Министерство образования Республики Беларусь
Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и
сетей Кафедра информатики
Дисциплина: Операционные среды и системное программирование

ОТЧЕТ

к лабораторной работе № 5

на тему «Реестр и журналы (Windows). Доступ к реестру Windows. Работа с журналами Windows. Другие вспомогательные средства управления»

Выполнил:
студент гр. 153504
Подвальников А.С.

Проверил:
Гриценко Н.Ю.

Минск 2023

СОДЕРЖАНИЕ

1 ЦЕЛИ РАБОТЫ

Изучить механизмы доступа к реестру операционной системы Windows. Изучить работу с журналами Windows. Изучить вспомогательные средства управления. Реализовать приложение для анализа реестра с целью выявления устаревших и ненужных записей, которые могут быть удалены.

2 КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Реестр Windows - это централизованное хранилище информации о конфигурации и настройках операционной системы Windows. В реестре хранятся ключи, значения и подключи, описывающие различные компоненты и параметры системы. Доступ к реестру может быть осуществлен с использованием Windows API функций.

Для работы с реестром Windows разработан ряд функций в рамках Windows API. Из наиболее распространенных функций: RegOpenKeyEx(эта функция позволяет открыть существующий ключ реестра или создать новый, если он не существует. Она позволяет указать путь к ключу, с которым вы хотите работать), RegCreateKeyEx(с помощью этой функции можно явно создать новый ключ реестра. Если ключ уже существует, функция просто откроет его), RegSetValueEx(данная функция позволяет установить значение для указанного ключа реестра. Она может использоваться для создания новых значений или изменения существующих), RegQueryValueEx(с помощью этой функции можно получить значение, связанное с указанным ключом реестра), RegDeleteKey и RegDeleteValue (эти функции используются для удаления ключей или значений из реестра соответственно).

Каждый элемент в реестре называется ключом. Каждый ключ может содержать подключи и значения. Путь к конкретной записи в реестре задается с помощью пути к ключу. Путь к ключу указывается с использованием заранее определенных корневых ключей и разделителя "\". В операционной системе Windows обычно используются следующие корневые ключи: HKEY_CLASSES_ROOT (этот ключ содержит информацию о типах файлов и их связи с приложениями), HKEY_CURRENT_USER (в этом ключе хранятся настройки и конфигурации текущего пользователя), HKEY_LOCAL_MACHINE (этот ключ содержит информацию, относящуюся к компьютеру в целом, включая установленное программное обеспечение и настройки аппаратных компонентов), HKEY_USERS (в этом ключе содержатся настройки для всех активных пользователей на компьютере), HKEY_CURRENT_CONFIG (этот ключ содержит информацию о текущей конфигурации аппаратного обеспечения), HKEY_DYN_DATA (этот ключ служит для системных динамических данных и используется редко).

Путь к ключу в реестре позволяет найти и изменить соответствующие значения или настройки операционной системы и приложений. Перед внесением изменений в реестр рекомендуется создавать резервные копии и быть уверенным в своих действиях.

3 РЕЗУЛЬТАТЫ ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ

В ходе выполнения лабораторной работы было реализовано приложение для анализа реестра с целью выявления устаревших и ненужных записей, которые могут быть удалены. Результат работы программы показан на рисунке 3.1.

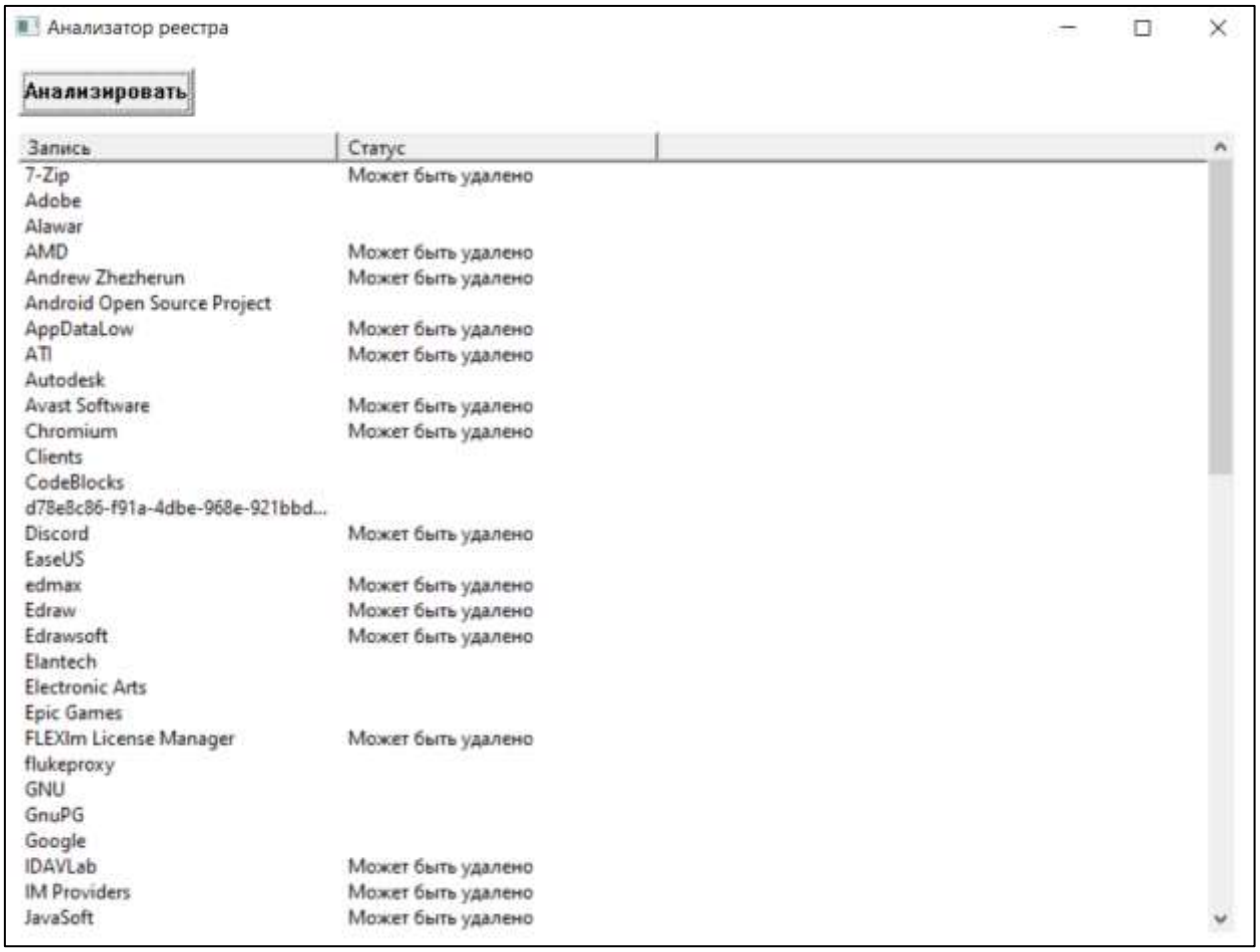

Рисунок 3.1 – Результат работы программы

ВЫВОДЫ

В ходе выполнения данной лабораторной работы были изучены механизмы доступа к реестру операционной системы Windows. Изучена работа с журналами Windows. Изучены вспомогательные средства управления. Реализовано приложение для анализа реестра с целью выявления устаревших и ненужных записей, которые могут быть удалены.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

[1] Щупак Ю. Win32 API. Разработка приложений для Windows. – СПб: Питер, 2008. – 592 с.: ип.

[2] [Электронный ресурс]. – Режим доступа: https://learn.microsoft.com/en-us/windows/win32/api/winreg/ – Дата доступа 20.10.2023

[3] [Электронный ресурс]. – Режим доступа: https://learn.microsoft.com/en-us/windows/win32/sysinfo/enumerating-registry-subkeys – Дата доступа 20.10.2023

[4] [Электронный ресурс]. – Режим доступа: https://learn.microsoft.com/en-us/windows/win32/sysinfo/registry – Дата доступа 20.10.2023

[5] [Электронный ресурс]. – Режим доступа: https://learn.microsoft.com/en-us/windows/win32/sysinfo/about-the-registry – Дата доступа 20.10.2023

[6] [Электронный ресурс]. – Режим доступа: https://learn.microsoft.com/en-us/windows/win32/sysinfo/registry-functions – Дата доступа 21.10.2023

ПРИЛОЖЕНИЕ А
(обязательное)
Листинг кода

Main.cpp

```
// Заголовочные файлы
#include <windows.h>
#include <commctrl.h>
#include <time.h>

// Идентификаторы контролов
#define ID_ANALYZE_BUTTON 1001
#define ID_LISTVIEW 1002

// Преобразование FILETIME в time_t
time_t FileTimeToTimeT(FILETIME* ft);

void AnalyzeRegistry(HWND hListView);

// Обработчик главного окна
LRESULT CALLBACK MainWindowProc(HWND hWnd, UINT message, WPARAM wParam,
LPARAM lParam)
{
    switch (message)
    {
    case WM_COMMAND:
        switch (LOWORD(wParam))
        {
        case ID_ANALYZE_BUTTON:
            // Обработка нажатия кнопки "Анализировать"
            HWND hListView = GetDlgItem(hWnd, ID_LISTVIEW);
            AnalyzeRegistry(hListView);
            break;
        }
        break;

    case WM_DESTROY:
        PostQuitMessage(0);
        break;

    default:
        return DefWindowProcA(hWnd, message, wParam, lParam);
    }

    return 0;
}

// Точка входа в приложение
int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR
```

```
lpCmdLine, int nCmdShow)
{
    // Регистрация класса окна
    WNDCLASSEXA wcex = { sizeof(WNDCLASSEXA) };
    wcex.lpfnWndProc = MainWindowProc;
    wcex.hInstance = hInstance;
    wcex.lpszClassName = "MainWindowClass";
    RegisterClassExA(&wcex);

    // Создание главного окна
    HWND hWnd = CreateWindowExA(
        0,
        "MainWindowClass",
        "Анализатор реестра",
        WS_OVERLAPPEDWINDOW,
        CW_USEDEFAULT, CW_USEDEFAULT, 800, 600,
        NULL, NULL, hInstance, NULL
    );

    if (hWnd == NULL)
        return 0;

    // Создание кнопки "Анализировать"
    HWND hButton = CreateWindowA(
        "BUTTON", "Анализировать",
        WS_CHILD | WS_VISIBLE,
        10, 10, 110, 30,
        hWnd, (HMENU)ID_ANALYZE_BUTTON, hInstance, NULL
    );

    // Создание представления списка
    HWND hListView = CreateWindowExA(
        0,
        WC_LISTVIEWA, "Результаты",
        WS_CHILD | WS_VISIBLE | LVS_REPORT,
        10, 50, 760, 500,
        hWnd, (HMENU)ID_LISTVIEW, hInstance, NULL
    );

    // Добавление колонок в список
    LVCOLUMNA lvColumn = { 0 };
    lvColumn.mask = LVCF_TEXT | LVCF_WIDTH;
    lvColumn.cx = 200;
    lvColumn.pszText = "Запись";
    ListView_InsertColumn(hListView, 0, &lvColumn);

    // Добавление второй колонки в список
    lvColumn.pszText = "Статус";
    ListView_InsertColumn(hListView, 1, &lvColumn);

    // Показ главного окна
```

```
    ShowWindow(hWnd, nCmdShow);
    UpdateWindow(hWnd);

    // Цикл обработки сообщений
    MSG msg;
    while (GetMessageA(&msg, NULL, 0, 0))
    {
        TranslateMessage(&msg);
        DispatchMessageA(&msg);
    }

    return (int)msg.wParam;
}

void AnalyzeRegistry(HWND hListView)
{
    // Очистка представления списка
    ListView_DeleteAllItems(hListView);

    // Получение доступа к корневому ключу реестра
    HKEY hKey;
    if (RegOpenKeyExA(HKEY_CURRENT_USER, "Software", 0, KEY_READ, &hKey)
== ERROR_SUCCESS)
    {
        DWORD subkeyCount;
        DWORD maxSubkeySize;
        RegQueryInfoKeyA(hKey, NULL, NULL, NULL, &subkeyCount,
&maxSubkeySize, NULL, NULL, NULL, NULL, NULL, NULL);

        // Установка временного порога для создания записи (в данном
случае 12 месяцев)
        time_t treshold = time(NULL) - 12 * 30 * 24 * 60 * 60;

        // Перебор подключей реестра
        for (DWORD i = 0; i < subkeyCount; i++)
        {
            char subkeyName[MAX_PATH];
            DWORD subkeyNameSize = MAX_PATH;
            FILETIME ftLastWriteTime;
            if (RegEnumKeyExA(hKey, i, subkeyName, &subkeyNameSize, NULL,
NULL, NULL, &ftLastWriteTime) == ERROR_SUCCESS)
            {
                // Анализ и проверка подключа реестра
                // Добавление информации о записи в представление списка
                LVITEMA lvItem = { 0 };
                lvItem.mask = LVIF_TEXT;
                lvItem.pszText = subkeyName;
                lvItem.iItem = i;
                ListView_InsertItem(hListView, &lvItem);

                // Проверка на устаревшую запись
                if (FileTimeToTimeT(&ftLastWriteTime) < treshold) {
```

```
                    ListView_SetItemText(hListView, i, 1, "Может быть
удалено");
                }
            }
        }

        // Закрытие корневого ключа реестра
        RegCloseKey(hKey);
    }
}

time_t FileTimeToTimeT(FILETIME* ft)
{
    ULARGE_INTEGER ui;
    ui.LowPart = ft->dwLowDateTime;
    ui.HighPart = ft->dwHighDateTime;

    return (time_t)((ui.QuadPart - 116444736000000000ULL) / 10000000ULL);
}
```