МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ «ВИТЕБСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ П. М. МАШЕРОВА»

Факультет математики и информационных технологий

Кафедра информационных технологий и управления бизнесом

КУРСОВОЙ ПРОЕКТ

по дисциплине «Операционные системы и системное программирование»

ПРИЛОЖЕНИЕ ДЛЯ РЕДАКТИРОВАНИЯ СИСТЕМНОГО РЕЕСТРА WINDOWS

.

Царенко Антон Дмитриевич,

3 курс, 34 группа

Научный руководитель:

Витько Елена Анатольевна,

доцент кафедры информационных технологий и управления бизнесом,

кандидат физико-математических наук

Витебск, 2022

**Реферат**

Курсовая работа 19 стр., 6 рис., 8 листингов, 5 источников.

WINDOWS, РЕЕСТР, WINAPI.

**Объект исследования** – технологии для разработки приложения редактора реестра.

**Предмет исследования** – алгоритмы проектирования и разработки приложений для взаимодействия с реестром Windows.

**Цель работы** – разработка приложения, позволяющего редактировать реестр операционной системы Windows с использование Windows API.

**Задачи**:

1. Выполнить анализ открытых источников информации, литературы в сфере базовых принципов функционирования операционных систем, а также реестра Windows.
2. Ознакомиться с особенностями реализации приложения на основе Windows API (язык программирования C++).
3. Подготовить план разработки приложения.
4. Разработать приложение.

**Методы исследования** – изучение документации, материалов научных и периодических изданий по теме исследования, существующих систем, общенаучные методы исследования (описание, анализ и т.д.).

**Теоретическая значимость и элементы новизны**: совершенствование знаний в области операционных систем, разработка приложения редактора реестра.

**Практическая значимость**: разработанное приложение позволяет редактировать реестр Windows.

Содержание

[Введение 4](#_Toc121300577)

[1 Обоснование выбора средства реализации и его краткая характеристика 5](#_Toc121300578)

[2 Общая характеристика и функционал разрабатываемого приложения. Этапы разработки 7](#_Toc121300579)

[2.1 Общая характеристика и функционал разрабатываемого приложения 7](#_Toc121300580)

[2.2 Описание этапов разработки 7](#_Toc121300581)

[Заключение 18](#_Toc121300582)

[Список использованных источников 19](#_Toc121300583)

Введение

Системный реестр Windows – это иерархическая база данных, содержащая данные, критически важные для работы Windows и приложений и служб, работающих на Windows.

Данные в реестре структурированы в виде дерева. Каждый узел в дереве является разделом. Каждый раздел может содержать как подразделы, так и записи данных, называемые параметрами. Каждый параметр представляет собой пару название – значение определенного типа.

**Цель работы** – разработка приложения, позволяющего редактировать реестр Windows, с использованием Windows API (WinAPI).

Для достижения поставленной цели необходимо решить следующие **задачи**:

1. Выполнить анализ открытых источников информации, литературы в сфере функционирования ОС Windows.
2. Ознакомиться с особенностями разработки приложения на основе WinAPI.
3. Составить план разработки приложения и подготовить необходимые шаблоны и библиотеки.

**Методы исследования** – изучение документации, материалов научных и периодических изданий по теме исследования, существующих систем разработки редакторов, общенаучные методы исследования (описание, анализ, классификация, сравнение, аналогия).

Работа включает введение, 2 главы, заключение, список использованных источников. В первой главе предоставлено обоснование выбора средства и платформы для разработки, а также языка программирования. Во второй главе приводится общая характеристика, функционал приложения и описание этапов разработки.

1 Обоснование выбора средства реализации и его краткая характеристика

Windows API — общее наименование набора базовых функций [интерфейсов программирования приложений](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F_%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B9) ОС семейств [Microsoft Windows](https://ru.wikipedia.org/wiki/Microsoft_Windows) корпорации [Майкрософт](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D0%B9%D0%BA%D1%80%D0%BE%D1%81%D0%BE%D1%84%D1%82).

WinAPI предоставляет прямой способ взаимодействия приложений пользователя с операционной системой Windows. Используется для написания [прикладных программ](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D0%BA%D0%BB%D0%B0%D0%B4%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5), предназначенных для работы под управлением операционной системы MS Windows. Работа через Windows API — это наиболее близкий к операционной системе способ взаимодействия с ней из прикладных программ.

К преимуществам WinAPI можно отнести:

1. Представляет множество функций, структур данных и числовых констант для создания оконных приложений.
2. Функции WinAPI – функции самой ОС. Отсюда следует, что при использовании WinAPI повышается надежность, стабильность и скорость кода.
3. Разработка приложений, которые успешно работают во всех версиях Windows, используя преимущества функций и возможностей, уникальных для каждой версии.
4. Наличие руководства пользователя по разработке и использованию WinAPI, что упрощает процесс освоения инструментов разработки.

Так как WinAPI создавался для языка С, то в нем имеется целый ряд недостатков в применении, как-то: большой объем ручного кодирования для решения простейших задач, С-стиль, плохо выглядящий в C++-приложениях и т.д.

Все языки программирования, способные вызывать функции и оперировать типами данных, следующих соглашениям языка C и конвенции вызовов stdcall, могут пользоваться Windows API, в частности:

1. C++,
2. C#,
3. Pascal,
4. Visual Basic.

В качестве используемого языка был выбран C++, на котором уже имеется опыт разработки, полученный во время обучения.

2 Общая характеристика и функционал разрабатываемого приложения. Этапы разработки

2.1 Общая характеристика и функционал разрабатываемого приложения

Целью разрабатываемого приложения является редактирование реестра Windows.

Базовый функционал приложения позволяет:

1. Просматривать, создавать и удалять разделы реестра;
2. Просматривать, создавать, редактировать и удалять параметры реестра.

2.2 Описание этапов разработки

Этапы проекта в соответствии с каскадной моделью:

1. Подготовка и проектирование.

Первым шагом в разработке является подготовка, в процессе которой изучается документация, выдвигаются технические требования, процесс разделяется на ряд мелких задач, ставятся сроки их выполнения.

Техническое задание определяет название средства, описание, основное его назначение, требования к выполнению и результатам работы. В ходе проектирования были выдвинуты следующие технические требования:

1. Возможность создавать и удалять разделы реестра;
2. Возможность создавать, редактировать и удалять параметры реестра.

Следующим этапом после подготовки идёт проектирование, в ходе которого разрабатываются системы и алгоритмы взаимодействия данных, а также структурные компоненты.

1. Реализация приложения.

Классическое приложение Windows состоит из нескольких частей:

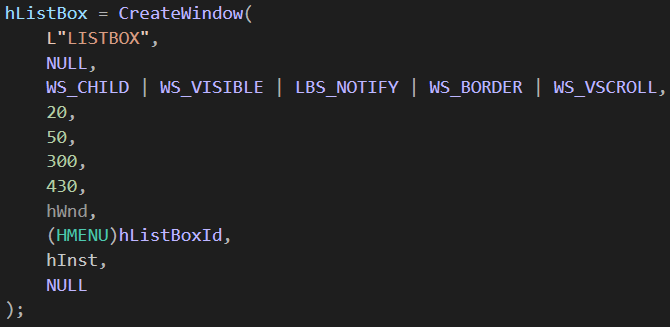
1. Функция wWinMain – точка входа в программу. Здесь создается главное окно приложения, а также запускается цикл обработки сообщений;
2. Функция WndProc – обрабатывает сообщения в главном окне.

Главное окно приложения будет поделено на 3 части:

1. Окно со списком разделов реестра;
2. Окно со списком параметров выбранного раздела;
3. Форма для добавления/редактирования раздела/параметра.

Просмотр разделов будет происходить следующим способом: пользователь делает двойной клик на элементе списка, затем список очищается и заполняется всеми подразделами выбранного раздела. Для возвращения назад на первую позицию в списке будет помещена специальная строка:

Листинг 1 – создание списка разделов реестра.



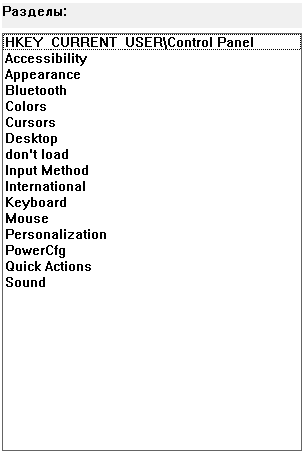
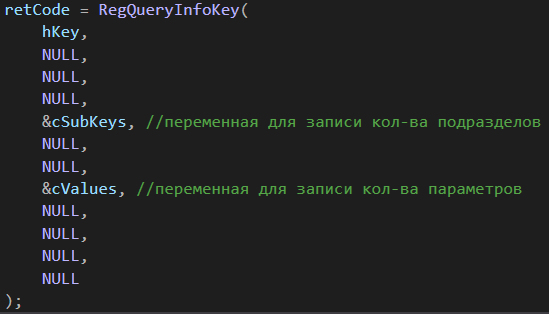


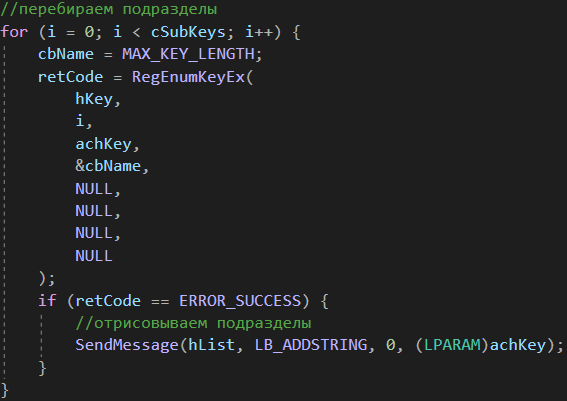
Рисунок 1 – окно для просмотра разделов.

Обработка двойного клика будет происходить следующим способом: при двойном клике по элементу главному окну отправляется сообщение WM\_COMMAND (младшее слово wParam – id окна со списком, старшее – LBN\_DBLCLK), при получении которого будет вызвана функция RegQueryInfoKey, которая вернет количество подразделов выбранного раздела. Затем в цикле будет происходить вызов функции RegEnumKeyEx для получения названия следующего подраздела и его отображения в списке.

Листинг 2 – получение количества подразделов и параметров.



Листинг 3 – получение название подразделов и их отрисовка.



Под списком разделов будут находиться две кнопки: добавление и удаление раздела.

При добавлении раздела будет создаваться соответствующая форма (рисунок 2). В форме будет задаваться название раздела. Создание раздела будет происходить с помощью функции RegCreateKeyExW.

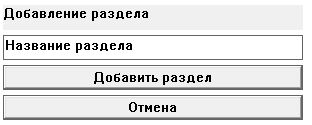
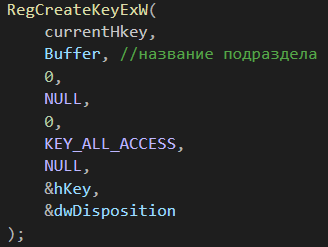


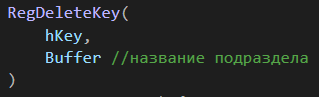
Рисунок 2 – форма для добавления раздела.

Листинг 4 – создание подраздела.



Для удаления раздела пользователю нужно будет выбрать его в списке, затем нажать соответствующую кнопку. Удаление будет происходить с помощью функции RegDeleteKey:

Листинг 5 – удаление подраздела.



Следующая часть приложения – окно со списком параметров.

Заполнение окна будет происходить при двойном клике на элементе списка разделов, при этом отобразятся все параметры выбранного раздела. Для начала с помощью функции RegQueryInfoKey будет получено количество параметров, затем в цикле будет вызываться функция RegEnumValue для получения имени следующего параметра и его отображения.

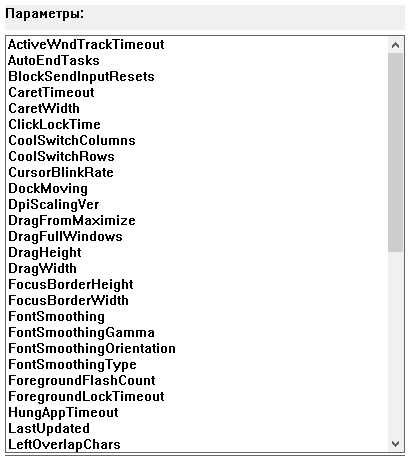
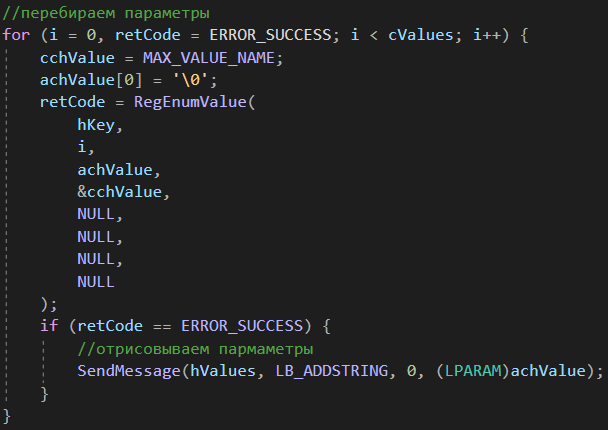


Рисунок 3 – окно со списком параметров.

Листинг 6 – получение названий и отрисовка параметров.



Под окном со списком параметров будут находиться две кнопки: создание и удаление параметра.

При нажатии на кнопку создания окна будет создаваться соответствующая форма:



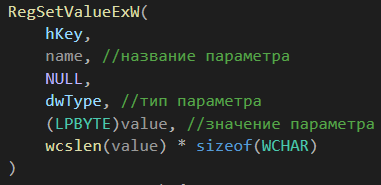
Рисунок 4 – форма для добавления параметра.

Форма содержит 3 поля: поле для ввода названия параметра, выпадающий список с типами данных, поля для ввода значения. Тип параметра может быть одним из следующих:

1. REG\_SZ – строка, завершающаяся символом NULL. Может быть как строкой Юникода, так и строкой ANSI;
2. REG\_BINARY – двоичные данные в любой форме;
3. REG\_DWORD – 32-разрядное число;
4. REG\_QWORD – 64-разрядное число;
5. REG\_MULTI\_SZ – массив строк, завершающихся значением NULL.
6. REG\_EXPAND\_SZ – строка, завершающаяся значением NULL, содержащая необъявленные ссылки на переменные среды.

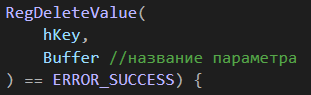
Добавление параметра осуществляется с помощью функции RegSetValueExW.

Листинг 7 – добавление параметра.



Для удаления параметра пользователю необходимо выбрать элемент в списке, а затем нажать на кнопку удаления. Удаление происходит с помощью функции RegDeleteValue:

Листинг 8 – удаление параметра.



Для просмотра значения параметра необходимо сделать двойной клик по элементу списка параметров. При этом откроется форма для редактирования параметра, заполненная соответствующими данными:

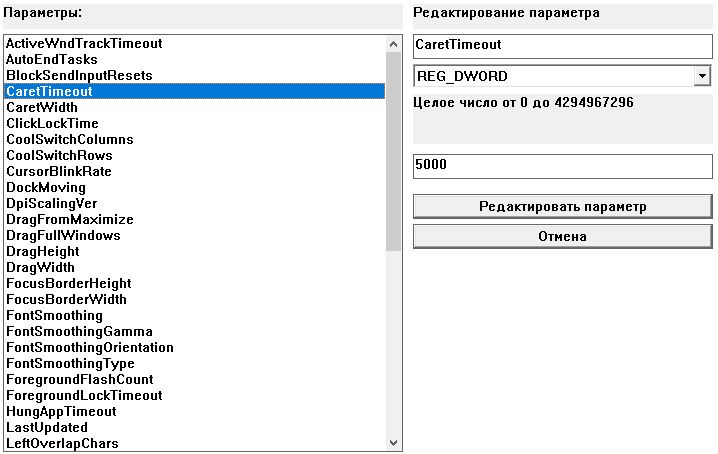


Рисунок 5 – просмотр/редактирование параметра.

При редактировании параметра сначала происходит удаление старого значения (функция RegDeleteValue), затем добавление нового параметра (функция RegSetValueExW).

В конечном итоге окно приложения выглядит следующим образом:

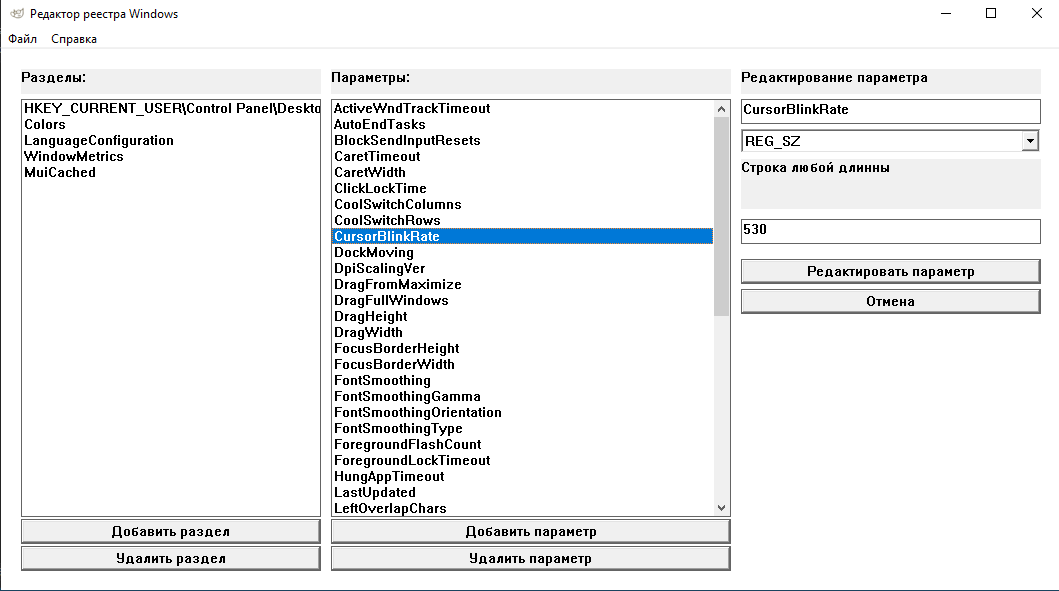


Рисунок 6 – окно приложения.

1. Тестирование.

Этап тестирования подразумевает детальную проверку каждого элемента программы с целью нахождения ошибок или неточностей в работе разрабатываемого средства.

В ходе разработки проводилось модульное, интеграционное и системное тестирование.

Модульное тестирование – процесс тестирования, позволяющий проверить работу отдельных модулей исходного кода программы на корректность. На этом этапе проверялась каждая отдельная функция (например, функция создания параметра в реестре).

Интеграционное тестирование – это тестирование, при котором программные модули объединяются логически и тестируются как группа (например, часть приложения для работы с разделами).

Системное тестирование – это тестирование программного обеспечения, выполняемое на полной, интегрированной системе, с целью проверки соответствия системы исходным требованиям, как функциональным, так и не функциональным.

1. *Написание документации.*

Завершающим этапом разработки редактора является документирование результатов работы. В случае курсовой работы документированием является детальное изложение каждого этапа разработки, описание технического задания.

Заключение

В данной курсовой работе выполнена разработка приложения, позволяющего редактировать реестр Windows, с использованием WinAPI.

В результате проделанной работы были решены следующие задачи:

1. Проанализированы открытые источники информации, литература по теме курсовой работы.
2. Изучены особенности создания приложений на WinAPI.
3. Подготовлен план разработки редактора реестра Windows.
4. Реализовано приложение.

Созданное приложение с заданным набором функций свидетельствует о выполнении поставленных задач и о том, что цель работы достигнута.

Список использованных источников

1. Руссинович М. Внутреннее устройство Microsoft Windows. 6-е изд. / М. Руссинович, Д. Соломон — СПб.: Питер, 2013. — 800 с.
2. Рихтер Дж. Windows для профессионалов: создание эффективных Win32 приложений с учетом специфики 64 разрядной версии Windows / Пер. с англ. — 4-е изд. / Дж. Рихтер — Спб.: Питер; М.: Издательство «Русская Редакция»; 2008. — 720 стр.
3. Build desktop Windows apps using the Wind32 API – Mode of access: <https://learn.microsoft.com/en-us/windows/win32/> – Date of access: 28.11.2022
4. Registry Reference – Mode of access: https://learn.microsoft.com/ru-ru/windows/win32/sysinfo/registry-reference – Date of access: 28.11.2022
5. Прата Стивен Язык программирования С++. Лекции и упражнения, 6-е изд.: Пер. с англ. – М.: ООО «И.Д. Вильямс», 2012. \ Стивен Прата – 1248 с.