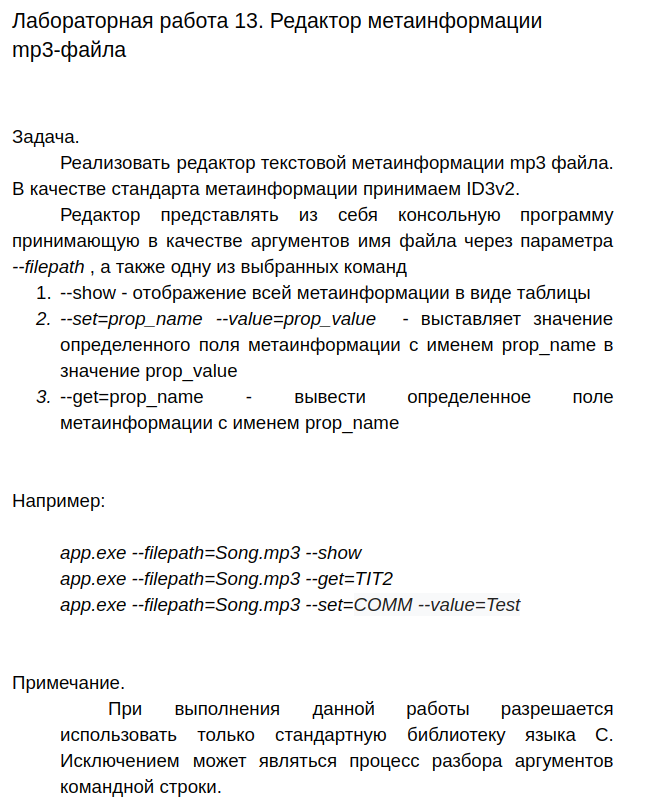
**Лабараторная работа 13**

Выполнил студент группы M3105

Клишевич Вадим Александрович

main.c

#include **"Settings.h"**  
#include **"mp3.h"**  
  
  
**int** main(**int** argc, **char** \*\*argv)  
{  
 Settings \*settings = setSettings(argc, argv);  
 **if** (settings->show) {  
 show(settings);  
 } **else if** (settings->get) {  
 get(settings);  
 } **else if** (settings->set) {  
 set(settings);  
 }  
  
 **return** 0;  
}

settings.h

*//*  
*// Created by vadim on 12.12.19.*  
*//*  
  
#ifndef **LABA15\_SETTINGS\_H**  
#define **LABA15\_SETTINGS\_H**  
  
#pragma pack(push, 1)  
**typedef struct** tagSettings {  
 **char** \*fileName;  
 **char** show;  
 **char** get;  
 **char** set;  
 **char** \*metaName;  
 **char** \*value;  
} Settings;  
#pragma pack(pop)  
  
Settings \*setSettings(**int** argc, **char** \*\*argv);  
  
#endif *//LABA15\_SETTINGS\_H*

settings.c

*//*  
*// Created by vadim on 12.12.19.*  
*//*  
  
#include **"Settings.h"**  
#include **<stdio.h>**  
#include **<stdlib.h>**  
#include **"string.h"**  
  
Settings \*setSettings(**int** argc, **char** \*\*argv) {  
 Settings \*settings = (Settings \*)malloc(**sizeof**(Settings));  
  
 **for** (**int** i = 1; i < argc; i++) {  
 **if** (!strcmp(**"--filepath"**, argv[i])) {  
 settings->fileName = argv[++i];  
 } **else if** (!strcmp(**"--show"**, argv[i])) {  
 settings->show = 1;  
 } **else if** (!strcmp(**"--get"**, argv[i])) {  
 settings->get = 1;  
 settings->metaName = argv[++i];  
 } **else if** (!strcmp(**"--set"**, argv[i])) {  
 settings->set = 1;  
 settings->metaName = argv[++i];  
 } **else if** (!strcmp(**"--value"**, argv[i])) {  
 settings->value = argv[++i];  
 }  
 }  
  
 **return** settings;  
}

mp3.h

*//*  
*// Created by vadim on 12.12.19.*  
*//*  
  
#ifndef **LABA13\_MP3\_H**  
#define **LABA13\_MP3\_H**  
  
#import **"Settings.h"**  
#import **<stdlib.h>**  
#import **<stdio.h>**  
  
**typedef union** tagTAGHEADER  
{  
 **char** buffer[12]; *// 12 total*  
  
**struct** {  
 **unsigned short** empty; *// 2*  
**unsigned char** version[3]; *// 3*  
**unsigned char** v1; *// 1*  
**unsigned char** v2; *// 1*  
**unsigned char** flags; *// 1*  
**unsigned int** size; *// 4*  
} data;  
} TAGHEADER;  
  
**typedef union** tagFRAMEHEADER  
{  
 **char** buffer[10]; *// 10 total*  
  
**struct** {  
 **unsigned char** name[4]; *// 4*  
**unsigned int** size; *// 4*  
**unsigned short** flags; *// 2*  
} data;  
} FRAMEHEADER;  
  
**typedef struct** tagMP3 {  
 FILE \*file;  
 **unsigned int** tagSize;  
 TAGHEADER \*tagHeader;  
 **char** frameName[4];  
} MP3;  
  
**void** show(Settings \*settings);  
**void** get(Settings \*settings);  
**void** set(Settings \*settings);  
  
#endif *//LABA13\_MP3\_H*

mp3.c

*//*  
*// Created by vadim on 12.12.19.*  
*//*  
  
#include **"mp3.h"**  
#include **"string.h"**  
  
**unsigned int** reverseBytes (**unsigned int** n)  
{  
 **return** ((n >> 24) & 0x000000ff) |  
 ((n >> 8) & 0x0000ff00) |  
 ((n << 8) & 0x00ff0000) |  
 ((n << 24) & 0xff000000);  
}  
  
**void** copyFile(FILE\* inp, FILE\* outp) {  
 **int** c;  
 **while** ((c = getc(inp)) != **EOF**)  
 putc(c, outp);  
}  
  
MP3 \*open(Settings \*settings) {  
 MP3 \*mp3 = (MP3 \*)malloc(**sizeof**(MP3));  
 mp3->tagHeader = (TAGHEADER \*)malloc(**sizeof**(TAGHEADER));  
 mp3->file = fopen(settings->fileName, **"r"**);  
 **if** (mp3->file == **NULL**) {  
 printf(**"File not found :/"**);  
 exit(404);  
 }  
 fseek(mp3->file, 0, **SEEK\_SET**);  
 fread(mp3->tagHeader->buffer + 2, **sizeof**(**char**), 10, mp3->file);  
  
 mp3->tagSize = reverseBytes(mp3->tagHeader->data.size);  
  
 **return** mp3;  
}  
  
**void** show(Settings \*settings) {  
 MP3 \*mp3 = open(settings);  
  
 printf(**"%sv%d.%d\n"**, mp3->tagHeader->data.version, mp3->tagHeader->data.v1, mp3->tagHeader->data.v2);  
  
 **while** (ftell(mp3->file) < mp3->tagSize + 10)  
 {  
 FRAMEHEADER frameHeader;  
 fread(frameHeader.buffer, **sizeof**(**char**), 10, mp3->file);  
 **if** (frameHeader.data.name[0] == 0)  
 **continue**;  
 printf(**"%s: "**, frameHeader.data.name);  
  
 **unsigned int** frameSize = reverseBytes(frameHeader.data.size);  
  
 **unsigned char**\* frameContent;  
 frameContent = malloc(**sizeof**(**char**) \* frameSize);  
 fread(frameContent, **sizeof**(**char**), frameSize, mp3->file);  
  
 **for** (**unsigned int** i = 0; i < frameSize && frameSize < 400; ++i)  
 {  
 printf(**"%c"**, frameContent[i]);  
 }  
 printf(**"\n"**);  
 free(frameContent);  
 }  
 fclose(mp3->file);  
}  
  
**void** get(Settings \*settings) {  
 MP3 \*mp3 = open(settings);  
 **while** (ftell(mp3->file) < mp3->tagSize + 10)  
 {  
 FRAMEHEADER frameHeader;  
 fread(frameHeader.buffer, **sizeof**(**char**), 10, mp3->file);  
  
 **unsigned int** frameSize = reverseBytes(frameHeader.data.size);  
  
 **if** (strcmp(frameHeader.data.name, settings->metaName) == 0)  
 {  
 printf(**"%s: "**, frameHeader.data.name);  
  
 **unsigned char**\* frameContent;  
 frameContent = malloc(**sizeof**(**char**) \* frameSize);  
 fread(frameContent, **sizeof**(**char**), frameSize, mp3->file);  
  
 **for** (**unsigned int** i = 0; i < frameSize; ++i)  
 {  
 printf(**"%c"**, frameContent[i]);  
 }  
 printf(**"\n"**);  
 free(frameContent);  
 fclose(mp3->file);  
 **return**;  
 }  
  
 fseek(mp3->file, frameSize, **SEEK\_CUR**);  
 }  
 fclose(mp3->file);  
 printf(**"No value found for %s!"**, settings->metaName);  
}  
  
**void** set(Settings \*settings)  
{  
 MP3 \*mp3 = open(settings);  
  
 **unsigned int** oldFramePos = 0;  
 **unsigned int** oldFrameSize = 0;  
  
 **while** (ftell(mp3->file) < mp3->tagSize + 10)  
 {  
 FRAMEHEADER frameHeader;  
 fread(frameHeader.buffer, **sizeof**(**char**), 10, mp3->file);  
  
 **unsigned int** frameSize = reverseBytes(frameHeader.data.size);  
  
 **if** (strcmp(frameHeader.data.name, settings->metaName) == 0)  
 {  
 oldFramePos = ftell(mp3->file) - 10;  
 oldFrameSize = frameSize;  
 **break**;  
 }  
 fseek(mp3->file, frameSize, **SEEK\_CUR**);  
 }  
  
 **unsigned int** valueSize = strlen(settings->value);  
  
 **unsigned int** newTagSize = mp3->tagSize - oldFrameSize + valueSize + 10 \* (oldFramePos != 0);  
  
 **if** (oldFramePos == 0)  
 {  
 oldFramePos = ftell(mp3->file);  
 }  
 **if** (valueSize == 0)  
 {  
 newTagSize -= 10;  
 }  
  
 FILE\* fileCopy;  
 fileCopy = fopen(**"temp.mp3"**, **"wb"**);  
  
 fseek(mp3->file, 0, **SEEK\_SET**);  
 fseek(fileCopy, 0, **SEEK\_SET**);  
 copyFile(mp3->file, fileCopy);  
  
 fclose(mp3->file);  
 fclose(fileCopy);  
  
 fileCopy = fopen(**"temp.mp3"**, **"rb"**);  
 mp3->file = fopen(settings->fileName, **"wb"**);  
  
 mp3->tagHeader->data.size = reverseBytes(newTagSize);  
  
 fwrite(mp3->tagHeader->buffer + 2, **sizeof**(**char**), 10, mp3->file);  
  
 fseek(fileCopy, 10, **SEEK\_SET**);  
  
 **for** (**unsigned int** i = 0; i < oldFramePos - 10; ++i)  
 {  
 **int** c;  
 c = getc(fileCopy);  
 putc(c, mp3->file);  
 }  
  
 **if** (valueSize > 0)  
 {  
 FRAMEHEADER frameHeader;  
 **for** (**unsigned int** i = 0; i < 4; ++i)  
 {  
 frameHeader.data.name[i] = settings->metaName[i];  
 }  
 frameHeader.data.size = reverseBytes(valueSize);  
 frameHeader.data.flags = 0;  
 fwrite(frameHeader.buffer, **sizeof**(**char**), 10, mp3->file);  
 }  
  
 fwrite(settings->value, **sizeof**(**char**), valueSize, mp3->file);  
  
 fseek(fileCopy, oldFramePos + 10 + oldFrameSize, **SEEK\_SET**);  
  
 **for** (**unsigned int** i = ftell(mp3->file); i < newTagSize; ++i)  
 {  
 **unsigned short int** c;  
 c = getc(fileCopy);  
 putc(c, mp3->file);  
 }  
  
 printf(**"New value for frame %s: %s\n"**, settings->metaName, settings->value);  
  
 copyFile(fileCopy, mp3->file);  
  
 fclose(mp3->file);  
 fclose(fileCopy);  
 remove(**"temp.mp3"**);  
}