

Министр науки и высшего образования Российской Федерации

**Федеральное государственное автономное образовательное учреждение высшего
образования**

«Национальный исследовательский университет ИТМО»

Факультет информационных технологий и программирования

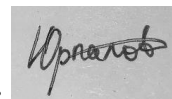
Лабораторная работа №13

Название работы: Редактор метаинформации tr3-файла.

Выполнил студент группы № М3105

Юрпалов Сергей Николаевич

Подпись:



Проверил:

Повышев Владислав Вячеславович

Санкт-Петербург
2020

Лабораторная работа 13. Редактор метаинформации mp3-файла

Задача.

Реализовать редактор текстовой метаинформации mp3 файла. В качестве стандарта метаинформации принимаем ID3v2.

Редактор представлять из себя консольную программу принимающую в качестве аргументов имя файла через параметра `--filepath`, а также одну из выбранных команд

1. `--show` - отображение всей метаинформации в виде таблицы
2. `--set=prop_name --value=prop_value` - выставляет значение определенного поля метаинформации с именем `prop_name` в значение `prop_value`
3. `--get=prop_name` - вывести определенное поле метаинформации с именем `prop_name`

Например:

```
app.exe --filepath=Song.mp3 --show  
app.exe --filepath=Song.mp3 --get=TIT2  
app.exe --filepath=Song.mp3 --set=COMM --value=Test
```

Примечание.

При выполнении данной работы разрешается использовать только стандартную библиотеку языка C. Исключением может являться процесс разбора аргументов командной строки.

Решение с комментариями:

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

void show(char *filename){
    char primarytag[10], tag[4], flag[2], size[4]; // создаём char массивы
    для хранения данных в соотв. с ID3v2
    int i = 0;
    FILE *file = fopen(filename, "r"); // открываем файл на чтение
    fread(&primarytag, 1, 10, file);
    while(1){
        fread(&tag, 1, 4, file); // в цикле читаем необходимые нам данные
        fread(&flag, 1, 2, file);
        fread(&size, 1, 4, file);

        if(tag[0] <= 'A' || tag[0] >= 'Z' ){ // если tag - не буква, то
останавливаем программу
            break;
        }
        else{
            for (int j = 0; j < 4; j++){
                printf("%c", tag[j]); // пишем сам tag
            }
            int n = (int)size[0] + (int)size[1] + (int)size[2] +
(int)size[3]; // определяем размер значения, которое соответствует tag
            char* value = (char*)malloc(n*sizeof(char));
            fread(value, 1, n, file); // создаём массив и записываем в него
это значение
            for (int j = 0; j < n; j++){
                printf("%c", value[j]);
            }
            printf("\n"); // выводим его на консоль
        }
        i++;
    }
    fclose(file); // закрываем файл
}

void get(char *filename, char *name){
    printf("%s", name);
    char primarytag[10], tag[4], flag[2], size[4]; // создаём char массивы
    для хранения данных в соотв. с ID3v2
    int i = 0;
    FILE *file = fopen(filename, "r"); // открываем файл на чтение
    fread(&primarytag, 1, 10, file);
    while(1){
        fread(&tag, 1, 4, file); // в цикле читаем необходимые нам данные
        fread(&flag, 1, 2, file);
        fread(&size, 1, 4, file);
        if(tag[0] <= 'A' || tag[0] >= 'Z'){ // если tag - не буква, то
останавливаем программу
            break;
        }
        else{
            int n = (int)size[0] + (int)size[1] + (int)size[2] +
(int)size[3]; // определяем размер значения, которое соответствует tag
            char* value = (char*)malloc(n*sizeof(char));
            fread(value, 1, n, file); // создаём массив и записываем в него
это значение
            int bool = 1;
            for(int j=0; j<4; j++){ // цикл проверки, необходимый ли tag мы
прочитали
                if (tag[j] != name[j]){
```

```

        bool = 0;
    }
}
if (bool) { // если прочитан необходимый tag, выводим его
значение
    for (int j = 0; j < n; j++) {
        printf("%c", value[j]);
    }
}
}
i++;
}
fclose(file);
}

void set(char *filename, char *name, char *changer, int lenofchanger) {
    FILE *file = fopen(filename, "rb+"); // открываем файл на чтение + запись
    size_t s; // переменная для вычисления размера входного файла
    fseek(file, 0, SEEK_SET); // алгоритм вычисления размера файла
    if (file != NULL) {
        fseek(file, 0, SEEK_END);
        s = ftell(file);
    }
    fseek(file, 0, SEEK_SET);
    char *data = (char *)malloc(sizeof(char)*s); // создаём массив для
хранения данных файла
    fread(data, 1, s, file); // читаем эти данные
    int pointer = 10;
    while (data[pointer] != 0) { // цикл чтения данных файла
        int j = pointer + 4; // указатель на элемент данных с шагом
sizeof(int)
        int size = data[j] * 16777216 + data[j+1] * 65536 + data[j+2] * 256 +
data[j+3];
        if ((data[pointer] == name[0]) && (data[pointer+1] == name[1]) &&
(data[pointer+2] == name[2]) && (data[pointer+3] == name[3])) {
            // если среди данных нашли необходимый tag, то
            pointer += 4;
            data[pointer] = (lenofchanger/16777216)%256;
            data[pointer+1] = (lenofchanger/65536)%256;
            data[pointer+2] = (lenofchanger/256)%256;
            data[pointer+3] = lenofchanger%256;
            fseek(file, 0, SEEK_SET);
            fwrite(data, 1, pointer+6, file); // записываем все данные файла
обратно, кроме значения необходимого tag
            fwrite(changer, 1, lenofchanger, file); // записываем изменённое
значение tag
            fwrite(&data[pointer+6+size], 1, s-size-pointer-6, file);
        }
        else {
            pointer += 10+size; // продолжаем выполнение цикла
        }
    }
    fclose(file);
    free(data);
}

int main(int argc, char *argv[]) {
    int k=1;
    while (k+1 < argc) { // читаем аргументы командной строки
        char *filename = strpbrk(argv[k], "=") + 1; // Парсим входные данные,
ищем имя файла
        if (argv[k+1][2] == 's') {
            if (argv[k+1][3] == 'h') { // запуск show

```

```

        show(filename);
    }
    else if (argv[k+1][3] == 'e'){
        char *task=strpbrk(argv[k+1], "=") + 1;
        char *value=strpbrk(argv[k+2], "=") + 1;
        for(int i = strlen(value); i > 0 ; i--) // цикл, который
исправляет ошибку отсутствия пробела
            value[i] = value[i-1];
        value[0] = ' ';
        set(filename, task, value, strlen(value)); // запуск set с
доп. параметрами названия tag, заменой и её длиной
    }
}
else{
    char *task=strpbrk(argv[k+1], "=") + 1;
    get(filename, task); // запуск get с доп. параметром название tag
}
k += 2;
}
}

```