Код, который я предоставил, представляет собой простое приложение Windows, которое мониторит и реагирует на изменения в системных параметрах, таких как разрешение экрана, язык ввода, обои рабочего стола, системные цвета и шрифт системы. Давайте рассмотрим его подробнее.

### Введение

В операционной системе Windows, для мониторинга и реагирования на системные изменения, вы можете использовать различные механизмы, включая обработку сообщений Windows и вызовы API Windows. Наше приложение будет мониторить изменения разрешения экрана, языка ввода, обоев рабочего стола, системных цветов и шрифта системы.

### Теоретические сведения

#### 1. Регистр Windows (Реестр)

Реестр Windows - это централизованное хранилище системных настроек и параметров. Все системные компоненты и приложения могут сохранять и читать данные из реестра. Для работы с реестром в Windows обычно используются API-функции, такие как `RegCreateKeyEx`, `RegOpenKeyEx`, `RegSetValueEx`, и `RegQueryValueEx`. Реестр представляет собой древовидную структуру, аналогичную файловой системе, с узлами, называемыми ключами, и значениями, хранящими фактические данные.

#### 2. Журналы Windows (Event Logs)

Журналы Windows представляют собой системный механизм записи и хранения событий, происходящих в операционной системе. Эти события могут быть системными, приложениями, службами и так далее. Журналы могут содержать информацию о предупреждениях, ошибках, информационных событиях и т. д. Для работы с журналами Windows, вы можете использовать функции API, такие как `OpenEventLog`, `ReadEventLog`, `ReportEvent`, и другие.

#### 3. Обработка сообщений Windows

Windows-приложения обычно работают в цикле обработки сообщений. Они ожидают и обрабатывают сообщения от операционной системы или пользовательских действий. Эти сообщения могут быть связаны с клавишами, мышью, окнами и другими событиями. В коде выше, цикл обработки сообщений осуществляется в функции `WinMain`, и все сообщения, такие как `WM\_SETTINGCHANGE`, `WM\_SYSCOLORCHANGE` и `WM\_FONTCHANGE`, обрабатываются в функции `WndProc`.

### Обзор кода

1. Мы определяем несколько глобальных переменных, чтобы отслеживать текущее разрешение экрана и язык ввода, а также флаги, чтобы убедиться, что сообщение об изменении параметра выводится только один раз.

2. В функции `HandleResolutionChange`, `HandleInputLocaleChange`, `HandleSystemParamChange`, `HandleSystemColorChange` и `HandleFontChange`, мы проверяем, изменилось ли значение, и если да, выводим соответствующее сообщение об изменении параметра.

3. В функции `WinMain`, мы регистрируем класс окна, создаем окно, инициализируем цикл обработки сообщений и устанавливаем таймер, чтобы регулярно проверять изменения в разрешении и языке ввода.

4. В функции `WndProc`, мы обрабатываем различные сообщения, включая те, которые связаны с изменениями в параметрах системы, и вызываем соответствующие функции для обработки этих изменений.

### Как это работает

При запуске приложения, оно создает окно и начинает ожидать сообщений. При изменении разрешения экрана, языка ввода, обоев, системных цветов или шрифта, соответствующая функция будет вызвана, и сообщение об изменении параметра будет выведено.

Таймер, установленный в `WM\_CREATE`, позволяет периодически проверять изменения разрешения экрана и языка ввода.

### Доработки и расширения

1. Вы можете дополнить код другими обработками параметров или событий, которые вас интересуют.

2. Вы можете сохранять записи о параметрах в журнал Windows Event Log для последующего анализа.

3. Обратите внимание, что код может выдавать множество сообщений об изменениях параметров в первый раз после запуска, так как он сравнивает текущие параметры с нулевыми значениями. Вы можете дополнительно настроить начальные значения параметров, чтобы избежать первоначальных сообщений.

4. Обработка других сообщений Windows и изменений параметров может потребовать дополнительных знаний и исследований в зависимости от конкретных требований вашего приложения.

Данный код представляет собой Windows-приложение, которое мониторит изменения в системных параметрах, таких как разрешение экрана, язык ввода и обои рабочего стола, и регистрирует события в журнале Windows Event Log. Вот подробное описание каждой части кода и связанной с ней теоретической информации:

1. \*\*Включение заголовочных файлов:\*\*

```cpp

#include <Windows.h>

#include <tchar.h>

#include <evntrace.h>

```

Здесь включаются заголовочные файлы, необходимые для работы с Windows API и журналами Windows Event Log. `evntrace.h` используется для работы с API журналов событий.

2. \*\*Объявление глобальных переменных:\*\*

```cpp

int mainScreenWidth = -1;

int mainScreenHeight = -1;

HKL mainInputLocale = NULL;

```

В коде объявляются глобальные переменные, в которых будут храниться текущие параметры мониторинга: разрешение экрана и язык ввода.

3. \*\*Функция `LogEventToEventLog`:\*\*

```cpp

void LogEventToEventLog(LPCTSTR message) { ... }

```

Эта функция создает запись в журнале Windows Event Log с указанным сообщением. Она использует функции Windows API, такие как `RegisterEventSource`, `ReportEvent` и `DeregisterEventSource`, для регистрации события. После этого она также выводит сообщение в стандартном диалоговом окне `MessageBox` для наглядности.

4. \*\*Функции `HandleResolutionChange`, `HandleInputLocaleChange`, `HandleSystemParamChange`:\*\*

Эти функции отвечают за обработку изменений разрешения экрана, языка ввода и других системных параметров. Если параметры изменились, функции вызывают `LogEventToEventLog` для регистрации события.

5. \*\*Функция `WinMain`:\*\*

```cpp

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

```

Эта функция представляет точку входа приложения. Она инициализирует окно, обрабатывает сообщения и вызывает функции мониторинга.

6. \*\*Функция `WndProc` (обработчик оконных сообщений):\*\*

```cpp

LRESULT CALLBACK WndProc(HWND hwnd, UINT msg, WPARAM wParam, LPARAM lParam) { ... }

```

Эта функция отвечает за обработку сообщений, отправляемых окну. Она устанавливает таймер для периодической проверки параметров и вызывает соответствующие функции мониторинга при получении определенных сообщений, таких как изменение разрешения экрана, языка ввода и системных параметров.

Когда приложение запущено, оно создает окно, и по таймеру и при получении определенных системных сообщений мониторит изменения параметров. Если какой-либо параметр изменился, оно регистрирует это событие в журнале Windows Event Log с помощью `LogEventToEventLog`. Кроме того, оно также выводит сообщения в стандартном диалоговом окне `MessageBox` для наглядности.