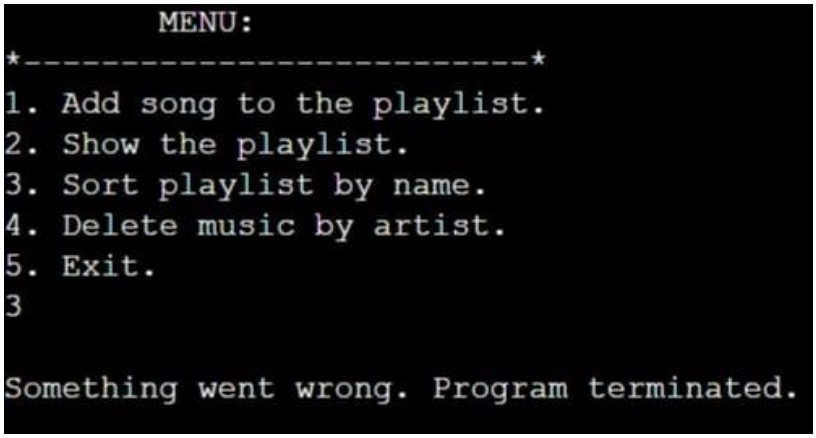

Рис. 2.2 Пример работы программы в случае отсутствия бинарного файла для записи.

Текст программы :

main.c - файл с основным кодом программы :

```
#include
<stdio.h>
#include
<string.h>
#include
<unistd.h>
#include
<stdlib.h>
#include
"music.h"

#define nameLen 255
#define fileLen 512

int input(char *buff, size_t
sz); int closeFile(FILE
**fp);
int print(char
*filename); int
add(char *filename);
int del(char
*filename); int
sort(char *filename);
void swap(FILE **fp, struct Music msc_1, struct Music
msc_2); void menu();

int main()
```

```
{
    FILE *fp;
    char *filename =
    "input.bin"; int option =
    1;
    int op;
    while (option){
        menu();
        scanf("%d",
        &op); getchar();
        if (op == 1)
            option =
        add(filename); else if (op
        == 4)
```

```
            option =
        del(filename); else if (op
        == 3)
            option =
        sort(filename); else if
        (op == 2)
            option =
        print(filename); else if
        (op == 5)
            option = 0;
        else
            printf("Unexpected option\n");
        printf("\n");
    }
    if (op == 5)
        printf("The program has succesessfully compilled.");
    else
        printf("Something went wrong. Program terminated.\
    n"); return 0;
}

int input(char *buff, size_t sz)
{
    int extra, ch;
    if (fgets (buff, sz, stdin) == NULL){
        printf("Wrong input.");
        return 0;
    }
    if (*(buff + (strlen(buff) - 1)) != '\
```

```
n'){ extra = 0;
while (((ch = getchar()) != '\n') && (ch !=
    EOF)) extra = 1;
if (extra){
    printf("Wrong input.");
}
return (extra == 1) ? 0 : 1;
```

```
    }
    *(buff + (strlen(buff) - 1)) =
    '\0'; return 1;
}

int closeFile(FILE **fp)
{
    if (fclose(*fp) == EOF) {
        printf("Can not close file.\
        n"); return 0;
    }
    return 1;
}

int print(char *filename)
{
    FILE *fp;
    struct Music msc;
    if((fp = fopen(filename, "rb")) ==
        NULL) return 0;
    while(fread(&msc, sizeof(msc), 1, fp) == 1) {
        printf("%s --- %s --- %d sec. --- #%d\n", msc.Artist, msc.Name,
msc.Duration, msc.Rank);
    }
    return closeFile(&fp);
}

int add(char *filename)
{
    FILE *fp;
```

```
struct Music msc;
char
buff[nameLen];
if((fp = fopen(filename, "ab")) ==
    NULL) return 0;
```

```
printf("Enter a name of a song:
"); while(!input(buff,
sizeof(buff))) {
    printf("Enter a name of a song: ");
}
strcpy(msc.Name, buff);
printf("Enter an artists nickname: ");
while(!input(buff, sizeof(buff))){
    printf("Enter an artists nickname: ");
}
strcpy(msc.Artist,
buff); printf("Enter
duration: ");
scanf("%d",
&msc.Duration);
while(msc.Duration <= 0)
{
    printf("Something went wrong. Try again.\n");
    scanf("%d", &msc.Duration);
}
printf("Rate the song:
"); scanf("%d",
&msc.Rank);
while(msc.Rank < 0) {
    printf("Something went wrong. Try again.\n");
    scanf("%d", &msc.Rank);
}
fwrite(&msc, sizeof(msc), 1,
fp); return closeFile(&fp);
```

```
}

int del(char *filename)
{
    char
    buff[nameLen];
    FILE *fp;
    int offset = 0, n = 0;
    printf("Enter an author of a song to delete: "); while(!
    input(buff, sizeof(buff))){
```

```
        printf("Enter an author of a song to delete: ");
    }
    struct Music msc;
    if((fp = fopen(filename, "rb+")) ==
        NULL) return 0;
    do {
        rewind(fp
        ); n = 0;
        offset = 0;
        while(fread(&msc, sizeof(msc), 1, fp) == 1)
            { n++;
            if (offset){
                fseek(fp, -sizeof(msc) * 2,
                1); fwrite(&msc, sizeof(msc),
                1, fp); fseek(fp,
                sizeof(msc), 1);
            }
            if (strcmp(buff, msc.Artist)==0)
                offset = 1;
        }
        if (offset)
            ftruncate(fileno(fp), sizeof(msc)*(n - 1));
    } while (offset);
    return
    closeFile(&fp);
}

int sort(char *filename)
{
```

```
FILE *fp;
int first_iter = 1, NoSwap =
0; struct Music msc,
msc_prev;
if((fp = fopen(filename, "rb+")) ==
    NULL) return 0;
while (!NoSwap) {
```

```
        rewind(fp);
        first_iter = 1;
        NoSwap = 1;
        while(fread(&msc, sizeof(msc), 1, fp) ==
            1) { if (!first_iter) {
                if (strcmp(msc_prev.Name, msc.Name) > 0)
                    { swap(&fp, msc_prev, msc);
                    NoSwap = 0;
                } else
                    msc_prev = msc;
            } else {
                first_iter =
                0; msc_prev =
                msc;
            }
        }
    }
    return closeFile(&fp);
}

void swap(FILE **fp, struct Music msc_1, struct Music msc_2)
{
    fseek(*fp, -sizeof(msc_1)*2, 1);
    fwrite(&msc_2, sizeof(msc_1), 1,
    *fp); fwrite(&msc_1,
    sizeof(msc_1), 1, *fp);
}

void menu()
```

```
{
    printf("       MENU:\n");
    printf("*……*\
    n"); printf("1. Add song to the
    playlist.\n"); printf("2. Show the
    playlist.\n"); printf("3. Sort
    playlist by name.\n");
```

```
    printf("4. Delete music by artist.\
    n"); printf("5. Exit.\n");
}
```

Файл под структуры - music.h :

```
struct Music
{
    char Name[255];
    char
    Artist[255];
    int  Duration;
    int Rank;
};
```