Министерство образования Республики Беларуси

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет Компьютерных систем и сетей  
Кафедра Информатики

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту

на тему: "Журналирование процессов и файловой системы в ОС "Windows" по сетевым протоколам"

Дисциплина: Операционные системы и среды (ОСиС)

Выполнил:  
Студент гр. 653505  
Сташинский Г.В.

Проверила:  
ассистент кафедры информатики  
Шнейдер В.

**Содержание**

Введение 3

1. Анализ предметной области 5

1.1. Аппаратное часть ПК 5  
1.2. ОС: процессы и потоки 5  
1.3. ОС: базовые сведения о прикладном ПО и самой операционной системе 6  
1.4. Обзор аналогов 7

2. Программная реализация 9

2.1. Мотивации выбора. 9

2.2. Язык C# и платформа .NET Framework 9

2.3. Windows Presentation Foundation. 10

2.4. Применяемые архитектурные решения 12

2.5. Детали реализации 13

3. Демонстрация работы с ПО 21

3.1. Пользовательский интерфейс 21

3.2. Элементы управления 22

Заключение. 26

Использованная литература. 28

**Введение**

Современные компьютерные системы требуют постоянного мониторинга состояния для избежания различных ошибок. К таким ошибкам можно отнести принудительные остановки процессов ОС, скорость работы и многое другое. Механизмы для журанлирования во многом похожи и позволяют производить мониторинг прямо «на месте», например, системным администратором. Существуют также механизмы, которые позволяют проводить журналирование и мониторинг удаленно, однако, зачастую такие системы требуют большого количества оборудования, навыков и, иногда, избыточны в функционале. Например, логгирование состояния СУБД Oracle требует как минимум глубоких навыков администрирования БД. Еще одной проблемой, которая может возникнуть, является отсутствие нужных систем. Например, СУБД Oracle требует собственного SMTP-cервера для отправки сообщений конечному пользователю.

В данной работе поставлена задача разработки системы удаленного журналирования процессов и файловой системы ОС Windows (а так же Linux посредством API), которая позволит без особых навыков системного администрирования задавать журналируемые события и конфигурировать подписчиков и инструменты их оповещения в конфигурационных файлах. В данной работе будут представлены следующие инструменты для оповещения: оповещение по локальной сети (посредством UDP-протокола), а так же отправка сообщений на почту посредством SMTP.

Особенностью данной системы является простота. В частности, она не требует наличия собственного SMTP-cервера. Это важно, например, для организаций, которые не обладаю большим бюджетом или вообще не связаны с автоматизацией процессов и чем-то подобным. Такие организации могут установить в качестве исходящего почтового ящика любой существующий ящик на популярных сервисах: Gmail, Yandex.Mail, Yahoo и т.д. Главное условие – наличие открытого доступа к серверу. Второй особенностью данной системы является, например, возможность широковещательной рассылки клиентам по локальной сети. Такая система удобна, например, при командной разработке ПО или при администрировании массива компьютерных систем сразу на нескольких машинах. Следует отметить, что современные процессы разработчки программного обеспечения включают многочисленные этапы, которые позволяют не только ускорять процесс разработки, но и обеспечить стабильность и корректность работы ПО. Такими этапами являются логгирование, удаленные и локальные репозитории, тестирование и многое другое. Однако, существуют ситуации, которые не покрыты тестами и они выявляются в процессе локальной разработки. Такие ошибки анализируются прямо на месте (в окне браузера, в лог-файлах), однако такой подход не является удобным. Именно для таких нужд может и потребоваться

**1. Общие характеристики проекта**

1.1. Назначение

Основной областью применения программного продукта, разработанного в рамках данной работы, является логгирование состояния операционной системы и её внутренних процессов в рамках локальной сети посредством протокола UDP, а также удаленно посредством рассылки журналов логгирования на электронную почту через SMTP. Данное решение может применятся при отладке кода, удаленной и локальной диагностике состояния компьютера/системы компьютеров.

1.2. Область действия

В данной работе библиотека реализована в качестве готовой кроссплатформенной .NET Standard Library сборки в виде API-сервисов, поэтому решение может быть использовано как на ОС Windows, так и на Linux (клиентское UI приложение доступно под ОС Windows, терминальный клиент доступен через dotnetcore в Linux и Windows).

Логгирование в локальной сети может быть полезно в закрытых системах, в которых нет доступа к сети Интернет, однако крайне важно наличие системы диагностики. К таким закрытым системам можно причислить лаборатории компаний, в которых изучаются отдельные вредоносные программы (выход в глоабальную сеть отсутствует в целях безопасности), а также организации или предприятия с большим количеством взаимосвязанного оборудования, которые соединены через локальную сеть.

Логгирование через рассылку электронных журналов на электронную почту также представляет интерес, т.к. журналы могут рассылаться на независимые почтовые сервера и в случае повреждения компьютера или сети компьютеров данные не будут утеряны и можно будет провести диагностику.

1.3. Определения, сокращения

Application Programming Interface (API) - программный интерфейс приложения,  описание способов которыми одна компьютерная программа может взаимодействовать с другой программой. Может быть реализована как готовая сборка, так и позволять работать посредством сетевых протоколов.

System Remote Logger – удаленный системный журналист (логгер). Название продукта, реализованного в рамках данной работы.

System Remote Logger (SRL) API – API-сервисы, реализованные в рамках данной работа. Представлены в качестве .NET Standard сборки SystemRemoteLogger.Services. Для уточнения протокола взаимодействия со сборкой можно обратиться к тестам SystemRemoteLogger.Services.Tests.

1.4. Аналоги

Т.к. двумя ключевыми особенностями данного ПО является имено рассылка по локальной сети и по почте, но разделим аналоги по двум катеориям:

* По почте:
  + СУБД Oracle: позволяет производить рассылку сообщений всем подписчикам. Основной недостаток: сложность конфигурирование и обязательные технические навыки по работе с данной СУБД
  + Библиотека Logging (Python): позволяет логгировать действия на рантайме в Python в файл или по почте. Основной недоcтаток: работает только с скриптами Python, нельзя запускать на уровне приложения ОС
* По рассылке по локальной сети:
  + Аналогов рассылки именно журналов ОС не было обнаружено, однако существуют различные инструменты по рассылке файлов по локальной сети. Из недостатков: слишком шировкая область применения и отстутствует гибкая система конфигурации

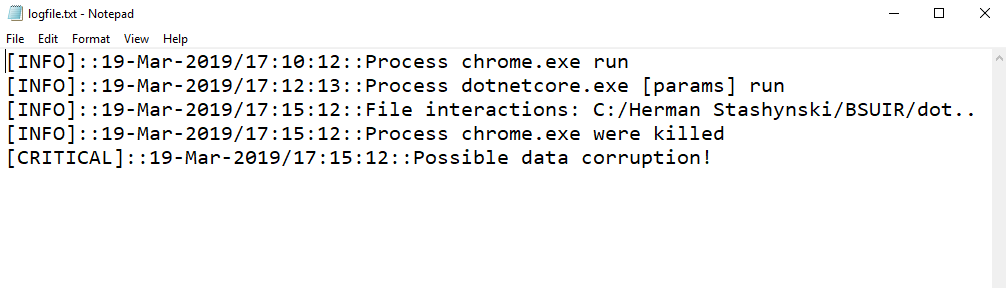
1.5.Краткий обзор

Данное решение состоит из 2 основных компонент:

* API-сервисов в виде сборки **SystemRemoteLogger.Services**
* WPF-клиента в виде сборки **SystemRemoteLogger.WPF**

Также к данной работе прилагаются Unit и интеграционные тесты в сборке **SystemRemoteLogger.Services.Tests**. Они позволят наглядно разобрать как устроен SRL API, а также гарантируют исправность работы кода на данных тестовых случаях.

Результатом работы являются текстовые сообщения (для поддерживания актуальной информации и обеспечения стабильности передаваемых данных) с системной информацией, которые присылаются по локальной сети всем подписчиками, файлы с журналами диагностики на электронной почте. Данная информация форматируется как стандартыне журналы с лог файлами. Ниже представлен возможный пример лог-файла, который присылается по почте:

  
Рис. 1.1. Пример лог-файла присылаемого на почтовый сервер

Клиентская часть реализована при помощи библиотеки MDIX (Material Design In XAML), которая позволяет реализовать Material Design компоненты для .NET приложений.

**2. Общее описание**2.1 Взаимодействие продукта (с другими продуктами и компонентами)

Данная система сервисов включает в себя ряд зависимостей, как низкоуровневых, так и высокоуровневых, которые позволяют реализовывать определенные задачи.

В частности, для получения информации о состоянии системы используются как высокоуровневые инструменты платформы .NET (FileSystemWatcher, Process), так и низкоуровневые запросы к ОС посредством языка запросов. Также используются инструменты WinAPI.

Говоря о низкоуровневых компонентах, стоит упомянуть реализацию сетевого общения в данном ПО:

1. UDP Multicast. В данной программе для реализации широковещательной трансляции всем подписчикам в локальной сети используются низкоуровневые инструменты, такие как System.Net.Sockets.UdpClient. Сборка System.Net.Sockets представляет из себя обертку, которая предоставляет базовый функционал для создания веб-сокетов. Сама сборка работает с неуправляемым кодом и создает соединения, которые нужно высвобождать. Однако, часть забот берет на себя платформа .NET.
2. SMTP. В данной работе журналирование посредством электронной почты реализовано на уровне схожего с предыдщим пунктом инструмента: System.Net.Mail.SmtpClient. Данная сборка имеет наименьший возможный функционал из доступных на рынке для .NET. Альтернатива – использование MailKit, MimeKit (атрибутировано Obsolete компанией Microsoft).

Также следует поговорить и высокоуровневых компонентах для создания интерфейса. Таким инструментом сдал MDIX (Material Design In XAML). Данный фреймворк позволяет создавать полноценные десктопные приложения в соответствии с Google Material Design. Это решение включает в себя не только систему визуальных стилей для стандартных компонентов XAML, но и анимации, которые применяются в Google Material Design.

Сам же итотовый продукт данной работы представляет из себя (говоря именно о ядре проекта) – сборку из API-сервисов «System Remote Logger (SRL) API». Данная сборка легко интегрируется в любые решения и полностью соотвествует принципам SOLID.

2.2 Функции продукта

В рамках кратакого обзора и описания проекта были перечислены основные направления, в рамках которого осуществляется деятельность данного проекта.

Говоря о функциях проекта, следует говорить о предоставленном сборкой SystemRemoteLogger.Services API, т.к. его методы раскрывают основные функции и цели данного проекта.

Отметим, что в данной таблице будут перечислены не все методы и сервисы, предоставленные в данной библиотеке. Данная ставит перед собой цель предоставить представление о возможностях данного API, которые выражают функции данного проекта:

|  |  |  |
| --- | --- | --- |
| MailSender [IMailSender] | Данный сервис ответственен за отправку сообщений по протоколу SMTP | SendMail представляет из себя обработчик для события сервиса RemoteLoggingService. Отправляет сообщения в соответствии с именем хоста и порта.  Сервис требует информацию для конфигурирования SMTP Client, а также экземпляр провайдера конфигурации. |
| UdpConnectionService | Данный сервис предоставляет возможность логгирования сообщений в поток передачи UDP Multicast | Обладает таким же методом SendMail, создан для тех же целей. Внутри также присутствует метод Сonnect для установки соединения.  Сервис подавляет возникшие в процессе работы исключения и логгирует эут ифнормацию в поток журнала (по почте или в UDP Multicast). |
| RemoteLoggingService | Является ключевым Core элементом данного проекта. | Соедержит набор из делегата EncryptedDataEventHandler и события NewMessageOn. Данный подход позволяет решить проблему дублирования кода и сделать код слабосвязным.  Основным методом данного сервиса является асинхронный метод-слушатель Start. Данный метод начинает «прослушивать» заданную в ConfigurationProvider директорию и исходя из происходящих событий отправляет информацию его подписчикам посредстом почты или UDP Multicast |
| ConfigurationProvider | Сервис для работы с конфигурационными данными | Является .NET Standard сборкой, поэтому реализация ConfigurationManager предоставляется NuGet и из коробки поддерживает лишь считывание данных (требуется для обратной совместимости).  Предоставляет набор публичных свойств-геттеров для получения информации о конфигурации приложения, свойствах сервисов и т.п.  Сервис получает ифнормацию с уровня предсталвения (Presentation Layer). |
| EncryptionHelper | Stateless-сервис, используемый в качестве инструмента для преобразования строковых данных в массив байт (для последующей передачи по UDP) и наоборот | Предоставляет публчиные методы Encode и Decode. Входные и выходные данные соответствуют примитиву string и массиву байт. |

Исходя из вышеперечисленного, функциями данного проекта являются глубокая поддержка API-сервисов для журналирования заранее определенных событий, происходящих в рамках ОС Windows и Linux. Вторичной функией данного проекта стоит демонстрация возможностей API посредством клиента для ОС Windows, построенного при помощи актуальной технологии MDIX. Именно демонстрация посредством визуального клиента может углубить понимание целей и функций этого программного продукта.

* 1. Характеристики пользователя

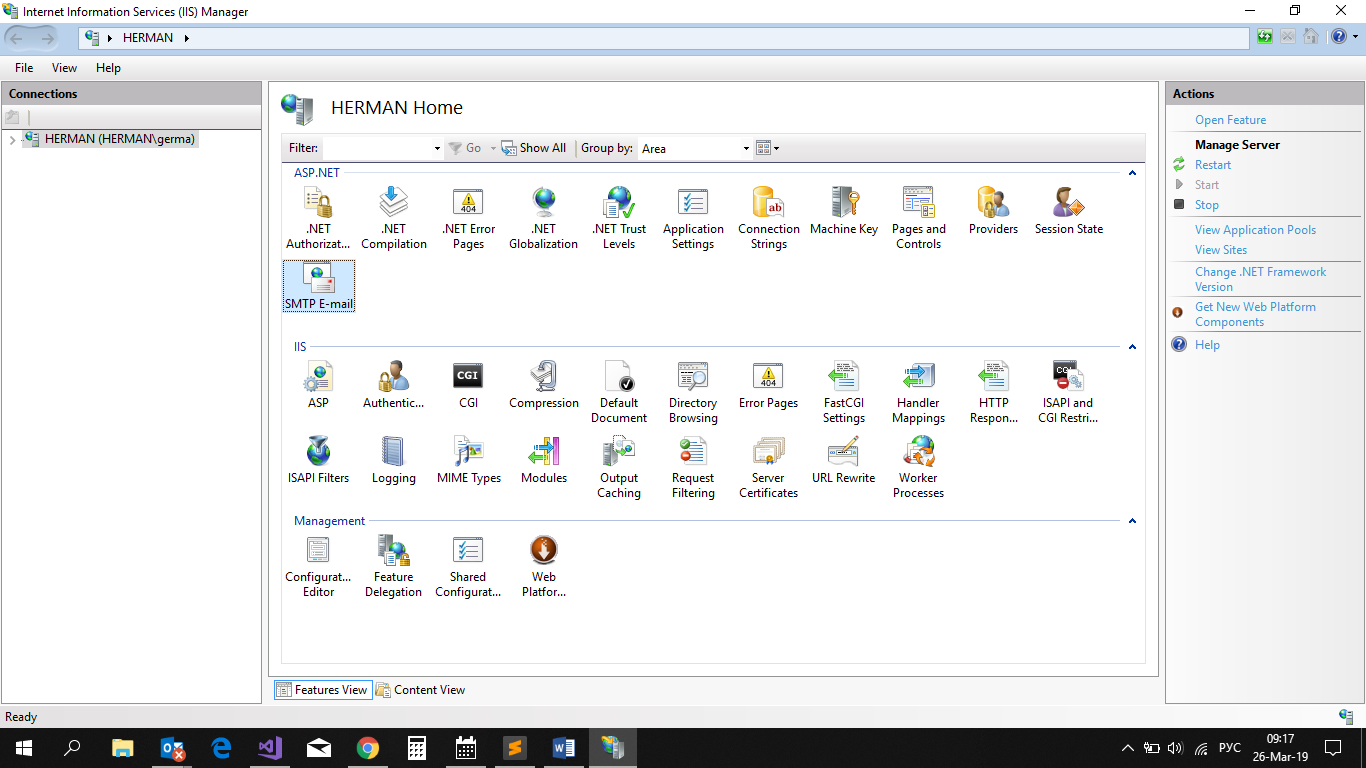
Целевая аудитория данного сервиса – разработчики, системные администраторы, пользователи ПК с большим опытом работы с ОС Windows и Linux.

В особенности следует выделить именно группу разработчиков и администраторов систем, так как данный инструмент позволяет устранять системные ошибки различного уровня на реальных проектах во время разработки, а также для поддержки уже запущенных веб/настольных приложений.

Минимальные требования к пользователю: знание понятий операционная система, процессы в ОС, файловая система, журналирование/логгирование, компьютерные сети; опыт работы с ОС Windows/Linux

Дополнительные требования (для команд разработки и системных администраторов): опыт работы с компьютерными сетями, понимание протокола UDP, опыт работы с почтовыми серверами и локальными сетями.

Затрагивая дополнительные требования, опыт работы с почтовыми серверами может быть разнообразным. Одним из возможных вариантов – использование стандартного (предустановленного) в ОС Windows сервера IIS (Internet Information Services) (отметим, что данный возможности сервера являются доступными после его активации в «Панели управления). Данный сервер представлен здесь как рекомендуемый, так как предоставялет из коробки возможности для работы с почтой. На рис. 2 представлен скриншот «домашней страницы» IIS с выделенным пунктом для работы с SMTP E-mail.

  
Рис. 2. Сервер IIS (Internet Information Services)

* 1. Ограничения

Данный продукт не рекомендуется использовать на системах, отличных от ОС Windows и ОС Ubuntu, т.к. даная библиотека была протестирована именно на данных операционный системах. Однако, отметим, что 2 данные системы покрывают огромную долю рынка ПК.

Данное ПО было протестировано на:

* ОС Windows 10 Home 1803
* ОС Ubuntu 18.04

Также не желательно использование данного ПО пользователями, которые не удовлетворяют требованиям, описанным выше (см. пункт 2.3. Характеристики пользователя).

1. **Детальные требования**

3.1  Требования к внешним интерфейсам

*3.1.1  Интерфейсы пользователя*

В данном продукте предусмотрены 2 основных варианта для взаимодействия с пользователем:

* Windows WPF MDIX Client
* .NET CLI в Windows/Linux (запуск библиотеки SRL API под управлением .NET SDK CLI прямо в терминале)