**Министр науки и высшего образования Российской̆ Федерации**

**Федеральное государственное автономное образовательное учреждение высшего образования**

**«Национальный исследовательский университет ИТМО»**

Факультет информационных технологий и программирования

Лабораторная работа №13

**Выполнил студент группы № M3104**

Гурман Тимофей Владимирович

**Подпись:**

****

**Проверил:**

Повышев Владислав Вячеславович

Санкт-Петербург

2020

Текст задания

## 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 выполнения данной работы разрешается использовать только стандартную библиотеку языка С. Исключением может являться процесс разбора аргументов командной строки.

Решение с комментариями

#include <stdio.h>  
#include <stdlib.h>  
#include <string.h>  
  
unsigned int reverseBytes(unsigned int N) {  
 return (N >> 24) | ((N >> 8) & 0x0000ff00) | ((N << 8) & 0x00ff0000) | ((N << 24) & 0xff000000);  
}  
  
union TagHeader {  
 char buffer[12];  
  
 struct {  
 unsigned short empty;  
 unsigned char version[3];  
 unsigned char v1;  
 unsigned char v2;  
 unsigned char flags;  
 unsigned int size;  
 } data;  
  
} Tagheader;  
  
union FrameHeader {  
 char buffer[10];  
  
 struct {  
 unsigned char name[4];  
 unsigned int size;  
 unsigned short flags;  
  
 } data;  
  
} Frameheader;  
  
  
void show(char \*filename) {  
 FILE\* file;  
 file = fopen(filename, "rb");  
 if (file == NULL)  
 {  
 printf("The file doesn't exist\n");  
 exit(1);  
 }  
 fseek(file, 0, 0);  
 fread(Tagheader.buffer + 2, 1, 10, file);  
 unsigned int tag\_size = reverseBytes(Tagheader.data.size);  
 while (ftell(file) < tag\_size + 10) {  
 fread(Frameheader.buffer, 1, 10, file);  
 if (Frameheader.data.name == 0)  
 break;  
 printf("%s: ", Frameheader.data.name);  
 unsigned int frame\_size = reverseBytes(Frameheader.data.size);  
 unsigned char\* frameContent;  
 frameContent = malloc(frame\_size);  
 fread(frameContent, 1, frame\_size, file);  
 for (int i =0; i < frame\_size; ++i)  
 {  
 printf("%c", frameContent[i]);  
 }  
 printf("\n");  
 }  
 fclose(file);  
}  
  
  
void get(char\* fileName, char Framename[4])  
{  
 FILE\* file;  
 file = fopen(fileName, "rb");  
  
 if (file == NULL)  
 {  
 printf("The file doesn't exist\n");  
 exit(1);  
 }  
  
 fread(Tagheader.buffer + 2, 1, 10, file);  
 unsigned int tag\_ize = reverseBytes(Tagheader.data.size);  
  
 while (ftell(file) < tag\_ize + 10) {  
 fread(Frameheader.buffer, 1, 10, file);  
 unsigned int frameSize = reverseBytes(Frameheader.data.size);  
 if (strcmp(Frameheader.data.name, Framename) == 0) {  
 printf("%s: ", Frameheader.data.name);  
 unsigned char\* FrameContent;  
 FrameContent = malloc(frameSize);  
 fread(FrameContent, 1, frameSize, file);  
 unsigned int i;  
 for (i = 0; i < frameSize; ++i)  
 {  
 printf("%c", FrameContent[i]);  
 }  
 printf("\n");  
 free(FrameContent);  
 fclose(file);  
 return;  
 }  
 fseek(file, frameSize, 1);  
 }  
 printf("No matches");  
}  
  
void Copy(FILE \*input, FILE \*output) {  
 int temp;  
 while ((temp = gets(input)) != EOF) {  
 putc(temp, output);  
 }  
}  
  
void set(char\* fileName, char frameName[4], char\* frameValue)  
{  
 FILE\* file;  
 file = fopen(fileName, "rb");  
  
 if (file == NULL)  
 {  
 printf("The file doesn't exist\n");  
 exit(1);  
 }  
  
 fread(Tagheader.buffer + 2, 1, 10, file);  
  
 unsigned int tagSize = reverseBytes(Tagheader.data.size);  
 unsigned int FramePos = 0;  
 unsigned int FrameSize = 0;  
  
 while (ftell(file) < tagSize + 10)  
 {  
 fread(Frameheader.buffer, 1, 10, file);  
  
 unsigned int frameSize = reverseBytes(Frameheader.data.size);  
  
 if (strcmp(Frameheader.data.name, frameName) == 0)  
 {  
 FramePos = ftell(file) - 10;  
 FrameSize = frameSize;  
 break;  
 }  
 if (Frameheader.data.name[0] == 0){  
 break;  
 }  
 fseek(file, frameSize, 1);  
 }  
  
 unsigned int valueSize = strlen(frameValue);  
  
 unsigned int newTagSize = tagSize - FrameSize + valueSize + 10 \* (FramePos == 0);  
  
 if (FramePos == 0)  
 {  
 FramePos = ftell(file) - 10;  
 }  
  
 if (valueSize == 0)  
 {  
 newTagSize -= 10;  
 }  
  
 FILE\* fileCopy;  
 fileCopy = fopen("tmp.mp3", "wb");  
  
 fseek(file, 0, SEEK\_SET);  
 fseek(fileCopy, 0, SEEK\_SET);  
 Copy(file, fileCopy);  
  
 fclose(file);  
 fclose(fileCopy);  
  
 fileCopy = fopen("tmp.mp3", "rb");  
 file = fopen(fileName, "wb");  
  
 Tagheader.data.size = reverseBytes(newTagSize);  
  
 fwrite(Tagheader.buffer + 2, 1, 10, file);  
  
 fseek(fileCopy, 10, 1);  
 for (int i = 0; i < FramePos - 10; ++i)  
 {  
 int c;  
 c = getc(fileCopy);  
 putc(c, file);  
 }  
  
 if (valueSize > 0)  
 {  
 for (int i = 0; i < 4; ++i)  
 {  
 Frameheader.data.name[i] = frameName[i];  
 }  
 Frameheader.data.size = reverseBytes(valueSize);  
 Frameheader.data.flags = 0;  
 fwrite(Frameheader.buffer, 1, 10, file);  
 }  
  
 fwrite(frameValue, 1, valueSize, file);  
  
 fseek(fileCopy, FramePos + 10 + FrameSize, SEEK\_SET);  
 for (int i = ftell(file); i < newTagSize; ++i) {  
 unsigned int c;  
 c = getc(fileCopy);  
 putc(c, file);  
 }  
  
 printf("New value for frame %s: %s\n", frameName, frameValue);  
  
 Copy(fileCopy, file);  
  
 fclose(file);  
 fclose(fileCopy);  
}  
  
  
int main(int argc, char\* argv[]) {  
 char\* fileName;  
 char\* frameName;  
 char\* value;  
 char showFlag = 0;  
 char setFlag = 0;  
 char getFlag = 0;  
 int i;  
 for (i = 1; i < argc; i++) {  
 if (strcmp(argv[i], "--show") == 0) {  
 showFlag = 1;  
 continue;  
 }  
 if (argv[i][2] == 'f')  
 {  
 fileName = strpbrk(argv[i], "=") + 1;  
 }  
 if (argv[i][2] == 'g') {  
 getFlag = 1;  
 frameName = strpbrk(argv[i], "=") + 1;  
 continue;  
 }  
 if (argv[i][2] == 's') {  
 setFlag = 1;  
 frameName = strpbrk(argv[i], "=") + 1;  
 continue;  
 }  
 if (argv[i][2] == 'v') {  
 value = strpbrk(argv[i], "=") + 1;  
 continue;  
 }  
 }  
  
 if (showFlag)  
 {  
 show(fileName);  
 }  
 if (getFlag)  
 {  
 get(fileName, frameName);  
 }  
 if (setFlag)  
 {  
 set(fileName, frameName, value);  
 }  
  
 return 0;  
}