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

ОТЧЁТ

к лабораторной работе №5

на тему

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

Выполнил: студент группы 153503

Киселёва Елизавета Андреевна

Проверил: Гриценко Никита Юрьевич

Минск 2023

СОДЕРЖАНИЕ

[1 Постановка задачи 3](#_Toc149265314)

[2 Краткие теоретические сведения 4](#_Toc149265315)

[2.1 Реестр *Windows* 4](#_Toc149265316)

[2.2 Журналы в *Windows* 5](#_Toc149265317)

[2.3 Другие вспомогательные средства управления 6](#_Toc149265318)

[3 Результаты выполнения лабораторной работы 8](#_Toc149265319)

[Выводы 10](#_Toc149265320)

[Список использованных источников 11](#_Toc149265321)

[Приложение А (обязательное) Листинг кода 12](#_Toc149265322)

## 1 ПОСТАНОВКА ЗАДАЧИ

Целью выполнения лабораторной работы является создание приложения для мониторинга изменений в системных параметрах и автоматической реакции на изменения, например, изменение разрешения экрана. Для этого будет создано приложение, сменяющее цвет окна в зависимости от выбранного клавиатурного языка.

## 2 КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

## 2.1 Реестр *Windows*

Реестр – это определяемая системой база данных, в которой приложения и системные компоненты хранят и извлекают данные конфигурации.

В реестре информация представлена в виде иерархии ключей и значений. Ключи можно рассматривать как папки, а значения – как файлы. Каждый ключ может содержать подключи (подразделы) и значения.

Примеры данных, хранящихся в реестре:

1 Настройки операционной системы: информация о конфигурации устройств, параметры работы операционной системы и ее компонентов.

2 Настройки приложений: многие программы хранят свои настройки и параметры в реестре.

3 Информация о пользователях: некоторые параметры пользователя и профили также хранятся в реестре.

4 Системные параметры: например, информация о программных и аппаратных ресурсах, параметры безопасности и т.д.

Для взаимодействия с реестром в *Windows* обычно используют *API* функции, предоставляемые *Windows* *API*, такие как *RegOpenKeyEx*, *RegCreateKeyEx*, *RegQueryValueEx* и другие. Важно быть осторожным при внесении изменений в реестр, поскольку неправильные изменения могут повлиять на стабильность и производительность системы.

Реестр заменяет множество текстовых *INI*-файлов, использованных в настройках *Windows* 3.*x* и *MS-DOS*, таких как *Autoexec.bat* и *Config.sys*. Несмотря на то, что реестр широко используется в различных версиях операционных систем *Windows*, между ними все же имеются определенные отличия. Куст реестра представляет собой группу ключей, подразделов и значений в реестре, а также соответствующие файлы резервных копий данных. Файлы резервных копий для всех кустов, кроме *HKEY\_CURRENT\_USER*, располагаются в папке %*SystemRoot*%\*System32*\*Config* в *Windows* *NT* 4.0, *Windows* 2000, *Windows* XP, *Windows* *Server* 2003 и *Windows Vista*. Файлы резервных копий для *HKEY\_CURRENT\_USER* находятся в папке %*SystemRoot*%\*Profiles*\*Username*. Расширения файлов в этих папках указывают на тип содержащихся данных. Также следует отметить, что иногда отсутствие расширения может также указывать на тип данных в файле. [1]

## 2.2 Журналы в *Windows*

Журналы *Windows* – это системные журналы событий, которые записывают информацию о различных событиях и состоянии операционной системы. Эти журналы предоставляют ценную отладочную и диагностическую информацию, помогая пользователям и администраторам отслеживать проблемы, мониторить работу системы и выявлять потенциальные угрозы безопасности.

В операционных системах *Windows* существует несколько типов журналов:

1 Журнал системы (*System Log*). Записи в этом журнале касаются системных событий, таких как ошибки драйверов устройств, загрузка операционной системы и другие важные системные события.

2 Журнал безопасности (*Security Log*). Содержит информацию обо всех событиях, связанных с безопасностью, таких как вход в систему, неудачные попытки входа и другие события, связанные с безопасностью.

3 Журнал приложений (*Application Log*). Записывает информацию о работе приложений, включая ошибки и предупреждения, которые могут возникнуть в процессе их выполнения.

4 Журнал служб (*Services Log*). Содержит информацию об изменениях в статусе служб, работающих в системе.

5 Журнал аудита (*Audit Log*). Используется для отслеживания событий аудита, таких как изменения важных настроек системы.

В структуре журнала событий выделяют кусты реестра, которые объединяют ключи, подразделы и значения вместе с вспомогательными файлами, содержащими резервные копии данных. Например, для куста *HKEY\_CURRENT\_USER* вспомогательные файлы располагаются в папке %*SystemRoot*%\*Profiles*\*Username*.

Расширения имен файлов в этих папках указывают на тип данных, содержащихся в файлах. Отсутствие расширения также может иногда указывать на тип данных. Это организация позволяет операционной системе эффективно управлять событиями, а также обеспечивает удобный доступ к данным для различных системных компонентов и приложений.

Для взаимодействия с журналами *Windows* в *Win32 API* используются несколько ключевых функций. *OpenEventLog* позволяет открыть доступ к указанному журналу, а *ReadEventLog* используется для чтения записей из этого журнала. Полученные данные можно обрабатывать, извлекая информацию о событиях. Также, для добавления пользовательских записей в журнал событий, применяется функция *ReportEvent*. Она позволяет регистрировать события, например, для собственного источника событий.

При работе с журналами следует учитывать права доступа, так как для некоторых операций могут потребоваться повышенные привилегии.

Манифест инструментирования (*Application Compatibility Manifest*) – это *XML*-файл, встраиваемый в исполняемый файл приложения. Он может использоваться для указания требований к совместимости приложения с определенными функциональными возможностями операционной системы. Манифесты инструментирования также могут влиять на запись в журналы событий для отслеживания событий, связанных с совместимостью.

Журналы событий (*Event Viewer*) в операционной системе *Windows* регистрируют различные события, происходящие в системе. В контексте манифестов инструментирования, события, связанные с совместимостью, могут включать в себя информацию о проблемах, возникающих из-за несовместимости приложений с текущей версией операционной системы, использования устаревших *API* или других сценариев, влияющих на совместимость.

Эти события обычно классифицируются как информационные, предупреждения или ошибки в зависимости от серьезности проблемы. Они предоставляют администраторам и разработчикам информацию о том, как приложение взаимодействует с операционной системой и о возможных проблемах, которые могут возникнуть.

Манифесты инструментирования могут быть использованы для документирования и указания на то, какие именно события, связанные с совместимостью, должны быть отслеживаемы в журналах событий. Такая информация может быть ценной при отладке, анализе проблем и обеспечении совместимости приложений с различными версиями операционной системы. [2]

## 2.3 Другие вспомогательные средства управления

В операционной системе *Windows* существует набор вспомогательных средств управления, предназначенных для диагностики, конфигурирования и мониторинга системы.

Одним из таких инструментов является *MSCONFIG*, или утилита конфигурации системы, которая позволяет управлять загрузкой операционной системы и службами при старте. Это полезно для выявления и решения проблем, связанных с запуском системы.

Еще одним важным инструментом является *REGEDIT*, редактор реестра, предоставляющий возможность непосредственного редактирования содержимого реестра Windows. Реестр заменяет многие текстовые *INI*-файлы, используемые в более ранних версиях Windows, и позволяет изменять настройки и параметры системы.

*Event Viewer*, или журнал событий, представляет собой визуальный интерфейс для анализа различных событий, происходящих в системе. Это полезное средство для мониторинга и диагностики, так как позволяет отслеживать ошибки, предупреждения и информационные сообщения.

*Task Manager*, или диспетчер задач, предоставляет управление запущенными процессами, службами и мониторит использование ресурсов системы. Этот инструмент обеспечивает контроль над работающими приложениями, процессами и ресурсами, такими как центральный процессор и память.

Эти вспомогательные инструменты имеют общую цель – обеспечение пользователей средствами управления, мониторинга и конфигурирования операционной системы *Windows*. Они находят применение в диагностике проблем, настройке системы под конкретные требования и обеспечивают прозрачность в управлении различными аспектами операционной системы. [3]

## 3 РЕЗУЛЬТАТЫ ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ

В ходе выполнения лабораторной работы было разработано приложение, сменяющее цвет окна в зависимости от выбранного клавиатурного языка с использованием цвета по умолчанию, устанавливаемого в реестр. (Рисунок 3.1).

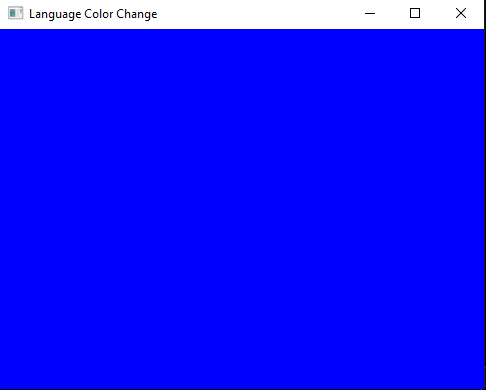
****

Рисунок 3.1 – Результат работы программы при установленном русском языке клавиатуры

Синий цвет окно приложения имеет при выбранном русском языке клавиатуры. При смене языка клавиатуры на английский цвет окна сменится на красный (Рисунок 3.2).

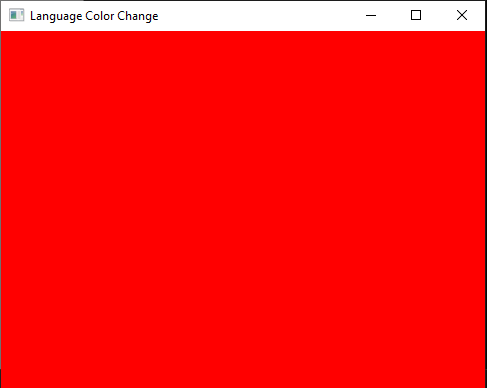


Рисунок 3.2 – Результат работы программы при установленном английском языке клавиатуры

Если язык клавиатуры изменится на тот, что не предусмотрен программой, то окно приобретет тот цвет, который будет находится в реестре. (Рисунок 3.3).

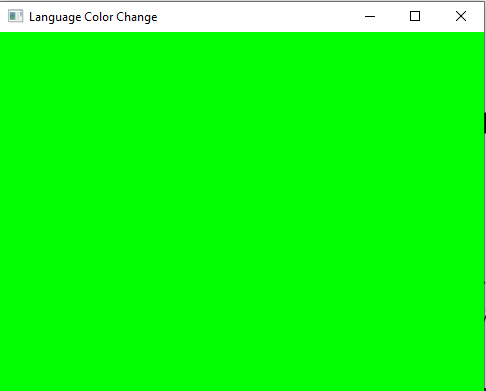


Рисунок 3.3 – В результате работы программы цвет окна взят из реестра

## ВЫВОДЫ

В ходе выполнения данной лабораторной работы было разработано приложение, сменяющее цвет окна в зависимости от выбранного клавиатурного языка с использованием цвета по умолчанию, устанавливаемого в реестр. Проект включает в себя создание графического окна приложения, обработку оконных сообщений, считывание данных системных параметров, работу с реестрами. Результатом стало рабочее приложение для индикации выбранного языка клавиатуры за счет смены цвета окна приложения.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

[1] Microsoft Learn [Электронный ресурс]. – Режим доступа : https://learn.microsoft.com/ru-ru/windows/win32/sysinfo/registry.

[2] Журналы Windows[Электронный ресурс]. – Режим доступа : https://learn.microsoft.com/ru-ru/windows/win32/api/\_wes.

[3] Обзор специальных возможностей [Электронный ресурс]. – Режим доступа : https://learn.microsoft.com/ru-ru/windows/apps/design/accessibility/accessibility-overview.

## ПРИЛОЖЕНИЕ А

## (обязательное)

## Листинг кода

**Source.cpp**

#include <Windows.h>

#include <string>

LRESULT CALLBACK WindowProc(HWND hwnd, UINT uMsg, WPARAM wParam, LPARAM lParam);

void SetBackgroundColor(HWND hwnd, COLORREF color);

COLORREF GetStoredColor();

void StoreColor(COLORREF color);

COLORREF reg\_color = RGB(0, 255, 0);

LANGID langId;

const wchar\_t CLASS\_NAME[] = L"LanguageColorChangeWindow";

const wchar\_t REGISTRY\_KEY[] = L"Software\\LanguageColorChangeApp"; // Gуть к ключу в реестре Windows

const wchar\_t REGISTRY\_VALUE[] = L"BackgroundColor"; // Имя значения в реестре

int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine, int nCmdShow) {

UNREFERENCED\_PARAMETER(hPrevInstance);

UNREFERENCED\_PARAMETER(lpCmdLine);

WNDCLASS wc = {};

wc.lpfnWndProc = WindowProc;

wc.hInstance = hInstance;

wc.lpszClassName = CLASS\_NAME;

wc.hbrBackground = (HBRUSH)GetSysColorBrush(COLOR\_BTNFACE);

RegisterClass(&wc);

HWND hwnd = CreateWindowEx(

0,

CLASS\_NAME,

L"Language Color Change",

WS\_OVERLAPPEDWINDOW,

CW\_USEDEFAULT, CW\_USEDEFAULT, 500, 400,

NULL,

NULL,

hInstance,

NULL

);

if (hwnd == NULL) {

return 0;

}

ShowWindow(hwnd, nCmdShow);

MSG msg = {};

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

TranslateMessage(&msg);

DispatchMessage(&msg);

}

return static\_cast<int>(msg.wParam);

}

LRESULT CALLBACK WindowProc(HWND hwnd, UINT uMsg, WPARAM wParam, LPARAM lParam) {

switch (uMsg) {

case WM\_CREATE:

SetTimer(hwnd, 1, 1000, NULL);

StoreColor(reg\_color);

break;

case WM\_TIMER:

langId = LOWORD(GetKeyboardLayout(0));

COLORREF backgroundColor;

if (langId == 0x0409) { // Английский

backgroundColor = RGB(255, 0, 0); // Красный

}

else if (langId == 0x0419) { // Русский

backgroundColor = RGB(0, 0, 255); // Синий

}

else {

backgroundColor = GetStoredColor(); // Цвет по умолчанию

}

SetBackgroundColor(hwnd, backgroundColor);

break;

case WM\_DESTROY:

KillTimer(hwnd, 1);

PostQuitMessage(0);

break;

default:

return DefWindowProc(hwnd, uMsg, wParam, lParam);

}

return 0;

}

void SetBackgroundColor(HWND hwnd, COLORREF color) {

SetClassLongPtr(hwnd, GCLP\_HBRBACKGROUND, (LONG\_PTR)CreateSolidBrush(color));

InvalidateRect(hwnd, NULL, TRUE);

}

COLORREF GetStoredColor() {

HKEY hKey;

COLORREF color = GetSysColor(COLOR\_BTNFACE);

// Открыть ключ реестра для чтения

if (RegOpenKeyEx(HKEY\_CURRENT\_USER, REGISTRY\_KEY, 0, KEY\_READ, &hKey) == ERROR\_SUCCESS) {

DWORD dataSize = sizeof(color);

// Получить значение цвета из реестра

RegQueryValueEx(hKey, REGISTRY\_VALUE, nullptr, nullptr, reinterpret\_cast<LPBYTE>(&color), &dataSize);

// Закрыть ключ реестра

RegCloseKey(hKey);

}

return color;

}

void StoreColor(COLORREF color) {

HKEY hKey;

// Создать или открыть ключ реестра для записи

if (RegCreateKeyEx(HKEY\_CURRENT\_USER, REGISTRY\_KEY, 0, nullptr, 0, KEY\_WRITE, nullptr, &hKey, nullptr) == ERROR\_SUCCESS) {

DWORD dwData = color;

// Записать значение цвета в реестр

RegSetValueEx(hKey, REGISTRY\_VALUE, 0, REG\_DWORD, reinterpret\_cast<BYTE\*>(&dwData), sizeof(dwData));

// Закрыть ключ реестра

RegCloseKey(hKey);

}

}