|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  |  |  | | МИНОБРНАУКИ РОССИИ | | | | Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | | | |
| Институт информационных технологий (ИТ) | |
| Кафедра инструментального и прикладного программного обеспечения (ИиППО) | |

|  |  |
| --- | --- |
| **ОТЧЁТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №2** | |
| **по дисциплине** | |
| «Системное программное обеспечение»  по теме: «Работа с файлами» | |
|  | |
| Выполнил студент группы ИКБО-16-18 | Павлов Д.В. |
| Принял преподаватель | Волков М.Ю. |

|  |  |  |
| --- | --- | --- |
| Работа выполнена | «\_\_»\_\_\_\_\_\_\_2021 г. | *(подпись студента)* |
|  |  |  |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_2021 г. | *(подпись руководителя)* |

Москва 2021

**Цель работы**

Изучить основы работы с двоичным и текстовым файлами на базе WIN32 API.

**Задание**

Разработать WIN32 Application с диалогами, которое должно обеспечить:

1. Запись в оба типа файлов данных следующих типов (значение данных задаются через диалоговое окно):
   1. BYTE,UINT;
   2. INT32/ INT64;
   3. WORD, DWORD;
   4. LONG, BOOL;
   5. Float; Double;
   6. Char;
   7. Строки (в файле должно быть три и более строки).
2. Чтение данных из двоичного файла и их отображение в диалоговом окне.
3. Чтение данных из текстового файла и их отображение в диалоговом окне.
4. Отображение дампов значений данных файлов.
5. Для сохранения файла и его загрузки использовать стандартные диалоги посредством вызова функций GetOpenFileName() и GetSaveFileName().

**Ход работы**

Итоговая программа представлена в листинге 1. Результат работы представлен на рисунках 1-5.

Листинг 1

#include <Windows.h>

#include <tchar.h>

#include <stdlib.h>

#include <string.h>

#include <iostream>

HINSTANCE hInst;

LRESULT CALLBACK WndProc(HWND, UINT, WPARAM, LPARAM);

HWND hwndButton1;

HWND hwndButton2;

HWND hwndButton3;

HWND hwndButton4;

HWND hwndButton5;

HWND hwndText, hwndTextInt;

HANDLE hFileFirstTxt, hFileFirstBin, hOutTxt, hInTxt, hOutBin, hInBin, hOutBinDump;

using namespace std;

int CALLBACK WinMain(

\_In\_ HINSTANCE hInstance,

\_In\_opt\_ HINSTANCE hPrevInstane,

\_In\_ LPSTR lpCmdLine,

\_In\_ int nCmdShow)

{

//window 1

WNDCLASSEX wcex;

HBRUSH greenBrush = CreateSolidBrush(RGB(204, 255, 204));

ZeroMemory(&wcex, sizeof(WNDCLASSEX));

wcex.cbSize = sizeof(WNDCLASSEX);

wcex.style = CS\_HREDRAW | CS\_VREDRAW;

wcex.lpfnWndProc = WndProc;

wcex.cbClsExtra = 0;

wcex.cbWndExtra = 0;

wcex.hInstance = hInstance;

wcex.hIcon = LoadIcon(hInstance, IDI\_APPLICATION);

wcex.hCursor = LoadCursor(NULL, IDC\_ARROW);

wcex.hbrBackground = greenBrush;

wcex.lpszMenuName = NULL;

wcex.lpszClassName = L"window class1";

wcex.hIconSm = LoadIcon(wcex.hInstance, IDI\_APPLICATION);

if (!RegisterClassEx(&wcex)) {

MessageBox(NULL,

\_T("Window class creation failed for window 1"),

\_T("Window Class Failed"),

MB\_ICONERROR);

return 1;

}

HWND hWnd = CreateWindow(

wcex.lpszClassName,

L"Parent Window",

WS\_OVERLAPPEDWINDOW,

CW\_USEDEFAULT, CW\_USEDEFAULT,

500, 500,

NULL,

NULL,

hInstance,

NULL

);

if (!hWnd)

{

MessageBox(NULL,

\_T("Call to CreateWindow1 failed!"),

\_T("Window Creation Filed"),

MB\_ICONERROR);

return 1;

}

ShowWindow(hWnd,

nCmdShow);

hwndButton1 = CreateWindow(L"BUTTON", // Название класса

L"Записать в текстовый", // Текст на кнопке

WS\_VISIBLE | WS\_CHILD | BS\_DEFPUSHBUTTON, // Стиль кнопки

30, // x координаты

300, // y координаты

170, // ширина кнопки

30, // высота кнопки

hWnd, // указатель родительского окна

(HMENU)1001,

(HINSTANCE)GetWindowLong(hWnd, GWL\_HINSTANCE),

NULL);

hwndButton2 = CreateWindow(L"BUTTON", // Название класса

L"Читать с текстового", // Текст на кнопке

WS\_VISIBLE | WS\_CHILD | BS\_DEFPUSHBUTTON, // Стиль кнопки

280, // x координаты

300, // y координаты

170, // ширина кнопки

30, // высота кнопки

hWnd, // указатель родительского окна

(HMENU)1002,

(HINSTANCE)GetWindowLong(hWnd, GWL\_HINSTANCE),

NULL);

hwndButton3 = CreateWindow(L"BUTTON", // Название класса

L"Записать в бинарный", // Текст на кнопке

WS\_VISIBLE | WS\_CHILD | BS\_DEFPUSHBUTTON, // Стиль кнопки

30, // x координаты

350, // y координаты

170, // ширина кнопки

30, // высота кнопки

hWnd, // указатель родительского окна

(HMENU)1003,

(HINSTANCE)GetWindowLong(hWnd, GWL\_HINSTANCE),

NULL);

hwndButton4 = CreateWindow(L"BUTTON", // Название класса

L"Читать с бинарного", // Текст на кнопке

WS\_VISIBLE | WS\_CHILD | BS\_DEFPUSHBUTTON, // Стиль кнопки

280, // x координаты

350, // y координаты

170, // ширина кнопки

30, // высота кнопки

hWnd, // указатель родительского окна

(HMENU)1004,

(HINSTANCE)GetWindowLong(hWnd, GWL\_HINSTANCE),

NULL);

hwndText = CreateWindowEx(WS\_EX\_CLIENTEDGE, TEXT("Edit"), NULL,

WS\_CHILD | WS\_VISIBLE, 100, 20, 140,

20, hWnd, NULL, NULL, NULL);

hwndTextInt = CreateWindowEx(WS\_EX\_CLIENTEDGE, TEXT("Edit"), NULL,

WS\_CHILD | WS\_VISIBLE, 100, 50, 140,

20, hWnd, NULL, NULL, NULL);

UpdateWindow(hWnd);

hFileFirstTxt = CreateFile(\_T("C:\\Users\\deftr\\OneDrive\\Documents\\spo\\readme.txt"), GENERIC\_WRITE, FILE\_SHARE\_WRITE, NULL,

CREATE\_ALWAYS, FILE\_ATTRIBUTE\_NORMAL, NULL);

CloseHandle(HANDLE(hFileFirstTxt));

hFileFirstBin = CreateFile(\_T("C:\\Users\\deftr\\OneDrive\\Documents\\spo\\readme.bin"), GENERIC\_WRITE, FILE\_SHARE\_WRITE, NULL,

CREATE\_ALWAYS, FILE\_ATTRIBUTE\_NORMAL, NULL);

CloseHandle(HANDLE(hFileFirstBin));

MSG msg;

while (GetMessage(&msg, NULL, 0, 0))

{

TranslateMessage(&msg);

DispatchMessage(&msg);

}

return (int)msg.wParam;

};

LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)

{

PAINTSTRUCT ps;

HDC hdc;

switch (message)

{

case WM\_PAINT:

hdc = BeginPaint(hWnd, &ps);

TextOut(hdc,

5, 5,

NULL, NULL);

EndPaint(hWnd, &ps);

break;

case WM\_DESTROY:

PostQuitMessage(0);

break;

case WM\_COMMAND:

{

if (wParam == 1001)

{

hInTxt = CreateFile(\_T("C:\\Users\\deftr\\OneDrive\\Documents\\spo\\readme.txt"), GENERIC\_WRITE, FILE\_SHARE\_WRITE, NULL,

OPEN\_EXISTING, FILE\_ATTRIBUTE\_NORMAL, NULL);

if (hInTxt == INVALID\_HANDLE\_VALUE) {

//для вывода ошибки

TCHAR s[100];

\_stprintf\_s(s, \_T("%X"), GetLastError());

MessageBox(0, s, L"Ошибка", MB\_OK);

return 2;

}

TCHAR str[100];

wchar\_t\* wEndString;

char\* cEndString;

char s[100];

int iBuff;

//SendMessage(hwndText, WM\_GETTEXT, 100, LPARAM(str)); //считали

GetWindowText(hwndText, str, sizeof(str));

GetWindowTextA(hwndTextInt, s, sizeof(iBuff));

iBuff = strtol(s, &cEndString, 0);

wsprintf(str, \_T("%s %d\n"), str, iBuff);

WriteFile(hInTxt, str, sizeof(str), NULL, NULL);

CloseHandle(HANDLE(hInTxt));

break;

}

if (wParam == 1002)

{

hOutTxt = CreateFile(\_T("C:\\Users\\deftr\\OneDrive\\Documents\\spo\\readme.txt"), GENERIC\_READ, FILE\_SHARE\_READ, NULL,

OPEN\_EXISTING, FILE\_ATTRIBUTE\_NORMAL, NULL);

if (hOutTxt == INVALID\_HANDLE\_VALUE) {

//для вывода ошибки

TCHAR s[100];

\_stprintf\_s(s, \_T("%X"), GetLastError());

MessageBox(0, s, L"Ошибка", MB\_OK);

return 2;

}

DWORD dwTemp;

TCHAR str[100];

ReadFile(hOutTxt, str, sizeof(str), &dwTemp, NULL);

MessageBox(hWnd, str, L"TXT", MB\_OK);

CloseHandle(HANDLE(hOutTxt));

break;

}

if (wParam == 1003)

{

hInBin = CreateFile(\_T("C:\\Users\\deftr\\OneDrive\\Documents\\spo\\readme.bin"), GENERIC\_WRITE, FILE\_SHARE\_WRITE, NULL,

OPEN\_EXISTING, FILE\_ATTRIBUTE\_NORMAL, NULL);

if (hInBin == INVALID\_HANDLE\_VALUE) {

//для вывода ошибки

TCHAR s[100];

\_stprintf\_s(s, \_T("%X"), GetLastError());

MessageBox(0, s, L"Ошибка", MB\_OK);

return 2;

}

TCHAR text[100];

//SendMessage(hwndText, WM\_GETTEXT, 100, LPARAM(text)); //считали

wchar\_t\* wEndString;

char\* cEndString;

char s[100];

int iBuff, sizeOfStr;

GetWindowText(hwndText, text, sizeof(text));

GetWindowTextA(hwndTextInt, s, sizeof(iBuff));

iBuff = strtol(s, &cEndString, 0);

sizeOfStr = (GetWindowTextLength(hwndText) + 1) \* sizeof(TCHAR);

WriteFile(hInBin, &sizeOfStr, sizeof(int), NULL, 0);

WriteFile(hInBin, text, sizeOfStr, NULL, 0);

WriteFile(hInBin, &iBuff, sizeof(int), NULL, 0);

CloseHandle(HANDLE(hInBin));

break;

}

if (wParam == 1004)

{

hOutBin = CreateFile(\_T("C:\\Users\\deftr\\OneDrive\\Documents\\spo\\readme.bin"), GENERIC\_READ, FILE\_SHARE\_READ, NULL,

OPEN\_EXISTING, FILE\_ATTRIBUTE\_NORMAL, NULL);

if (hOutBin == INVALID\_HANDLE\_VALUE) {

//для вывода ошибки

TCHAR s[100];

\_stprintf\_s(s, \_T("%X"), GetLastError());

MessageBox(0, s, L"Ошибка", MB\_OK);

return 2;

}

DWORD dwTemp;

TCHAR str[100];

int sizeOfStr, number;

ReadFile(hOutBin, &sizeOfStr, sizeof(int), &dwTemp, NULL);

ReadFile(hOutBin, str, sizeOfStr, &dwTemp, NULL);

ReadFile(hOutBin, &number, sizeof(number), &dwTemp, NULL);

wsprintf(str, \_T("%s %d\n") , str, number);

MessageBox(hWnd, str, L"BIN", MB\_OK);

CloseHandle(HANDLE(hOutBin));

break;

}

break;

}

default:

return DefWindowProc(hWnd, message, wParam, lParam);

break;

}

return 0;

};

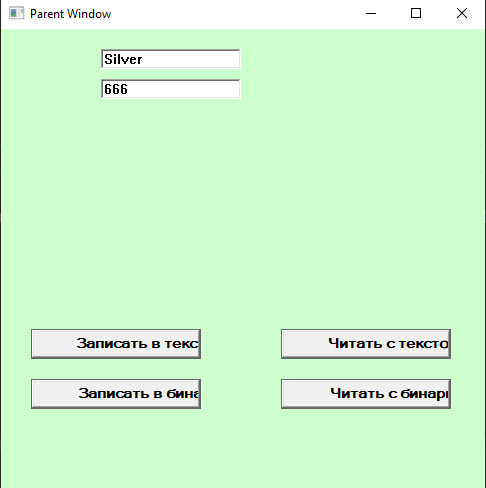
****

Рис. 1 – Главное меню.

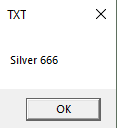


Рис.2 – Чтение из текстового файла.

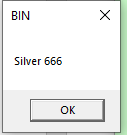


Рис.3 – Чтение из бинарного файла.

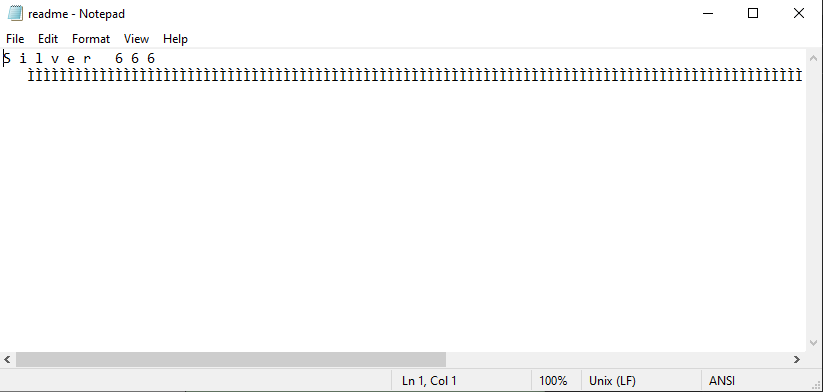


Рис.4 – Текстовый файл.

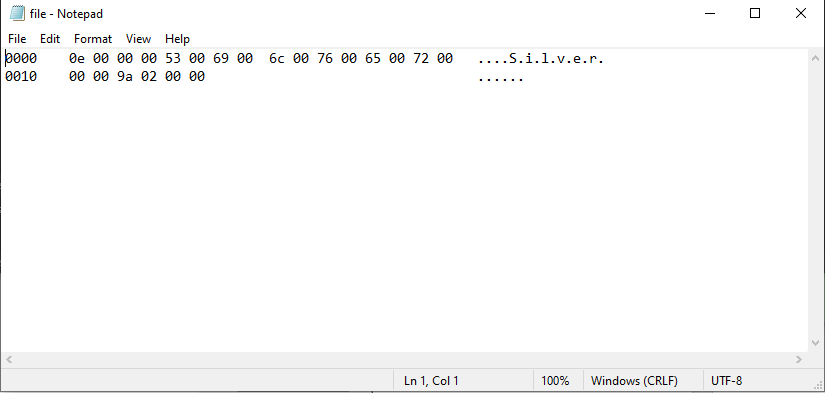


Рис.5 – Бинарный файл.

**Вывод**

В ходе практической работы была изучена работа с текстовыми и бинарными файлами с помощью WINAPI.

**Список использованных источников**

1. Пошаговое руководство. Создание классического оконного приложение [Электронный ресурс]. URL: https://docs.microsoft.com/ru-ru/cpp/windows/walkthrough-creating-windows-desktop-applications-cpp?view=msvc-160
2. Администрирование в Windows 10. [Электронный ресурс] : [https://docs.microsoft.com/ru-ru/windows/client-management/administrative-tools-in-windows-10 /](https://docs.microsoft.com/ru-ru/windows/client-management/administrative-tools-in-windows-10%20/) Ричард Ворд 2020 — 100 с.
3. «Тайм-менеджмент для системных администраторов». / Лимончелли Т.А. М.: Символ-плюс 2007 — 247 с. ISBN:978-5-93286-090-8