

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
профессионального образования
**«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И
РАДИОЭЛЕКТРОНИКИ» (ТУСУР)**

Кафедра комплексной информационной безопасности электронно-вычислительных систем
(КИБЭВС)

УТВЕРЖДАЮ
заведующий каф. КИБЭВС
_____ А.А. Шелупанов
«_____» 2014г.

КОМПЬЮТЕРНАЯ ЭКСПЕРТИЗА
Отчет по групповому проектному обучению
Группа КИБЭВС-1401

Ответственный исполнитель
студент гр. 722
_____ О.В. Лобанов
«_____» 2014г.

Научный руководитель
аспирант каф. КИБЭВС
_____ А.И. Гуляев
«_____» 2014г.

РЕФЕРАТ

Курсовая работа содержит 50 страниц, 37 рисунка, 7 таблицы, 7 источников, 1 приложение.

КОМПЬЮТЕРНАЯ ЭКСПЕРТИЗА, ФОРЕНЗИКА, ЛОГИ, QT, XML, GIT, LATEX, ICQ, MS OUTLOOK, WINDOWS, PST, MSG, RTF, HTML, БИБЛИОТЕКИ, РЕПОЗИТОРИЙ, МЕССЕНДЖЕР, ПОЧТОВЫЙ КЛИЕНТ, SQLLITE, РЕЕСТР, ИЗОБРАЖЕНИЯ, READPST, JPEG, PNG.

Цель работы — создание программного комплекса, предназначенного для проведения компьютерной экспертизы.

Задачей, поставленной на данный семестр, стало написание программного комплекса, имеющего следующие возможности:

- 1) сбор и анализ информации из реестра;
- 2) сбор и анализ информации из журналов истории браузеров;
- 3) сбор и анализ информации из мессенджеров;
- 4) сбор и анализ информации из почтовых приложений;
- 5) идентификации файлов изображений по внутреннему содержимому и их проверка;
- 6) сбора информации об установленном ПО по остаточным файлам.

Результаты работы в данном семестре:

- реализован алгоритм извлечения строковых переменных из реестра Windows;
- реализован алгоритм побитового считывания файла формата PST;
- реализован импорт истории (посещений, поисковых запросов, загруженных файлов), закладок и другой информации (версия приложения, логин аккаунта google) из приложения Google Chrome;
- реализован алгоритм парсинга контактного листа пользователя, сохраняемого приложением ICQ;
- реализована проверка конца файла для форматов JPEG и PNG (для идентификации файлов изображений) и проверка заголовков 5 форматов изображений;

Пояснительная записка выполнена при помощи системы компьютерной вёрстки L^AT_EX.

Список исполнителей

Лобанов О.В. – программист, ответственный исполнитель, ответственный за разработку функций сбора информации из реестра.

Мейта М.В. – документатор, программист, ответственный за написание части системы для сбора информации из мессенджера ICQ.

Шиповской В.В. – программист, ответственный за написание части системы для сбора и обработки информации из браузера Google Chrome.

Серяков А.В. – программист, ответственный за написание части системы для сбора информации из почтового клиента MS Outlook.

Боков И.М. – программист, ответственный за написание части системы для идентификации файлов изображений по внутреннему содержимому и их проверки.

Чадайкин М.В. – программист, ответственный за написание части системы для сбора информации об установленном ПО по остаточным файлам.

Содержание

Введение	6
1 Назначение и область применения	6
2 Постановка задачи	6
3 Инструменты	7
3.1 Система контроля версий Git	7
3.2 Система компьютерной вёрстки ТЕХ	7
3.3 Qt - кроссплатформенный инструментарий разработки ПО	8
3.3.1 Автоматизация поиска журнальных файлов	10
3.3.2 Реализация сохранения результатов работы программного комплекса в XML	10
4 Технические характеристики	10
4.1 Требования к аппаратному обеспечению	10
4.2 Требования к программному обеспечению	10
4.3 Выбор единого формата выходных файлов	10
5 Разработка программного обеспечения	11
5.1 Архитектура	11
5.1.1 Основной алгоритм	11
5.1.2 Описание основных функций модуля системы	12
5.2 Сбор информации из браузера Google Chrome	15
5.2.1 Реализация программного модуля	15
5.2.2 Определение форматов файлов	15
5.2.3 Алгоритм работы модуля	19
5.2.4 Структура логов, записанных в формате XML	19
5.2.5 Задачи на следующий семестр	20
5.3 Сбор информации из мессенджера ICQ	25
5.3.1 Общие сведения о программах мгновенного обмена сообщениями	25
5.3.2 Реализация программного модуля	28
5.3.3 Алгоритм работы модуля	29
5.3.4 Задачи на следующий семестр	29
5.4 Сбор информации из почтового клиента MS Outlook	32
5.4.1 Некоторые сведения о почтовых клиентах	32
5.4.2 Реализация программного модуля	32
5.4.3 Исследование файловых форматов	32
5.4.4 Задачи на следующий семестр	34
5.5 Идентификации файлов изображений	38
5.5.1 Реализация программного модуля	38
5.5.2 Алгоритм работы модуля	38
5.5.3 Структура XML-файла	38
5.5.4 Задачи на следующий семестр	40
5.6 Сбор и анализ информации из реестра ОС MS Windows	42
5.6.1 Этап второй — «исследование». Структура и модель формирования	42

5.6.2 Этап второй — «исследование». Исследование готовых решений	43
5.6.3 Этап третий — «разработка». Выбор метода получения информации из реестра	45
5.6.4 Этап третий — «разработка». Разработка приложения	45
5.6.5 Задачи на следующий семестр	45
5.6.6 Ссылки на Интернет-ресурсы	45
Заключение	48
Список использованных источников	49
Приложение А Компакт-диск	50

Введение

Компьютерно-техническая экспертиза – это самостоятельный род судебных экспертиз, относящийся к классу инженерно-технических экспертиз, проводимых в следующих целях: определения статуса объекта как компьютерного средства, выявление и изучение его роли в рассматриваемом деле, а так же получения доступа к информации на электронных носителях с последующим всесторонним её исследованием [1]. Компьютерная экспертиза помогает получить доказательственную информацию и установить факты, имеющие значение для уголовных, гражданских и административных дел, сопряжённых с использованием компьютерных технологий. Для проведения компьютерных экспертиз необходима высокая квалификация экспертов, так как при изучении представленных носителей информации, попытке к ним доступа и сбора информации возможно внесение в информационную среду изменений или полная утрата важных данных.

Компьютерная экспертиза, в отличие от компьютерно-технической экспертизы, затрагивает только информационную составляющую, в то время как аппаратная часть и её связь с программной средой не рассматривается.

На протяжении предыдущих семестров нами были рассмотрены такие направления компьютерной экспертизы, как исследование файловых систем, сетевых протоколов, организация работы серверных систем, механизм журналирования событий. Также нами были изучены основные задачи, которые ставятся перед сотрудниками правоохранительных органов, которые проводят компьютерную экспертизу, и набор существующих утилит, способных помочь эксперту в проведении компьютерной экспертизы. Было выявлено, что существует множество разрозненных программ, предназначенных для просмотра лог-файлов системы и таких приложений, как мессенджеры и браузеры, но для каждого вида лог-файлов необходимо искать отдельную программу. Так как ни одна из них не позволяет эксперту собрать воедино и просмотреть все логи системы, браузеров и мессенджеров, было решено создать для этой цели собственный автоматизированный комплекс, которому на данный момент нет аналогов.

1 Назначение и область применения

Разрабатываемый комплекс предназначен для автоматизации процесса сбора информации с исследуемого образа жёсткого диска.

2 Постановка задачи

На данный семестр были поставлены следующие задачи:

- изучение проекта «Компьютерная экспертиза»;
- изучение теоретического материала и основных инструментов разработки;
- определение индивидуальных задач для каждого участника проектной группы;
- исследование предметных областей в рамках индивидуальных задач;
- реализация нескольких программных модулей.

Задачи по проектированию модулей:

- 1) сбор и анализ информации из браузера Google Chrome;
- 2) сбор и анализ информации из реестра Windows;

- 3) сбор и анализ информации из мессенджера ICQ;
- 4) сбор и анализ информации из почтового клиента MS Outlook;
- 5) идентификации файлов изображений по внутреннему содержимому и их проверка;
- 6) сбора информации об установленном ПО по остаточным файлам.

3 Инструменты

3.1 Система контроля версий Git

Для разработки программного комплекса для проведения компьютерной экспертизы решено использовать Git.

Git — распределённая система управления версиями файлов. Проект был создан Линусом Торвальдсом для управления разработкой ядра Linux, как противоположность системе управления версиями Subversion (также известная как «SVN») [2].

Необходимость использования системы версий, очевидна. Так как в группе несколько программистов и тестер, мы имеем:

- возможность удаленной работы с исходными кодами;
- возможность создавать свои ветки, не мешая при этом другим разработчикам;
- доступ к последним изменениям в коде, т.к. все исходники хранятся на сервере git.keva.su;
- исходные коды защищены, доступ к ним можно получить лишь имея RSA-ключ;
- возможность откатиться к любой стабильной стадии проекта.

Основные постулаты работы с кодом в системе Git:

- каждая задача решается в своей ветке;
- коммитим сразу, как что-то получили осмысленное;
- в master мерджится не разработчиком, а вторым человеком, который производит вычитку и тестирование изменения;
- все коммиты должны быть осмысленно подписаны/прокомментированы.

Для работы над проектом нами был поднят собственный репозиторий на сервере git.keva.su.

Адреса репозиториев следующие:

Исходные файлы проекта:

```
git clone git@git.keva.su:gpo.git gpo.git
```

Репозиторий для тестирования проекта:

```
git clone git@git.keva.su:gpo-testdata.git gpo-testdata.git
```

3.2 Система компьютерной вёрстки TeX

TeX — это созданная американским математиком и программистом Дональдом Кнутом система для вёрстки текстов. Сам по себе TeX представляет собой специализированный язык программирования. Каждая издательская система представляет собой пакет макроопределений этого языка.

LATeX — это созданная Лэсли Лэмпартом издательская система на базе TeX'a[3]. LATEX позволяет пользователю сконцентрировать свои усилия на содержании и структуре текста, не заботясь о деталях его оформления.

Для подготовки отчётной и иной документации нами был выбран LATEX так как совместно с системой контроля версий Git он предоставляет возможность совместного создания и редактирова-

ния документов. Огромным достоинством системы L^AT_EX то, что создаваемые с её помощью файлы обладают высокой степенью переносимости [4].

Совместно с L^AT_EX часто используется BibT_EX — программное обеспечение для создания форматированных списков библиографии. Оно входит в состав дистрибутива L^AT_EX и позволяет создавать удобную, универсальную и долговечную библиографию. BibT_EX стал одной из причин, по которой нами был выбран L^AT_EX для создания документации.

3.3 Qt - кроссплатформенный инструментарий разработки ПО

Qt - это кроссплатформенная библиотека C++ классов для создания графических пользовательских интерфейсов (GUI) от фирмы Digia. Эта библиотека полностью объектно-ориентированная, что обеспечивает легкое расширение возможностей и создание новых компонентов. Ко всему прочему, она поддерживает огромнейшее количество платформ.

Qt позволяет запускать написанное с его помощью ПО в большинстве современных операционных систем путём простой компиляции программы для каждой ОС без изменения исходного кода. Включает в себя все основные классы, которые могут потребоваться при разработке прикладного программного обеспечения, начиная от элементов графического интерфейса и заканчивая классами для работы с сетью, базами данных и XML. Qt является полностью объектно-ориентированным, легко расширяемым и поддерживающим технику компонентного программирования.

Список использованных классов фреймворка QT

- iostream
- QChar
- QCryptographicHash
- QDateTime
- QDir
- QDirIterator
- QFile
- QFileInfo
- QIODevice
- QList
- QRegExp
- QString
- QTextStream
- QSql/QSqlDatabase
- QVector
- QMap
- QXmlStreamReader
- QXmlStreamWriter
- Conversations

Класс QXmlStreamWriter представляет собой XML писателя с простым потоковым.

Класс QXmlStreamReader представляет собой быстрый синтаксически корректный XML анализатор с простым потоковым API.

QVector представляет собой класс для создания динамических массивов.

Модуль QSql/QSqlDatabase помогает обеспечить однородную интеграцию БД в ваши Qt приложения.

Класс QTextStream предоставляет удобный интерфейс для чтения и записи текста.

QTextStream может взаимодействовать с QIODevice, QByteArray или QString. Используя потоковые операторы QTextStream, вы можете легко читать и записывать слова, строки и числа. При формировании текста QTextStream поддерживает параметры форматирования для заполнения и выравнивания полей и форматирования чисел. [5]

Класс QString предоставляет строку символов Unicode.

Класс QMap — контейнерный класс для хранения элементов различных типов данных.

Класс QDateTime используется для работы с форматом даты, в который записывается информация о файле.

QString хранит строку 16-битных QChar, где каждому QChar соответствует один символ Unicode 4.0. (Символы Unicode со значениями кодов больше 65535 хранятся с использованием суррогатных пар, т.е. двух последовательных QChar.)

Unicode - это международный стандарт, который поддерживает большинство использующихся сегодня систем письменности. Это расширение US-ASCII (ANSI X3.4-1986) и Latin-1 (ISO 8859-1), где все символы US-ASCII/Latin-1 доступны на позициях с тем же кодом.

Внутри QString использует неявное совместное использование данных (копирование-при записи), чтобы уменьшить использование памяти и избежать ненужного копирования данных. Это также позволяет снизить накладные расходы, свойственные хранению 16-битных символов вместо 8-битных.

В дополнение к QString Qt также предоставляет класс QByteArray для хранения сырых байт и традиционных нультерминальных строк. В большинстве случаев QString - необходимый для использования класс. Он используется во всем API Qt, а поддержка Unicode гарантирует, что ваши приложения можно будет легко перевести на другой язык, если в какой-то момент вы захотите увеличить их рынок распространения. Два основных случая, когда уместно использование QByteArray: когда вам необходимо хранить сырье двоичные данные и когда критично использование памяти (например, в Qt для встраиваемых Linux-систем).[6]

Класс QRegExp предоставляет сопоставление с образцом при помощи регулярных выражений.

Регулярное выражение, или "regexp", представляет собой образец для поиска соответствующей подстроки в тексте. Это полезно во многих ситуациях, например:

Проверка правильности – регулярное выражение может проверить, соответствует ли подстрока каким-либо критериям, например, целое ли она число или не содержит ли пробелов. Поиск – регулярное выражение предоставляет более мощные шаблоны, чем простое соответствие строки, например, соответствие одному из слов mail, letter или correspondence, но не словам email, mailman, mailer, letterbox и т.д. Поиск и замена – регулярное выражение может заменить все вхождения подстроки другой подстрокой, например, заменить все вхождения & на & , исключая случаи, когда за & уже следует amp;. Разделение строки – регулярное выражение может быть использовано для определения того, где строка должна быть разделена на части, например, разделяя строку по символам табуляции.

QFileInfo - Во время поиска возвращает полную информацию о файле.

Класс QDir обеспечивает доступ к структуре каталогов и их содержимого.

QIODevice представляет собой базовый класс всех устройств ввода/вывода в Qt.

Класс QCryptographicHash предоставляет способ генерации криптографических хэшей. QCryptographicHash могут быть использованы для генерации криптографических хэшей двоичных или текстовых данных. В настоящее время MD4, MD5, и SHA-1 поддерживаются.[6]

QChar обеспечивает поддержку 16-битных символов Unicode.

3.3.1 Автоматизация поиска журнальных файлов

Для сканирования образа на наличие интересующих лог файлов использовался класс QDirIterator. После вызова происходит поочередный обход по каждому файлу в директории и поддиректории. Проверка полученного полного пути к файлу осуществляется регулярным выражением, если условие выполняется, происходит добавление в список обрабатываемых файлов.

3.3.2 Реализация сохранения результатов работы программного комплекса в XML

Сохранение полученных данных происходит в ранее выбранный формат XML(Extensible Markup Language). Для этого используется класс QXmlStreamReader и QxmlStreamWriter. Класс QXmlStreamWriter представляет XML писателя с простым потоковым API.

QXmlStreamWriter работает в связке с QXmlStreamReader для записи XML. Как и связанный класс, он работает с QIODevice, определённым с помощью setDevice().

Сохранение данных реализовано в классе WriteMessage. В методе WriteMessages, структура которого представлена на UML диаграмме в разделе Архитектура.

4 Технические характеристики

4.1 Требования к аппаратному обеспечению

Минимальные системные требования:

- процессор 1ГГц Pentium 4;
- оперативная память 512 Мб;
- место на жёстком диске – 9 Гб.

4.2 Требования к программному обеспечению

Для корректной работы разрабатываемого программного комплекса на компьютере должна быть установлена операционная система Debian Squeeze или выше, данная система должна иметь набор библиотек QT.

4.3 Выбор единого формата выходных файлов

Для вывода результата был выбран формат XML-документов, так как с данным форматом легко работать при помощи программ, а результат работы данного комплекса в дальнейшем планируется обрабатывать при помощи программ.

XML - eXtensible Markup Language или расширяемый язык разметки. Язык XML представляет собой простой и гибкий текстовый формат, подходящий в качестве основы для создания новых языков разметки, которые могут использоваться в публикации документов и обмене данными [7]. Задумка языка в том, что он позволяет дополнять данные метаданными, которые разделяют документ на объекты с атрибутами. Это позволяет упростить программную обработку документов, так как структурирует информацию.

Простейший XML-документ может выглядеть так:

```
\noindent <?xml version="1.0"?> \\
<list\_of\_items> \\
<item id="1"><first/> \\
<item id="2"> \\
<item id="3"> \\
<item id="4"><last/> \\
</list\_of\_items>
```

Первая строка - это объявление начала XML-документа, дальше идут элементы документа `<list_of_items>` - тег описывающий начало элемента `list_of_items`, `</list_of_items>` - тег конца элемента. Между этими тегами заключается описание элемента, которое может содержать текстовую информацию или другие элементы (как в нашем примере). Внутри тега начала элемента так же могут указывать атрибуты элемента, как например атрибут `id` элемента `item`, атрибуту должно быть присвоено определенное значение.

5 Разработка программного обеспечения

5.1 Архитектура

5.1.1 Основной алгоритм

В ходе разработки был применен видоизменённый шаблон проектирования Factory method.

Данный шаблон относится к классу порождающих шаблонов. Шаблоны данного класса - это шаблоны проектирования, которые абстрагируют процесс инстанцирования (создания экземпляра класса). Они позволяют сделать систему независимой от способа создания, композиции и представления объектов. Шаблон, порождающий классы, использует наследование, чтобы изменять инстанцируемый класс, а шаблон, порождающий объекты, делегирует инстанцирование другому объекту. Пример организации проекта при использовании шаблона проектирования Factory method представлен на рисунке 5.1.

Использование данного шаблона позволило нам разбить наш проект на независимые модули, что весьма упростило задачу разработки, так как написание алгоритма для конкретного таска не влияло на остальную часть проекта. При разработке был реализован базовый класс для работы с образом диска. Данный класс предназначался для формирования списка настроек, определения операционной системы на смонтированном образе и инстанцировании и накапливание всех необходимых классов-тасков в очереди тасков. После чего каждый таск из очереди отправлялся на выполнение. Блоксхема работы алгоритма представлена на рисунке 5.2.

Каждый класс-таск порождался путем наследования от базового абстрактного класса кото-

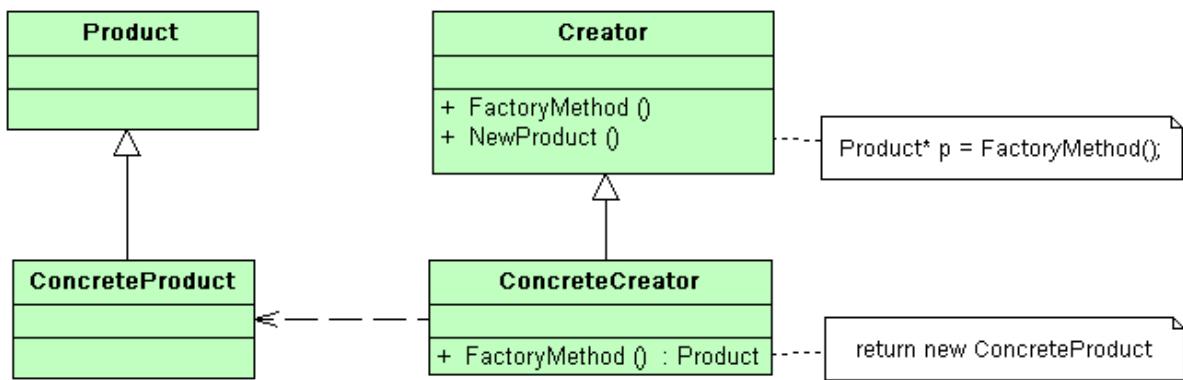


Рисунок 5.1 – Пример организации проекта при использовании шаблона проектирования Factory method

рый имеет 8 методов и 3 атрибута:

- 1) `QString manual()` - возвращает справку о входных параметрах данного таска;
- 2) `void setOption(QStringList list)` - установка флагов для поданных на вход параметров;
- 3) `QString command()` - возвращает команду для инициализации таска вручную;
- 4) `bool supportOS(const coex::typeOS &os)` - возвращает флаг, указывающий на возможность использования данного таска для конкретной операционной системы;
- 5) `QString name()` - возвращает имя данного таска;
- 6) `QString description()` - возвращает краткое описание таска;
- 7) `bool test()` - предназначена для теста на доступность таска;
- 8) `bool execute(const coex::config &config)` - запуск таска на выполнение;
- 9) `QString m_strName` - хранит имя таска;
- 10) `QString m_strDescription` - хранит описание таска;
- 11) `bool m_bDebug` - флаг для параметра `-debug`;

На данный момент в проекте используется восемь классов. UML-диаграмма классов представлена на рисунке 5.3.

Классы `taskSearchSyslogsWin`, `taskSearchPidginWin` и `taskSearchSkypeWin` - наследники от класса `task` являются тасками. Класс `winEventLog` и `_EVENTLOGRECORD` предназначены для конвертации журнальных файлов операционной системы Windows XP, а класс `writerMessages` для преобразования истории переписки.

5.1.2 Описание основных функций модуля системы

Любой модуль системы является классом-наследником от некоторого абстрактного класса используемого как основу для всех модулей программы (шаблон проектирования Factory method). Модуль содержит в себе 8 методов и 3 атрибута:

`QString manual()` - возвращает справку о входных параметрах данного таска

`void setOption(QStringList list)` - установка флагов для поданных на вход параметров

`QString command()` - возвращает команду для инициализации таска вручную

`bool supportOS(const coex::typeOS &os)` - возвращает флаг указывающий на возможность использования данного таска для конкретной операционной системы

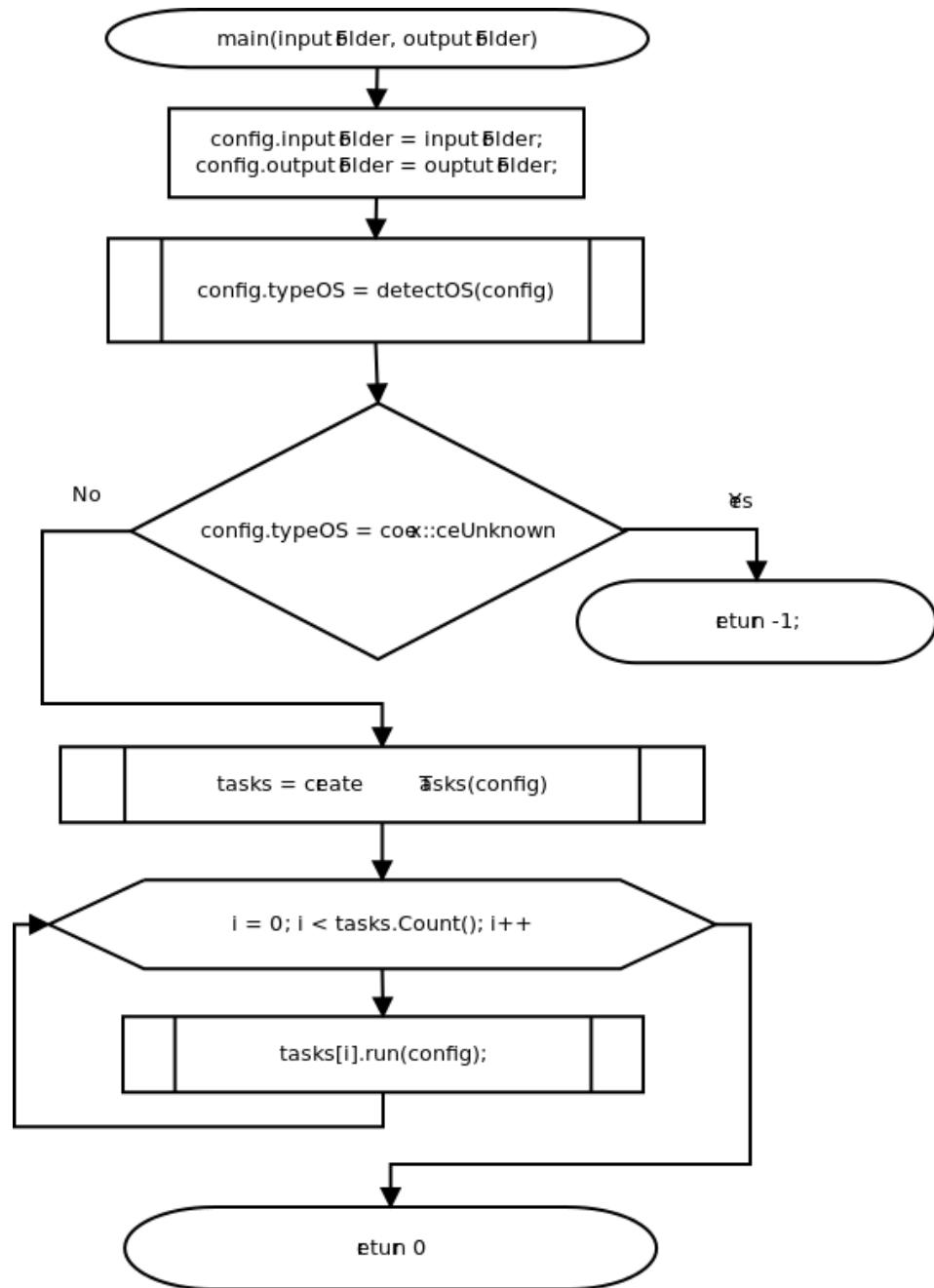


Рисунок 5.2 – Алгоритм работы с образом диска

`QString name()` - возвращает имя данного таска

`QString description()` - возвращает краткое описание таска

`bool test()` - предназначена для проверки работоспособности таска

`bool execute(const coex::config &config)` - запуск таска на выполнение

`QString m_strName` - хранит имя таска

`QString m_strDescription` - хранит описание таска

`bool m_bDebug` - флаг для параметра `-debug`

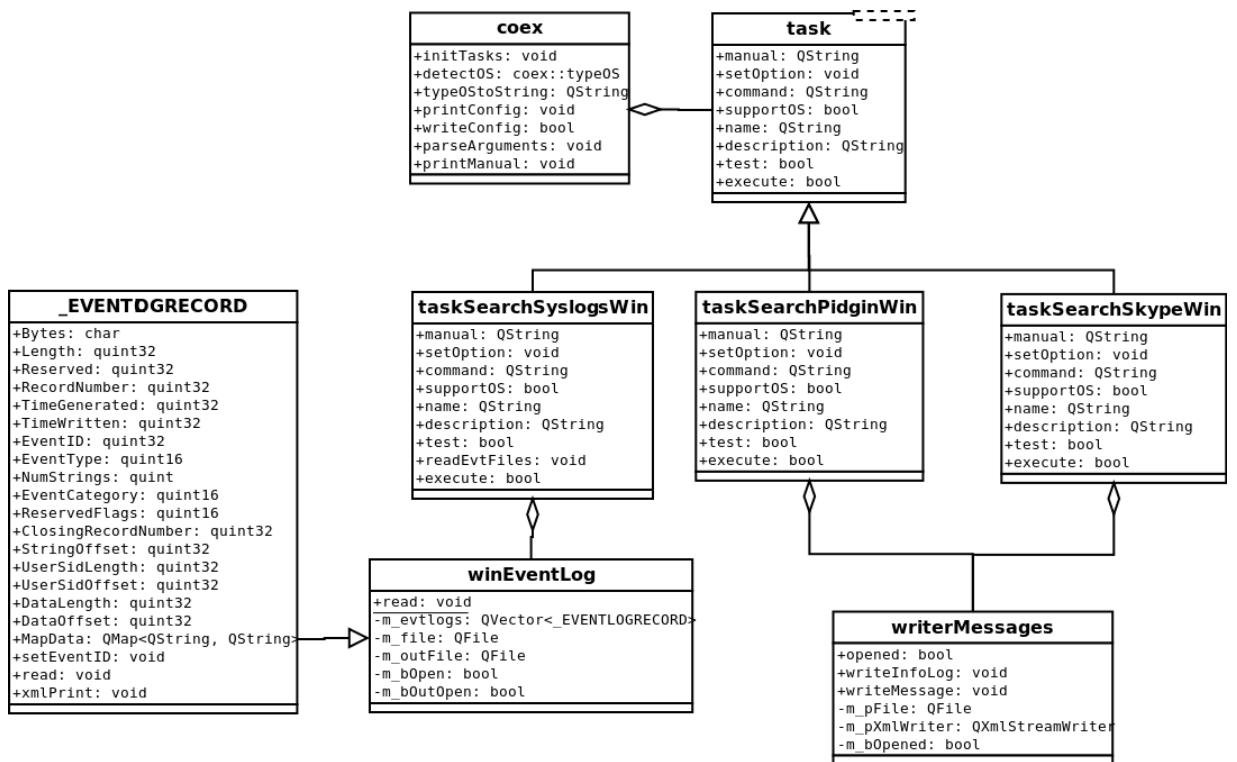


Рисунок 5.3 – UML-диаграмма классов

5.2 Сбор информации из браузера Google Chrome

Целью данной работы стало исследование лог-файлов и написание программного модуля для сбора пользовательских данных (закладки, история посещений, история загруженных файлов, поисковые запросы) приложения Google Chrome и представления их в формате XML.

5.2.1 Реализация программного модуля

Реализация программного модуля включала в себя следующие шаги:

- 1) Изучение проекта соех;
- 2) Изучение особенностей работы с библиотеками QT;
- 3) Изучение особенностей работы с XML-форматом (языка разметки);
- 4) Изучение системы компьютерной верстки Latex для написания документации;
- 5) Изучение системы распределенного контроля версий Git и ее основных возможностей;
- 6) Исследование приложения Google Chrome и директории хранения логов;
- 7) Определение форматов файлов, хранимых программой Google Chrome;
- 8) Разработка программного модуля.

В ходе исследования приложения Google Chrome было обнаружено, что данный браузер хранит пользовательские данные локально на машинах пользователей. По умолчанию эти данные хранятся в директории, представленной в таблице 5.2, интересующие нас файлы — в таблице 5.1.

Таблица 5.1 – Интересующие нас файлы

Bookmarks	Закладки
History	История посещений, история запросов, история загруженных файлов
Preferences	Настройки (директория загрузки файлов, версия программы, логин аккаунта Google)

Таблица 5.2 – Директории хранения логов Chrome

Операционная система	Директория
Linux Mint(Ubuntu)	/home/имя_пользователя/.config/google-chrome/Default/
Win7	C:\Users\имя_пользователя\AppData\Local\Google\Chrome\User Data\Default\

5.2.2 Определение форматов файлов

Раннее найденные файлы не имеют расширения, но если открыть History, Bookmarks и Preferences текстовым редактором, то можно понять, что Bookmarks и Preferences имеют формат JSON. JSON (JavaScript Object Notation) — текстовый формат обмена данными, основанный на JavaScript и обычно используемый именно с этим языком. Как и многие другие текстовые форматы, JSON легко читается людьми. History — это реляционная база данных основанная на СУБД SQLite.

Результат открытия файла History представлен на рисунке 5.4.

```

History — Блокнот
Файл Правка Формат Вид Справка
SQLITE format 3 + @ € X
съеъиH,о,%да•То--`•

```

Рисунок 5.4 – Файл History

Структура файла Bookmarks приведена на рисунке 5.5, файла Preferences — на рисунке 5.6.

```

1  {
2      "checksum": "2a1a55626afe67f31ca3693710c0a54a",
3      "roots": {
4          "bookmark_bar": {
5              "children": [ {
6                  "date_added": "13053208197956328",
7                  "id": "75",
8                  "meta_info": {
9                      "stars.id": "ssc_8aceafc1a9c1fd4f",
10                     "stars.isSynced": "true"
11                 },
12                 "name": "Яндекс",
13                 "sync_transaction_version": "54",
14                 "type": "url",
15                 "url": "http://www.yandex.ru/"
16             },
17             "date_added": "13051623335593501",
18             "id": "76",
19             "meta_info": {
20                 "stars.id": "ssc_5089e09b4154a82e",
21                 "stars.isSynced": "true",
22                 "stars.pageData": "Ig5sS3cyY3I4RmdPeWROTQ==",
23                 "stars.type": "2"
24             },
25             "name": "Новости",
26             "sync_transaction_version": "83",
27             "type": "url",
28             "url": "https://vk.com/feed"
29         ]
30     }
31 }
    
```

Рисунок 5.5 – Структура файла Bookmarks

Необходимо рассмотреть структуру файла History .sqlite. БД содержит 9 таблиц:

- 1) downloads
- 2) downloads_url_chains

```

1  {
2      "autofill": {
3          "enabled": false
4      },
5      "bookmark_bar": {
6          "show_apps_shortcut": true,
7          "show_on_all_tabs": true
8      },
9      "browser": {
10         "clear_data": {
11             "cache": false,
12             "cookies": false,
13             "form_data": false,
14             "hosted_apps_data": false,
15             "passwords": false,
16             "time_period": 4
17         },
18         "clear_lso_data_enabled": true,
19         "enable_spellchecking": true,
20         "last_known_google_url": "https://www.google.ru/",
21         "last_prompted_google_url": "https://www.google.ru/",
22         "pepper_flash_settings_enabled": true,
23         "window_placement": {
24             "bottom": 1033,
25             "left": 280,
26             "maximized": false,
27             "right": 1330,
28             "top": 69,
29             "work_area_bottom": 1040,
30             "work_area_left": 0,
31             "work_area_right": 1920,
32             "work_area_top": 0
33         }
34     }
}.

```

Рисунок 5.6 – Структура файла Preferences

- 3) keyword_search_terms
- 4) meta
- 5) segment_usage
- 6) segments
- 7) urls
- 8) visit_source
- 9) visits

Таблицы которые были рассмотрены на данный момент:

- 1) downloads
- 2) downloads_url_chains
- 3) keyword_search_terms
- 4) urls

Таблица downloads содержит информацию о загруженных файлах (путь, время начала загрузки, размер файла, тип файла), но не содержит адрес, откуда был загружен данный файл. Необходимый адрес находится в таблице downloads_url_chains, которая связана с таблицей downloads по полю id. SQL запрос для выборки информации о истории загруженных файлов: SELECT downloads.target_path, downloads.referrer, downloads.start_time, downloads.received_bytes, downloads_url_chains.url FROM downloads, downloads_url_chains WHERE downloads.id=downloads_url_chains.id.

Пример таблицы downloads можно увидеть на рисунке 5.7.

id	current_path	target_path	start_time	received_bytes	total_bytes	state	dangerous	interrupted	end_time	opened	referrer
1	C:\Users\blacksc\Desktop\	C:\Users\blacksc\Desktop\	13057012671412926	100121965	100121965	1	4	0	13057012702020926	0	http://www.asus.com
2	C:\Users\blacksc\Desktop\	C:\Users\blacksc\Desktop\	13057012844385018	5788258	5788258	1	4	0	13057012860894018	0	http://www.asus.com
3	C:\Users\blacksc\Desktop\	C:\Users\blacksc\Desktop\	13057012986784500	5467407	5467407	1	4	0	13057013024087014	0	http://www.asus.com
4	C:\Users\blacksc\Desktop\	C:\Users\blacksc\Desktop\	13057013018772014	23811254	23811254	1	4	0	13057013035650014	0	http://www.asus.com
5	C:\Users\blacksc\Desktop\	C:\Users\blacksc\Desktop\	13057060535907300	1142392	1142392	1	4	0	13057060548920300	1	http://store.steampowered.com
6	C:\Users\blacksc\Desktop\	C:\Users\blacksc\Desktop\	13057061670894320	336677034	336677034	1	4	0	13057061955917084	1	http://miui.su/forum
7	C:\Users\blacksc\Desktop\	C:\Users\blacksc\Desktop\	13057073181103344	775168	775168	1	4	0	13057073185059344	1	http://e5.onthehub.ru
10	C:\Users\blacksc\Desktop\	C:\Users\blacksc\Desktop\	13057073555650852	1677920	1677920	1	4	0	13057073561215852	1	http://www.skype.com
11	C:\Users\blacksc\Desktop\	C:\Users\blacksc\Desktop\	13057075072297792	6629960	6629960	1	4	0	13057075112352224	1	http://www.teamviewer.com
15	C:\Users\blacksc\Desktop\	C:\Users\blacksc\Desktop\	13057072039659752	16145	16145	1	4	0	13057072040355752	1	http://otherreferat.com
16	C:\Users\blacksc\Desktop\	C:\Users\blacksc\Desktop\	13057072161695616	16145	16145	1	4	0	13057072163098616	0	http://otherreferat.com
17	C:\Users\blacksc\Desktop\	C:\Users\blacksc\Desktop\	13057072429856542	541059	541059	1	4	0	13057072431480542	0	http://otherreferat.com
30	C:\Users\blacksc\Desktop\	C:\Users\blacksc\Desktop\	13057082196817424	100121965	100121965	1	4	0	13057082267706856	0	http://www.asus.com
31	C:\Users\blacksc\Desktop\	C:\Users\blacksc\Desktop\	13057082209387856	8594778	8594778	1	4	0	1305708222116856	0	http://www.asus.com
32	C:\Users\blacksc\Desktop\	C:\Users\blacksc\Desktop\	13057082219281856	5108502	5108502	1	4	0	1305708224092856	0	http://www.asus.com
33	C:\Users\blacksc\Desktop\	C:\Users\blacksc\Desktop\	13057082396423152	100121965	100121965	1	4	0	13057082445647152	1	http://www.asus.com

Рисунок 5.7 – Таблица downloads

Таблица keyword_search_terms содержит информацию о поисковых запросах (рис. 5.8). SQL запрос: SELECT DISTINCT term FROM keyword_search_terms.

keyword_id	url_id	lower_term	term
2	11	google	google
2	14	asus zenbook ux32vd	asus zenbook ux32vd
2	114	ыеуфь	ыеуфь
2	126	гора хроттару	гора Хроттару
2	129	праховы скалы	праховы скалы
2	147	miui nexus 5 торрент	miui nexus 5 торрент
2	148	miui nexus 5	miui nexus 5
2	184	skype	skype
2	189	teamviewer	teamviewer
2	220	пользуюсь полиномом лагранжа 5-й степени, найти значения функции в точке	Пользуюсь полиномом Лагранжа 5-й степени, найти значения функции в точке
2	221	пользуюсь полиномом лагранжа 5-й степени, найти значения	Пользуюсь полиномом Лагранжа 5-й степени, найти значения

Рисунок 5.8 – Таблица keyword_search_terms

Таблица urls содержит информацию о посещенных ресурсах (адрес, заголовок, время последнего посещения). SQL запрос: SELECT url,title,last_visit_time FROM urls. Данная таблица приведена на рисунке 5.9.

id	url	title	visit_count	typed_count	last_visit_time	hidden	favicon_id
1	https://www.google.ru/intl/... Браузер Chrome		0	0	1305701231200000	1	0
2	https://dl.google.com/upd...		0	0	13057012304000000	1	0
3	http://www.bing.com/searc... google chrome - Bing		0	0	13057012292000000	1	0
4	http://www.bing.com/searc... RSS		0	0	13057012292000000	1	0
5	file:///E:/Bin/HomeTab/EN... E:\Bin\HomeTab\ENG\nis.html		0	0	13057012144000000	1	0
6	https://www.google.ru/intl/... Браузер Chrome		0	0	13057012298000000	1	0
7	http://tools.google.com/ch... Начало работы		1	0	13057012337524409	0	0
8	https://www.google.com/i... Начало работы		1	0	13057012337524409	0	0
9	https://www.google.ru/chr... Начало работы		1	0	13057012337524409	0	0
10	https://www.google.ru/web... Google		13	0	13057489600619987	0	0
11	https://www.google.ru/web... google - Поиск в Google		2	0	13057013065620014	0	0
12	https://www.google.ru/ur...		1	0	13057012602158135	0	0

Рисунок 5.9 – Таблица urls

Ниже, на рисунке 5.10, приведена таблица downloads_url_chains.

Как можно увидеть из рисунка 5.5, структура файла Bookmarks такова: имеется заголовок «roots» — корень, после которого следует подзаголовок «bookmark_bar», в теле которого находятся все закладки, расположенные на панели закладок. Тело разбито на блоки, в которых находится такая информация, как дата добавления, адрес ресурса, заголовок страницы. Также присутствует подзаголовок «other», в теле которого находятся все остальные закладки.

id	chain_index	url
1	0	http://dlcdn.asus.com/pub/ASUS/nb/DriversForWin8/WiFi/Wifi_Intel_Win7_64_VER15704.zip
2	0	http://dlcdn.asus.com/pub/ASUS/nb/X75A/LAN_Atheros_Win7_64_Z201516.zip
3	0	http://dlcdn.asus.com/pub/ASUS/nb/Drivers/USB3.0/USB3_Intel_Win7_64_Z105235.zip
4	0	http://dlcdn.asus.com/pub/ASUS/nb/DriversForWin8/SmartGesture/SmartGesture_Win7_64_VER201.zip
5	0	http://media.steampowered.com/client/installer/SteamSetup.exe
6	0	http://miui.su/firmware_manager/firmware/download/9440/
6	1	http://ota.miui.su/roms/google_nexus_5/miuisu_v4.4.2_hammerhead_4.9.26.zip
7	0	http://e5.onthehub.com/Static/Installers/SDM_EN.msi
8	0	http://e5.onthehub.com/WebStore/Account/SdxRequestHandler.ashx?on=100244978398&uid=3acf2d2e-8a0b-e211-bd05-f04da23e67f6&oid=e5e33df2-1e54
9	0	http://dl.rutracker.org/forum/dl.php?t=4572044
10	0	http://www.skype.com/go/getskype
10	1	http://download.skype.com/a22041668cd904272aeed6da1d43a7a0/SkypeSetup.exe

Рисунок 5.10 – Таблица downloads_url_chains

В файле Preferences хранится вся возможная информация о настройке браузера. Необходимая информация (версия, директория загрузки и сохранённый логин) была найдена простым поиском слов (chrome_version, default_directory_download, username) в файле.

5.2.3 Алгоритм работы модуля

Блок-схема алгоритма парсинга файла Bookmarks представлена на рисунке 5.11.

Извлечение подстроки из строки осуществляется с помощью регулярного выражения: `\".*\".*\"(.*)\"`. Например есть строка "name": "Яндекс данное регулярное выражение возвращает 2 подстроки name и Яндекс. Из строки "url": "http://www.yandex.ru/" будет найдено url и http://www.yandex.ru/. Нам необходимы только подстроки с названием страницы и адресом.

Блок-схема алгоритма парсинга файла Preferences представлена на рисунке 5.12.

Блок-схема алгоритма выборки данных из БД History представлена на рисунке 5.13.

5.2.4 Структура логов, записанных в формате XML

Документы XML имеют иерархическую структуру и начинаются с пролога, который указывает, что документ написан на XML, а также указывает версию XML. Следующий элемент `<add>` — это начальный элемент документа (корень). Далее будут записаны n дочерних элементов `<doc>` где n — количество объектов, в теле которых будет записано нужное нам количество полей для записи информации. В последнюю очередь записывается конечный элемент `</add>`.

Пример файла bookmarks.XML приведен на рисунке 5.14.

Значение поля id — это сгенерированный хэш MD5 от строки bookmark_url + «Chrome», application — наименование приложения, doc_type — тип информации (рис. 5.14).

Значение поля id — это сгенерированный хэш MD5 от строки preferences_param_value + «Chrome» (рис. 5.15).

Значение поля id — это сгенерированный хэш MD5 от строки history_name + history_url + history_date + «Chrome» (рис. 5.16).

Значение поля id — это сгенерированный хэш MD5 от строки «download» + dowload_url + dowload_start_time + «Chrome» (рис. 5.17). Значение download_size записывается в байтах.

Значение поля id — это сгенерированный хэш MD5 от строки keyword_term + «Chrome» (рис. 5.18).

На данный момент реализован импорт истории (посещений, поисковых запросов, загруженных файлов), закладок и другой информации (версия приложения, логин аккаунта google) из приложения

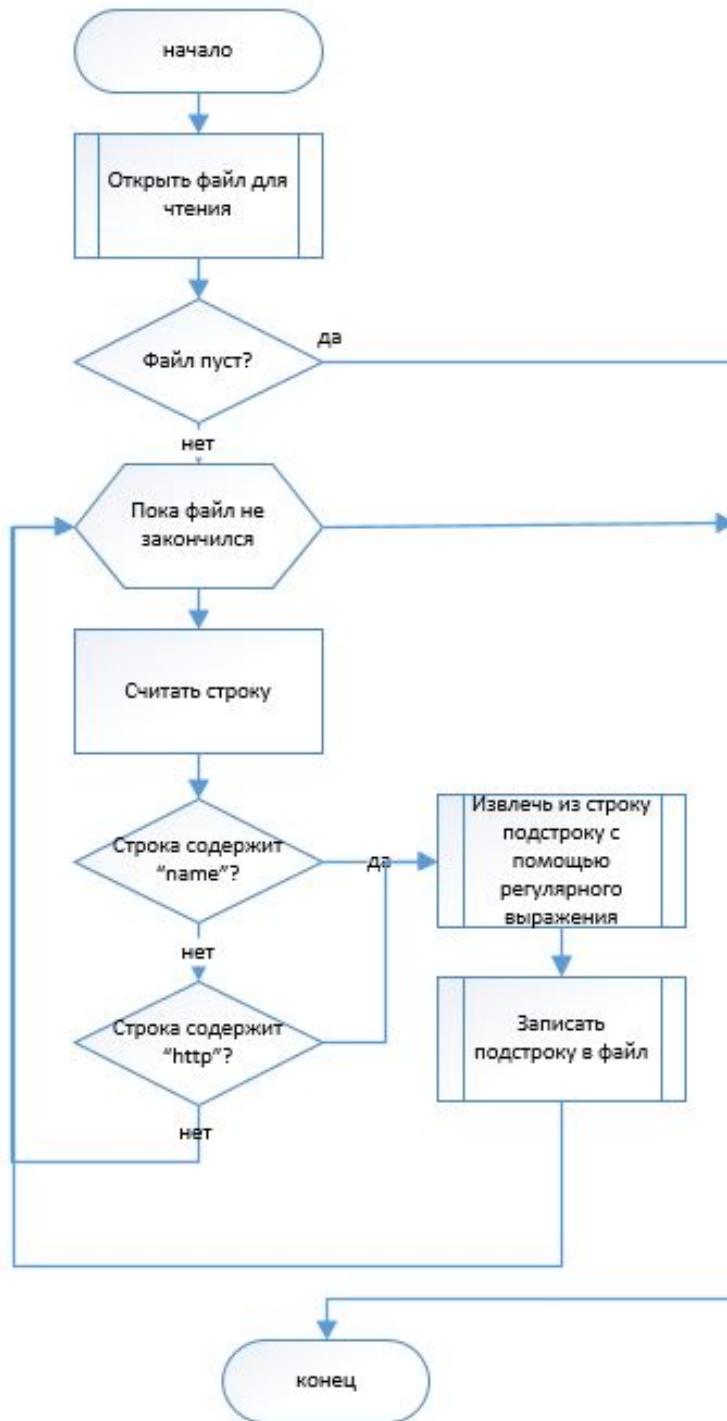


Рисунок 5.11 – Блок-схема алгоритма парсинга файла Bookmarks

Google Chrome.

5.2.5 Задачи на следующий семестр

Задачи на следующий семестр:

- 1) нормализовать формат даты;
- 2) реализовать рекурсивный обход файловой системы для нахождения необходимых файлов;
- 3) поиск другой информации приложения Google Chrome.

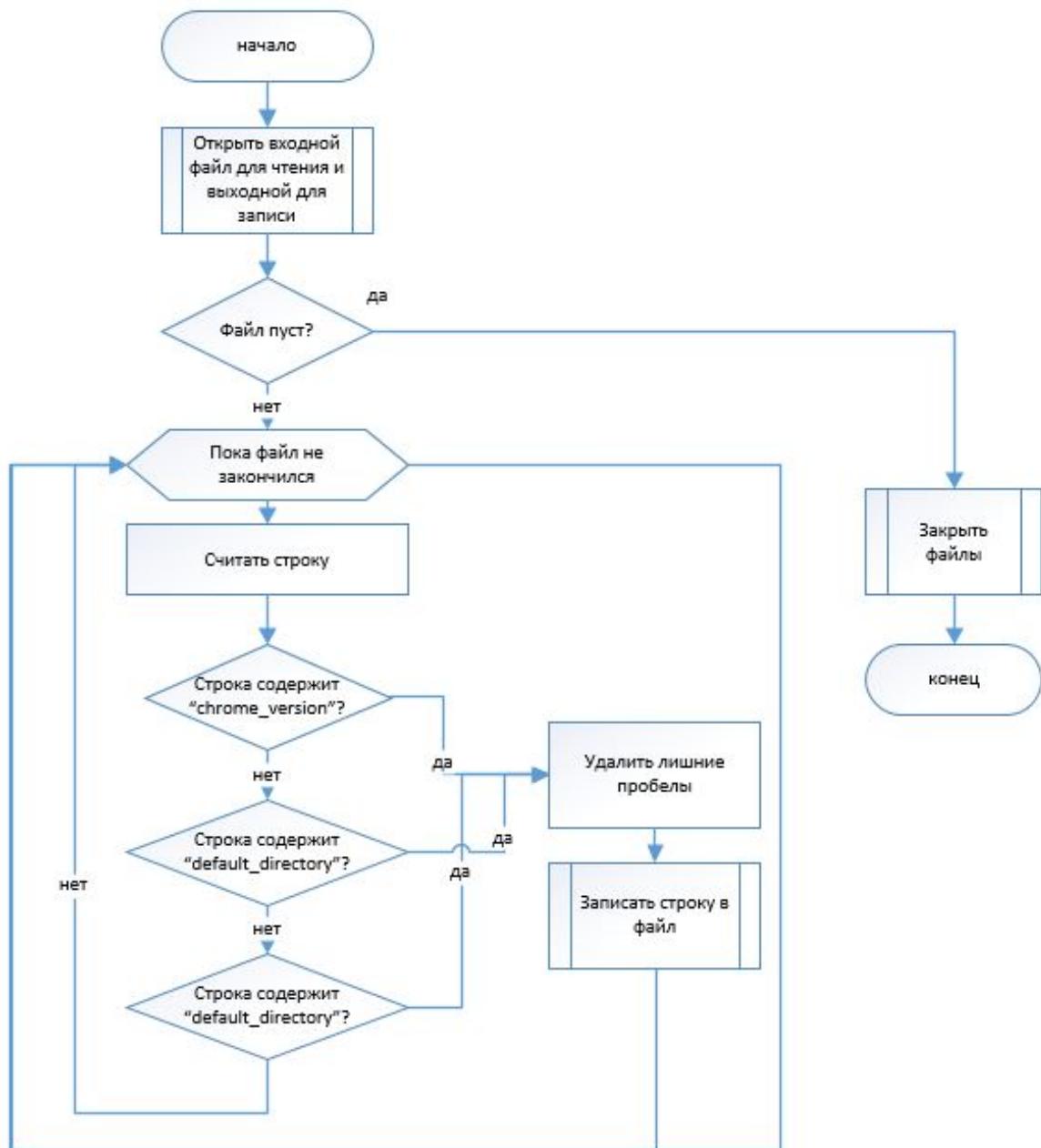


Рисунок 5.12 – Блок-схема алгоритма парсинга файла Preferences

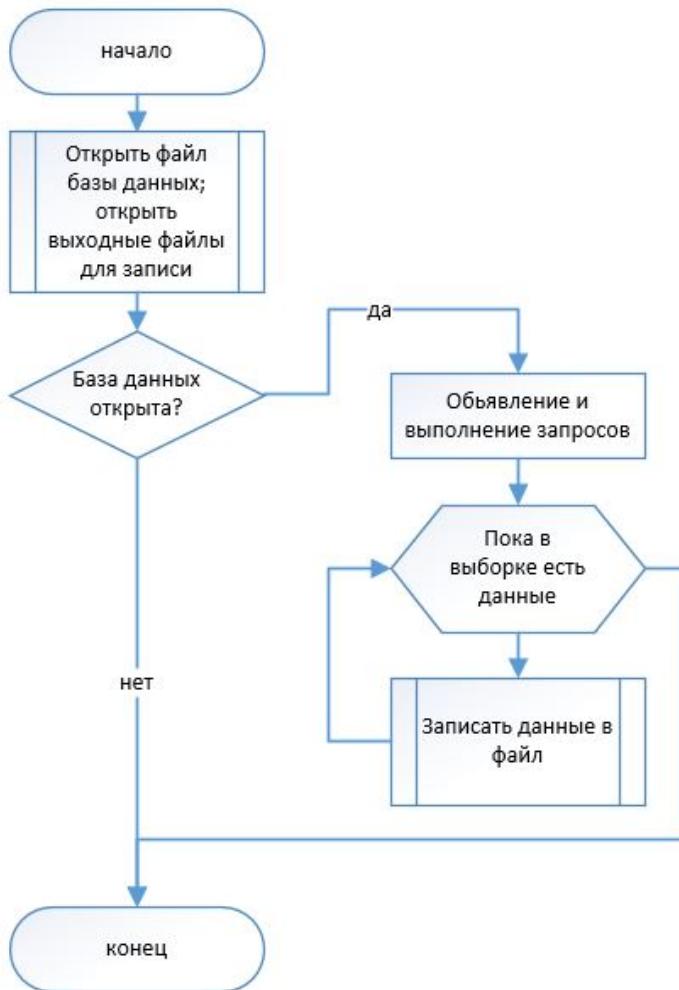


Рисунок 5.13 – Блок-схема алгоритма выборки данных из БД History

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <add>
3   <doc>
4     <field name="id">bookmark_2ce4440bbdbbc5c90e8fdda94c9a596a</field>
5     <field name="application">Chrome</field>
6     <field name="doc_type">bookmarks</field>
7     <field name="bookmark_url">http://www.yandex.ru/</field>
8     <field name="bookmark_name">Яндекс</field>
9   </doc>
10  <doc>
11    <field name="id">bookmark_fb8af80b528fd8b633d511bb0664e7ac</field>
12    <field name="application">Chrome</field>
13    <field name="doc_type">bookmarks</field>
14    <field name="bookmark_url">https://vk.com/feed</field>
15    <field name="bookmark_name">Новости</field>
16  </doc>
  
```

Рисунок 5.14 – Файл bookmarks.XML

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <add>
3   <doc>
4     <field name="id">preferences_8b9f68c6448a56183149128b9ada43ac</field>
5     <field name="application">Chrome</field>
6     <field name="doc_type">preferences</field>
7     <field name="preferences_param_name">default_directory_download</field>
8     <field name="preferences_param_value">&quot;C:\\\\Users\\\\blacksc\\\\Desktop&quot;:&quot;C:\\\\Users\\\\blacksc\\\\Desktop&quot;:&quot;C:\\\\Users\\\\blacksc\\\\Desktop&quot;</field>
9   </doc>
10  <doc>
11    <field name="id">preferences_13e4474f2558d39ca536bdb3a3911e83</field>
12    <field name="application">Chrome</field>
13    <field name="doc_type">preferences</field>
14    <field name="preferences_param_name">chrome_version</field>
15    <field name="preferences_param_value">&quot;last_chrome_version&quot;:&quot;38.0.2125.101&quot;,</field>
16  </doc>
17  <doc>
18    <field name="id">preferences_99582dcf5a0dfcb8997775f0ae5c5558</field>
19    <field name="application">Chrome</field>
20    <field name="doc_type">preferences</field>
21    <field name="preferences_param_name">username</field>
22    <field name="preferences_param_value">&quot;USERNAME&quot;:&quot;sgipovskoi@gmail.com&quot;</field>
23  </doc>
24 </add>
25

```

Рисунок 5.15 – Файл preferences.XML

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <add>
3   <doc>
4     <field name="id">history_ad7ddcadb91695f55ae0e5444a24eec3</field>
5     <field name="application">Chrome</field>
6     <field name="doc_type">history</field>
7     <field name="history_name">Браузер Chrome</field>
8     <field name="history_url">https://www.google.ru/intl/ru/chrome/bro
amp;installdataindex=defaultbrowser</field>
9     <field name="history_date">13057012312000000</field>
10   </doc>
11   <doc>
12     <field name="id">history_a72143dc985bda7a7c63fe1ec80a0aa</field>
13     <field name="application">Chrome</field>
14     <field name="doc_type">history</field>
15     <field name="history_name"></field>
16     <field name="history_url">https://dl.google.com/update2/1.3.24.15/
application?appguid%3D%7B8A69D345-D564-463C-AFF1-A69D9E530F96%7D%2
229151%7D%26lang%3Drus%26browser%3D2%26usagestats%3D0%26appname%3DG
26installdataindex%3Ddefaultbrowser</field>
17     <field name="history_date">13057012304000000</field>
18   </doc>

```

Рисунок 5.16 – Файл history.XML

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <add>
3   <doc>
4     <field name="id">history_download_fb4321a913b0903991e195f2ad5f24bb</field>
5     <field name="application">Chrome</field>
6     <field name="doc_type">history_download</field>
7     <field name="download_url">http://dlcdn.net.asus.com/pub/ASUS/nb/DriversForWin8/WiFi/Wifi_Intel_Wi
field>
8     <field name="download_path">C:\\Users\\blacksc\\Downloads\\Wifi_Intel_Win7_64_VER15704.zip</field>
9     <field name="download_start_time">13057012671412926</field>
10    <field name="download_size">100121965</field>
11    <field name="download_referrer">http://www.asus.com/ru/support/Download/3/411/0/2/Z9zboGSUuWyLgw
12  </doc>
13  <doc>
14    <field name="id">history_download_cf833e966164f4281adc57a588068864</field>
15    <field name="application">Chrome</field>
16    <field name="doc_type">history_download</field>
17    <field name="download_url">http://dlcdn.net.asus.com/pub/ASUS/nb/X75A/LAN_Atheros_Win7_64_Z201516.
18    <field name="download_path">C:\\Users\\blacksc\\Downloads\\LAN_Atheros_Win7_64_Z201516.zip</field>
19    <field name="download_start_time">13057012844385018</field>
20    <field name="download_size">5788258</field>
21    <field name="download_referrer">http://www.asus.com/ru/support/Download/3/411/0/2/Z9zboGSUuWyLgw
22  </doc>

```

Рисунок 5.17 – Файл download.XML

```
1 |<?xml version="1.0" encoding="UTF-8"?>
2 |<add>
3 |  <doc>
4 |    <field name="id">keyword_search_term_a8dc0a9d61c3d5cd221020aa0b947f95</field>
5 |    <field name="application">Chrome</field>
6 |    <field name="doc_type">search_term</field>
7 |    <field name="keyword_term">2гис томск</field>
8 |  </doc>
9 |  <doc>
10 |    <field name="id">keyword_search_term_2f3b040d1e658da6a70240b6e3afec29</field>
11 |    <field name="application">Chrome</field>
12 |    <field name="doc_type">search_term</field>
13 |    <field name="keyword_term">CSMA/CD</field>
14 |  </doc>
15 |  <doc>
16 |    <field name="id">keyword_search_term_6089cc6ab7e8133b567f7a43ed135eac</field>
17 |    <field name="application">Chrome</field>
18 |    <field name="doc_type">search_term</field>
19 |    <field name="keyword_term">FDDI</field>
20 |  </doc>
```

Рисунок 5.18 – Файл search_term.XML

5.3 Сбор информации из мессенджера ICQ

В ходе проведения компьютерной экспертизы может потребоваться извлечение всех возможных контакт-листов злоумышленника, сохраненных локально различными программами, в особенности программами мгновенного обмена сообщениями.

Программы мгновенного обмена сообщениями (Instant Messenger, IM) — программы-клиенты, предназначенные для обмена сообщениями в реальном времени через Интернет. С их помощью могут передаваться текстовые сообщения, звуковые сигналы, изображения, видео. Многие из таких программ-клиентов могут применяться для организации групповых текстовых чатов или видеоконференций.

Потребовалось проведение сравнительного анализа подобного рода программ-клиентов (ICQ, Pidgin, irc, Skype, Google Hangouts, Miranda IM и др.), в результате чего для разработки программного модуля в рамках проекта «Компьютерная экспертиза» была выбрана программа ICQ.

ICQ является централизованной службой мгновенного обмена сообщениями, использующей протокол OSCAR и локализованно хранящей различного рода информацию о пользователе: переданные и полученные электронные сообщения и файлы, а также список контактов.

В результате исследования данной службы была получена информация о структуре хранения данных ICQ. Это, в свою очередь, позволило написать программный модуль, позволяющий извлекать контакт-лист из файлов, сохраняемых ICQ на жестком диске злоумышленника.

5.3.1 Общие сведения о программах мгновенного обмена сообщениями

В ходе проведения сравнительного анализа программ-клиентов обмена мгновенными сообщениями исследовались и сравнивались основные функции различных программ, их версии и год популярности. Результаты данного исследования представлены в таблице 5.3.

Таблица 5.3 – Результаты сравнительного анализа программ-клиентов обмена мгновенными сообщениями

Имя	Год популярности	Основные функции	Текущая версия
ICQ	1990-ые	Микроблогинг, текстовые сообщения, заметки и напоминатели, аудио/видео сообщения, видеозвонки, отправка файлов, изображений и видео, звонки на мобильные и городские телефоны, поддержка популярных социальных сетей;	8.2 Build 7135 (Windows) — 2 сентября 2014 года 1.3.1 (Mac OS X) — 10 июля 2014 года Linux (beta) — 22 апреля 2011 года

Pidgin	2007	Метаконтакты, запись протокола событий, поддержка вкладок в окне разговора, подключение к нескольким аккаунтам одновременно, модульная структура, установка аватаров, настраиваемые сигналы действий пользователей, интеграция с GNOME, обмен файлами, кроссплатформенность;	2.10.9 (2 февраля 2014)
irc	1991	Текстовые сообщения, групповое/приватное общение, обмен файлами;	1.2.5-alt1 (2010-04-05)
Skype	2014	Текстовые сообщения (чат), видеозвонки, конференц-звонки, обмен файлами, звонки на мобильные и стационарные телефоны, передача изображения с монитора;	Windows: 6.18.66.106 (5 августа 2014); Windows 8.1: 2.8 (7 мая 2014); Linux x86: 4.3.0.37 (18 июня 2014); Mac OS X: 6.19 (9 июля 2014);
Google Hangouts	2013 - 2014	Видеоконференции, текстовые сообщения, онлайн трансляция через Youtube, обмен файлами, групповой чат, звонки на мобильные и стационарные телефоны;	Последняя версия — 2.0
Line	2014	Текстовые сообщения, аудио- и видеозвонки, передача файлов; имеет встроенную социальную сеть, в которой поддерживаются блоги и комментарии;	Последняя версия — 4.0.0 (03/09/2014)
Miranda IM	2005	Текстовые сообщения, обмен SMS-сообщениями с мобильными устройствами, поддержка плагинов; возможность определения приложения, при помощи которого работает собеседник; в контакт-листе выдает полные сведения о контакте, включая внешний IP-адрес; голосовая и видеосвязь отсутствуют;	0.10.24 (9 сентября 2014)
Yahoo! Messenger	2012	Текстовое сообщение, голосовое сообщение (в частности многопользовательский голосовой чат), видеоконференции, звонки на мобильные и стационарные телефоны, обмен файлами;	11.5 (Windows) / 2.5.3 (Mac) / 1.0.6 (Unix) (15 января 2012 (Windows))

Viber	2013 - 2014	Бесплатные звонки через Wi-Fi и сети 3G, текстовые сообщения, передача изображений, видео- и аудиосообщения;	Latest version: 4.1.0.1703
Mail.Ru Агент	2010 - 2011	Текстовые сообщения, IP-телефония, видеозвонки и отправка SMS, микроблогинг, конференции, обмен файлами.	Windows: 6.3, сборка 8050 — 2 сентября 2014; OS X: 4.0.2 — 29 мая 2014;
MySpace IM	2009	Поддержка Skype, звонки на сотовые телефоны с ПК, возможность получить собственный локальный номер с голосовой почтой обмен текстовыми сообщениями с другими пользователями MySpace; настройка уровня прозрачности для списка контактов и окна чата; журнал сообщений, а также его гибкая настройка; настройка прокси-сервера.	1.0.823.0 (1 декабря 2009 года)
QIP IM	2010	Поддержка внешних плагинов, уведомления о новой почте, обмен файлами и текстовыми сообщениями (в т.ч. SMS), аудио- и видеозвонки, интеграция с популярными соцсетями.	QIP 2012 — версии 4.0 (сборка 9379) (23 июня 2014 года);
Zoho Chat	2010	Групповой вэб - чат, интеграция с Yahoo, AIM, MSN, ICQ, GTalk и Jabber, обмен сообщениями с незарегистрированными в Zoho пользователями через браузер.	05.08.10 Zoho Chat

На выбор программы-клиента для реализации программного модуля повлияло не только исследование программ мгновенного обмена сообщениями, но и уровень развития навыков разработчика, а также были исключены программы Pigin и Skype, поскольку данные модули были разработаны ранее.

Далее потребовалось ознакомиться с самим приложением ICQ версии 8.2, найти директорию, в которой хранится необходимая информация, изучить форматы и содержимое найденных файлов.

Для идентификации пользователей служба ICQ использует UIN (Universal Identification Number) — уникальный для каждой учётной записи номер, состоящий из 5-9 арабских цифр. Этот номер присваивается учётной записи при первичной регистрации пользователя в системе, после чего, в паре с паролем, может использоваться для аутентификации в системе. Контакты злоумышленника будут храниться в виде пар значений — «ник» (nick) пользователя и его идентификационного номера (email) — разделе C:\Users\UserName\AppData\Roaming\ICQ - Profile.

Данная информация сохраняется программой ICQ в XML - документе, пример которого представлен на рисунке 5.19). Каждый xml-тэг соответствует одному элементу контактного листа и содержит значения атрибутов «nick» и «email», которые и будут считываться программой и сохраняться в выходной файл.

```
- <r>
<c email="455260724" nick="П@НкУША"/>
<c email="215474961" nick="Yorik"/>
<c email="227183455" nick="ShaudeR"/>
<c email="371665896" nick="СедЦеЕда"/>
<c email="379756640" nick="snowflake"/>
<c email="393353147" nick="Bodia_nv"/>
<c email="395562933" nick="Ромыч"/>
<c email="403710276" nick="Barney"/>
<c email="404114172" nick="M1r-acle"/>
<c email="405558668" nick=")Алёшка@)"/>
<c email="409599023" nick="Вождь плюшевых апачи"/>
<c email="422999080" nick="Ходящий Кор"/>
<c email="4274115" nick="Мэкш"/>
<c email="434189566" nick="Серега"/>
<c email="448238528" nick=".БеСПЕрСПЕкТИВЯк."/>
<c email="452214814" nick="Black Angel"/>
<c email="459063060" nick="_D.e.N.i.S_"/>
<c email="460626318" nick="AnekBot"/>
<c email="464770743" nick="ПуГоффк@"/>
<c email="478768966" nick="+РНЕ+Слава России*"/>
<c email="483586036" nick="@ксютKa"/>
<c email="486433999" nick="Оленьк@"/>
<c email="497820848" nick="Малышка"/>
<c email="552938635" nick="PEPSI MEN"/>
<c email="553807091" nick="Ягодная"/>
<c email="619197596" nick="Оленька)"/>
<c email="635766618" nick="Влад"/>
<c email="6861796" nick="I-Bot Переводчик"/>
<c email="7004092" nick="-ReanimatoR"/>
<c email="607589331" nick="Star Wars"/>
<c email="657182629" nick="Марина"/>
</r>
```

Рисунок 5.19 – Список контактов пользователя в формате XML

5.3.2 Реализация программного модуля

Реализация программного модуля включала в себя следующие шаги:

- 1) Изучение проекта соех;
- 2) Изучение особенностей работы с библиотеками QT;
- 3) Изучение особенностей работы с XML-форматом (языка разметки);
- 4) Изучение системы компьютерной верстки Latex для написания документации;
- 5) Изучение системы распределенного контроля версий Git и ее основных возможностей;
- 6) Установка ICQ 8.2 на виртуальную машину с операционной системой Windows 7;
- 7) Создание информационной базы для исследования (нескольких аккаунтов, обмен сообщениями и файлами);
- 8) Разработка программного модуля.

5.3.3 Алгоритм работы модуля

Алгоритм работы модуля выглядит следующим образом: на вход программе подается XML-документ, в котором приложение ICQ хранит контактный лист пользователя. Выполняется проверка, является ли данный файл доступным для чтения. Если он таковым является, то далее создается потоковая переменная, в которую считывается информация из файла, а затем программа находит значения нужных нам атрибутов «email» и «nick», записывая найденные значения в выходной файл, который также будет иметь формат XML-документа. Операция будет выполняться до тех пор, пока не будет достигнут конец файла, после чего файл благополучно закрывается и завершается выполнение программы.

Алгоритм синтаксического анализа входного XML-файла, содержащего контакт-лист, представлен на блок-схеме ниже (рис. 5.20).

Исходный код функции синтаксического анализа контактного листа приложения ICQ можно увидеть на рисунке 5.21.

Пример результата работы модуля в виде файла в формате XML представлен на рисунке 5.22.

5.3.4 Здачи на следующий семестр

В дальнейшем планируется дополнить модуль функцией рекурсивного обхода файловой системы для нахождения необходимых файлов, а также функциями для синтаксического анализа файлов с сообщениями злоумышленника.

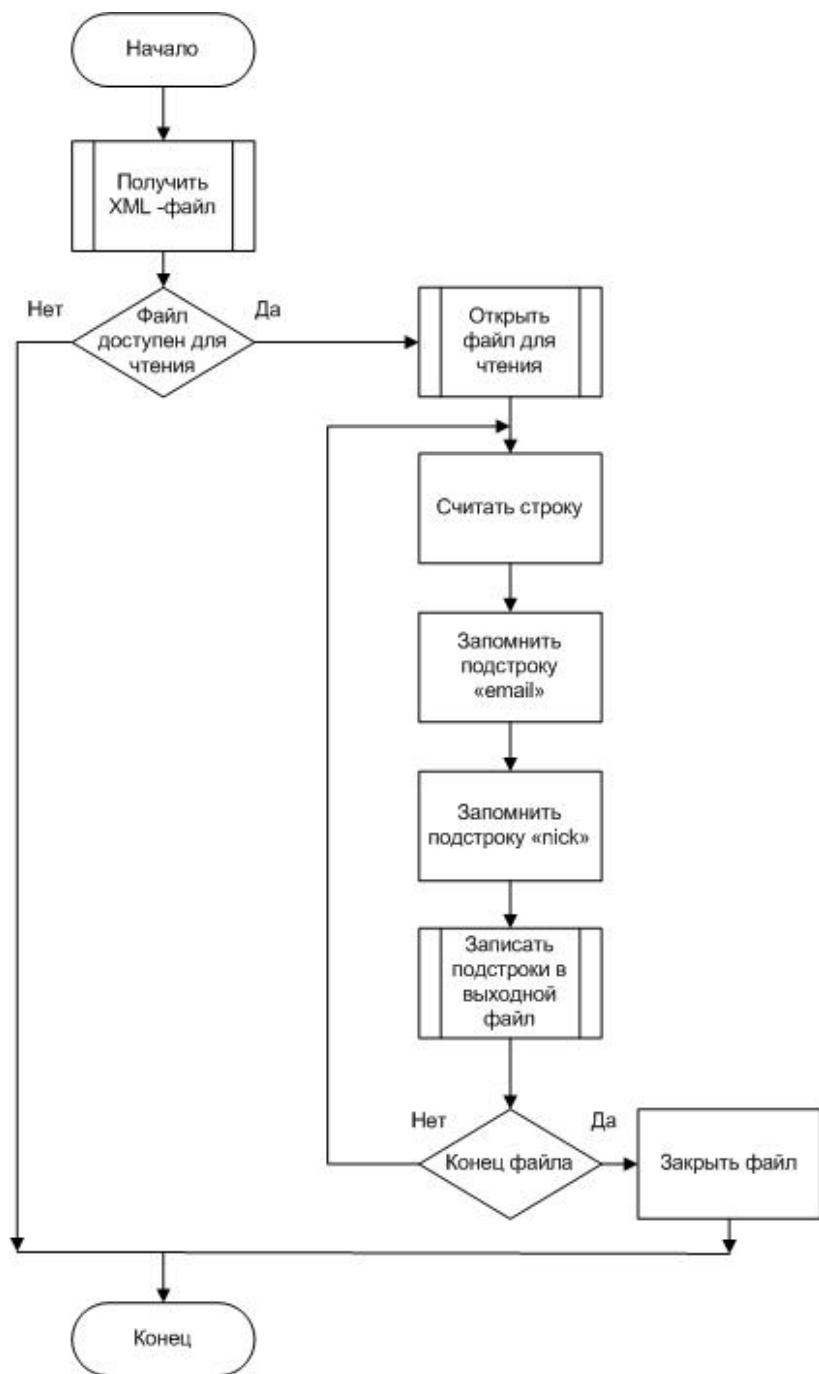
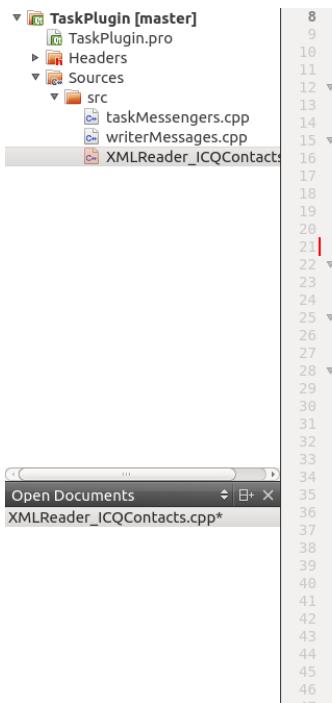


Рисунок 5.20 – Блок-схема синтаксического анализа входного файла



```

8     |
9     };
```

```

10    // -----
11
12    bool XMLReader_ICQContacts::read(QString inputFile, QString outPath)
13    {
14        QFile file(inputFile);
15        QDir dir(outPath);
16        dir.mkdir("icq");
17    }
18    writerMessagesPidgin icqContacts(outPath + "/icq/contacts.xml", "icq");
19
20    if(file.open(QIODevice::ReadOnly)) {
21        QXmlStreamReader xml(&file);
22
23        do {
24            xml.readNext();
25
26            if (xml.tokenType() != QXmlStreamReader::EndElement && xml.tagName() == "c")
27            {
28                QString mail, nick;
29                if(xml.attributes().value("email") != "")
30                    mail = xml.attributes().value("email").toString();
31
32                if(xml.attributes().value("nick") != "")
33                    nick = xml.attributes().value("nick").toString();
34
35                icqContacts.writeContactList(mail, nick);
36            }
37
38        } while(!xml.atEnd());
39        // закрываем файл если удалось его открыть
40        file.close();
41        return true;
42    }
43    return false;
44}
45
46

```

Рисунок 5.21 – Исходный код функции синтаксического анализа контактного листа приложения ICQ

```

- <add>
- <doc>
<field name="doc_type">contact</field>
<field name="application">icq</field>
<field name="contact_account">455260724</field>
<field name="contact_id">П@НКУША</field>
</doc>
- <doc>
<field name="doc_type">contact</field>
<field name="application">icq</field>
<field name="contact_account">215474961</field>
<field name="contact_id">Yorik</field>
</doc>
- <doc>
<field name="doc_type">contact</field>
<field name="application">icq</field>
<field name="contact_account">227183455</field>
<field name="contact_id">ShaudeR</field>
</doc>
-
```

Рисунок 5.22 – Пример результата работы модуля в виде файла в формате XML

5.4 Сбор информации из почтового клиента MS Outlook

В ходе проведения компьютерной экспертизы может возникнуть необходимость проанализировать электронные письма злоумышленника, его контакты и прикрепленные файлы. Подобную информацию можно получить из файлов, сохраняемых программой OutLook на ПК пользователя. Для осуществления данной задачи был разработан программный модуль Outlook 2007.

5.4.1 Некоторые сведения о почтовых клиентах

Почтовая программа (почтовый клиент, клиент электронной почты, мейлер, майл-клиент) — это ПО, которое инсталлируется на компьютер пользователя и предназначено для написания, получения, хранения, отправки электронной почты одного или нескольких пользователей (например, когда имеется несколько учетных записей на компьютере), или нескольких учетных записей пользователя.

В самом начале работы была найдена статья с расположением файлов сообщений, адресной книги, название прикрепленных файлов. В ходе поиска в операционной системе был найден только файл данных pst, при разборе выяснилось, что данное расположение соответствует версии Outlook 2003, но она используется крайне редко, поэтому выбор пал на следующие версии (Outlook 2007, Outlook 2010 и Outlook 2013), которые все данные хранят в бинарном файле pst. Версию Outlook 2013 можно установить в ОС Windows версии не ниже 7. Окончательно для дальнейшего написания программного модуля была выбрана версия Outlook 2007, так как на данную версию почтового клиента у разработчика имелась лицензия.

5.4.2 Реализация программного модуля

Реализация данного программного модуля включала в себя следующие шаги:

- 1) Изучение проекта соех;
- 2) Изучение особенностей работы с библиотеками QT;
- 3) Изучение особенностей работы с XML-форматом (языка разметки);
- 4) Изучение системы компьютерной верстки Latex для написания документации;
- 5) Изучение системы распределенного контроля версий Git и ее основных возможностей;
- 6) Исследование почтового клиента Outlok 2007 (хранение логов и настроек);
- 7) Изучение различных файловых форматов, таких как PST, PAB, MSG, RTF, HTML;
- 8) Разработка программного модуля для сбора информации из почтового клиента MS Outlook и ее записи в XML-файл.

5.4.3 Исследование файловых форматов

После нахождения файла, где хранится информация о интересующих нас данных, возник вопрос как считать эти данные, так как формат pst предназначен для открытия только программой Outlook и является бинарным файлом. Для решения задачи и подтверждения предположения о том, что данный файл содержит интересующую нас информацию, была найдена программа Readpst для чтения формата pst под операционной средой Linux. Но исходный код данной программы не был найден. После чего возникла необходимость поиска способов чтения и любых других действий с

данным форматом.

Пример чтения файла в формате pst представлен на рисунке 5.23.

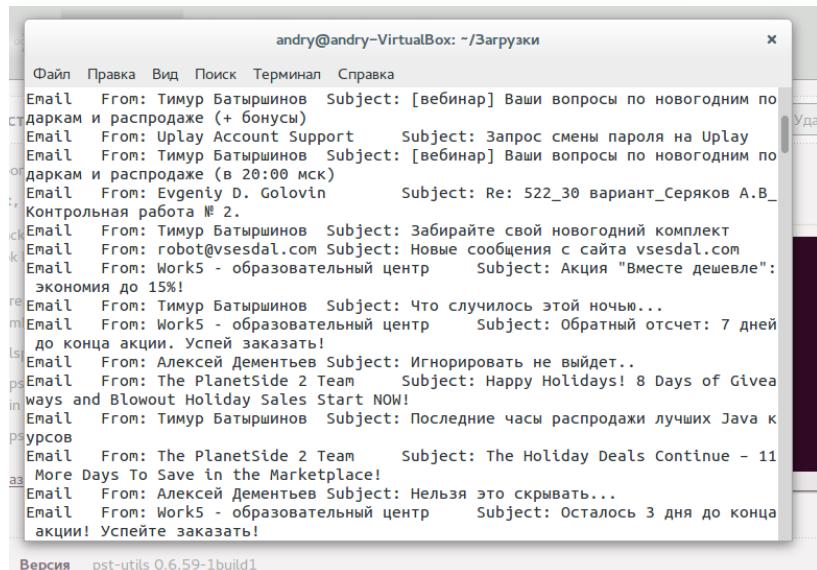


Рисунок 5.23 – Чтение файла в формате pst

В статье от 2009 года компании Microsoft говорится о расшифровке и полном описании некоторых форматов MS. После прочтения данного документа был проведен поиск технического описания формата pst, в результате которого был найден архив с техническими описаниями некоторых форматов, в том числе и формата pst, на английском языке. В нем описана основная структура. Файловые структуры Pst логически устраиваются в трех слоях, как это показано на рисунке 5.24.

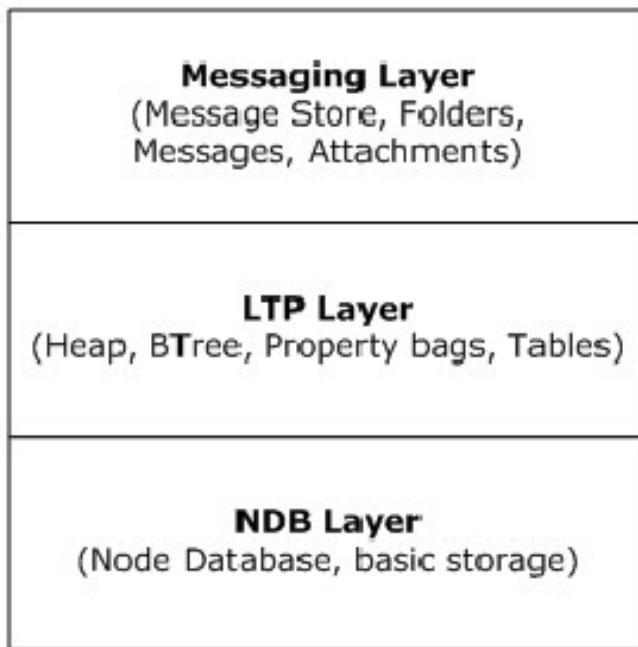


Figure 1: Logical layers of a PST file

Рисунок 5.24 – Слой NDB

Слой NDB состоит из базы данных узлов, которая представляет склады низшего уровня фор-

мат файла PST. С точки зрения внедрения слой NDB состоит из заголовка, файла информации о распределении, блоков, узлов и двух BTrees: узел BTree (NBT) и блок BTree (BBT). NBT содержит ссылки на все доступные узлы в файле Pst. Его выполнение Btree позволяет в поиске размещать любой специфический узел. Bbt содержит ссылки на все блоки данных файла Pst. Его выполнение Btree позволяет в поиске размещать любой специфический блок.

Слой Ltp (Списки, Таблицы, и Свойства) осуществляет высокоуровневые понятия сверху конструкции NDB. Основные элементы слоя LTP — Property Context (PC) и Table Context (TC). PC представляет коллекцию свойства. TC представляет двумерный стол. Строки представляют коллекцию свойств. Колонки представляют свойства в пределах строк.

Передающий слой состоит из высокоуровневых правил и бизнес-логики, которые позволяют объединять структуры LTP и слои NDB и интерпретировать как объекты «Папки», объекты «Сообщения», объекты «Приложения» и «Свойства». Передающий слой также определяет правила и требования.

Это сопровождается изменением содержания файла PST так, чтобы измененный файл PST мог все еще быть успешно прочитан путем внедрения этого формата файла. Последовательности бит информации записываются в специальные для них блоки, что позволяет, опираясь на тег блока, брать нужное количество битовой информации для выборки интересующей нас информации. Например: 0x1000001f -PidTagBody_W-PtypBinary-58 Byte(s), 0x10130102-PidTagHtml-PtypBinary-1638 Byte(s), 0x1035001f-PidTagInternetMessageId_W-PtypBinary-164 Byte(s), 0x3003001f -PidTagEmailAddress_W-178 Byte(s).

После чего был изучен код других плагинов для нахождения возможных решений возникшей проблемы. В результате проведенных исследований были написаны два программных модуля, требующих дальнейшей модернизации. Первая программа ищет файлы разрешения pst и rab в папках стандартного размещения их в OutLook. Вторая программа считывает побитово файл pst и записывает полученные данные в строковую переменную.

Алгоритм поиска файлов в папках пользователей представлен на блок-схеме ниже (рис. 5.25 и 5.26).

Побитовое считывание файла формата PST представлено на блок-схеме на рисунке 5.27.

5.4.4 Задачи на следующий семестр

В следующем семестре планируется переделать поиск файлов не только в стандартном расположении Outlook, но и в других директориях, за непродолжительное время. Также нужно модернизировать алгоритм чтения потока данных с фильтрацией ненужной информации, написать часть кода, в которой полученные данные будут записываться в xml-файл, изучить старые возможные форматы хранения данных rab, ost, msq, после чего написать функции работы с этими файлами в плагине Outlook2007 и записать полученные данные в xml-файл.

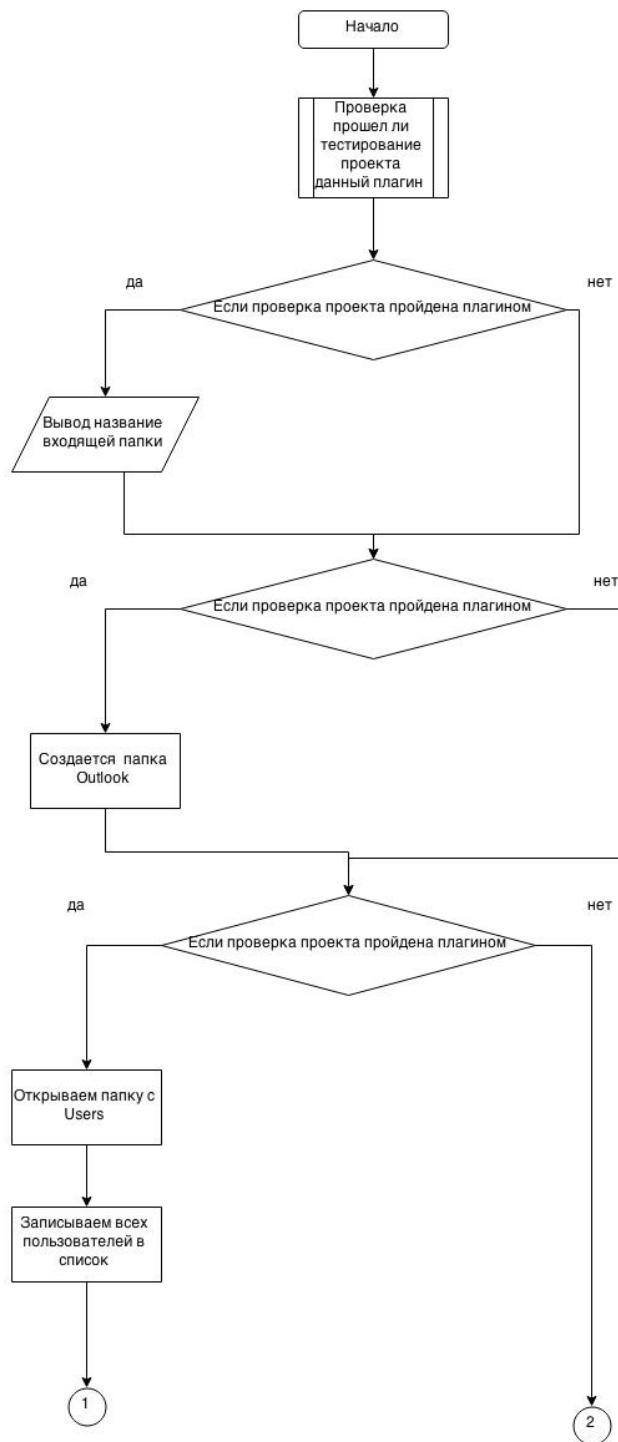


Рисунок 5.25 – Блок-схема алгоритма поиска файлов в папках пользователей

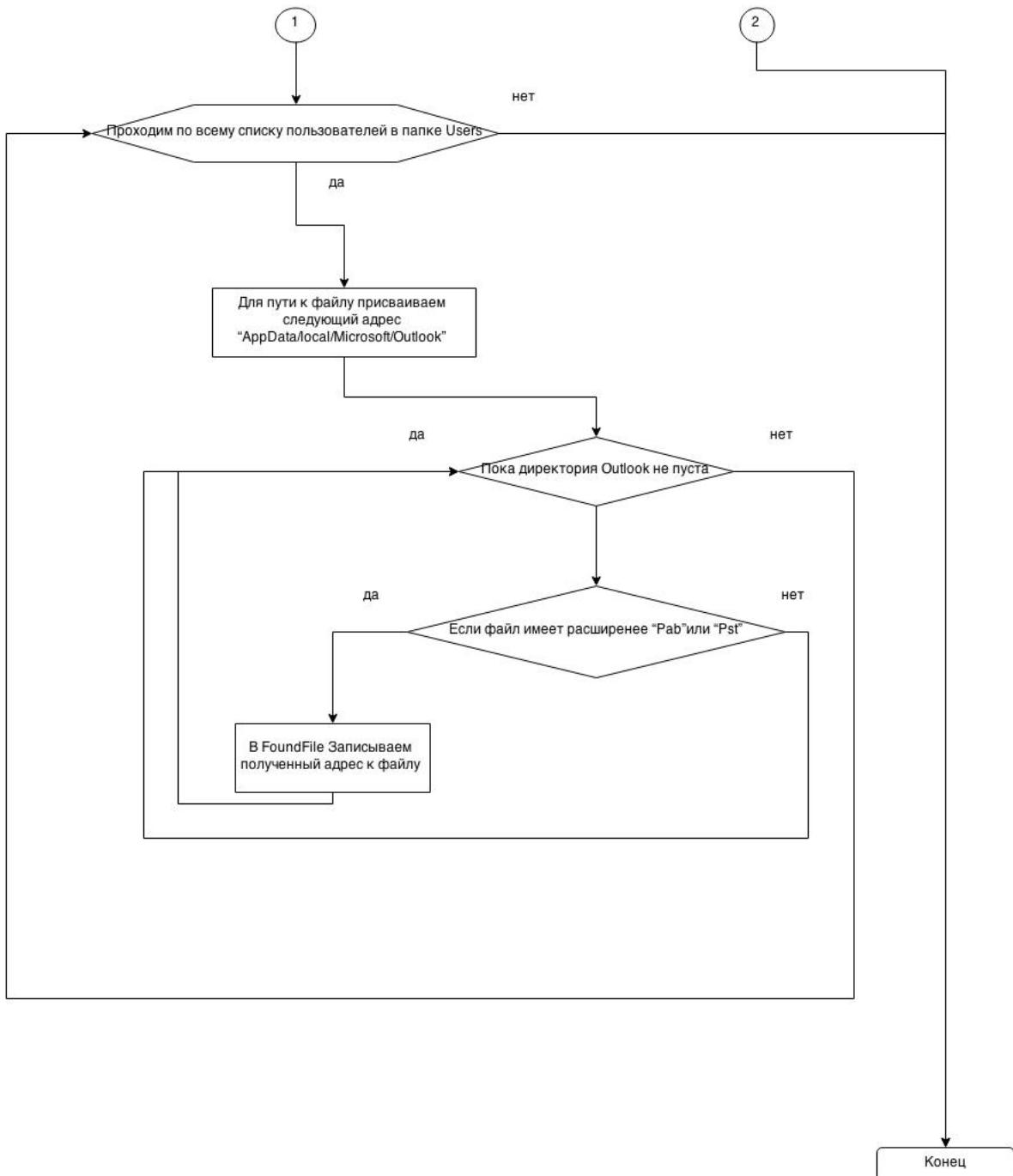


Рисунок 5.26 – Блок-схема алгоритма поиска файлов в папках пользователей (продолжение)

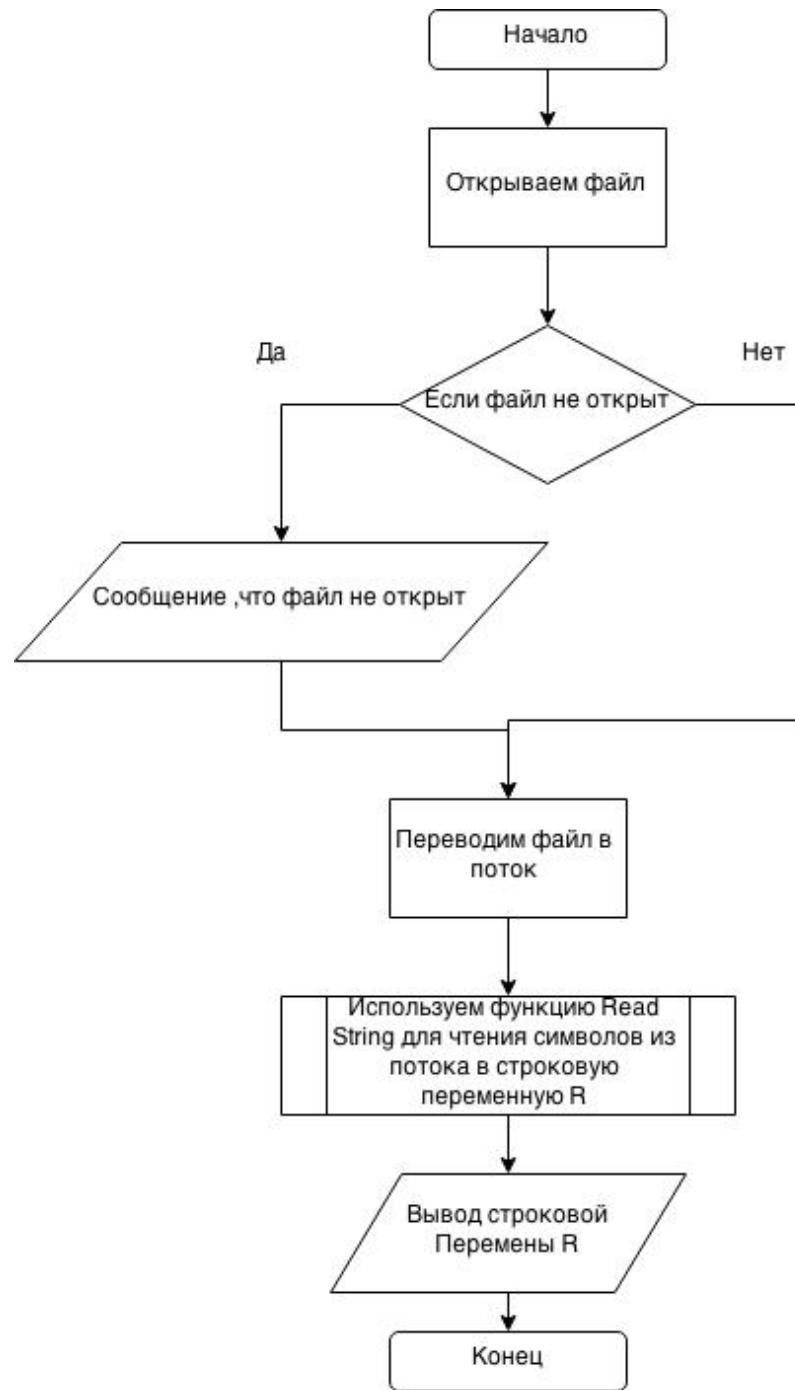


Рисунок 5.27 – Блок-схема алгоритма побитового считывания файла формата PST

5.5 Идентификации файлов изображений

Целью работы стало написание модуля для программного комплекса. Модуль выполняет проверку файлов-изображений на их подлинность (являются ли они действительно изображением), и выводить информацию в формате XML.

5.5.1 Реализация программного модуля

Сначала необходимо было узнать, как различать изображения и файлы с расширением изображений. Было принято решение считывать заголовки файлов и сравнивать их с корректными заголовками, являющимися уникальными для соответствующего формата. Далее была найдена информация о заголовках нескольких форматов. Форматы заголовков представлены в таблице 5.4.

Таблица 5.4 – Форматы заголовков

Формат	Заголовок
JPEG	FFD8
PNG	89504E47
GIF	474946
BMP	424D
TIFF	49492A

Также было принято решение проверять на корректность конец файла, так как к изображению можно прикрепить архив в конец файла. На данный момент реализована проверка конца файла для форматов JPEG и PNG как самых популярных и простых для реализации (табл. 5.5).

Таблица 5.5 – Форматы конца файла

Формат	Конец
JPEG	FFD9
PNG	AE426082

5.5.2 Алгоритм работы модуля

Алгоритм программы, проверяющей заголовки и концы файлов, представлен на рисунке 5.28.

5.5.3 Структура XML-файла

Файл начинается с пролога, описывающего версию XML и кодировку. Далее идет начальный элемент `<add>`, а для каждого изображения создается элемент `<doc>`. В теле `<doc>` записаны поля `<field name="value"></field>`, далее закрывается элемент `</doc>`, а в конце документа находится конечный элемент `</add>`.

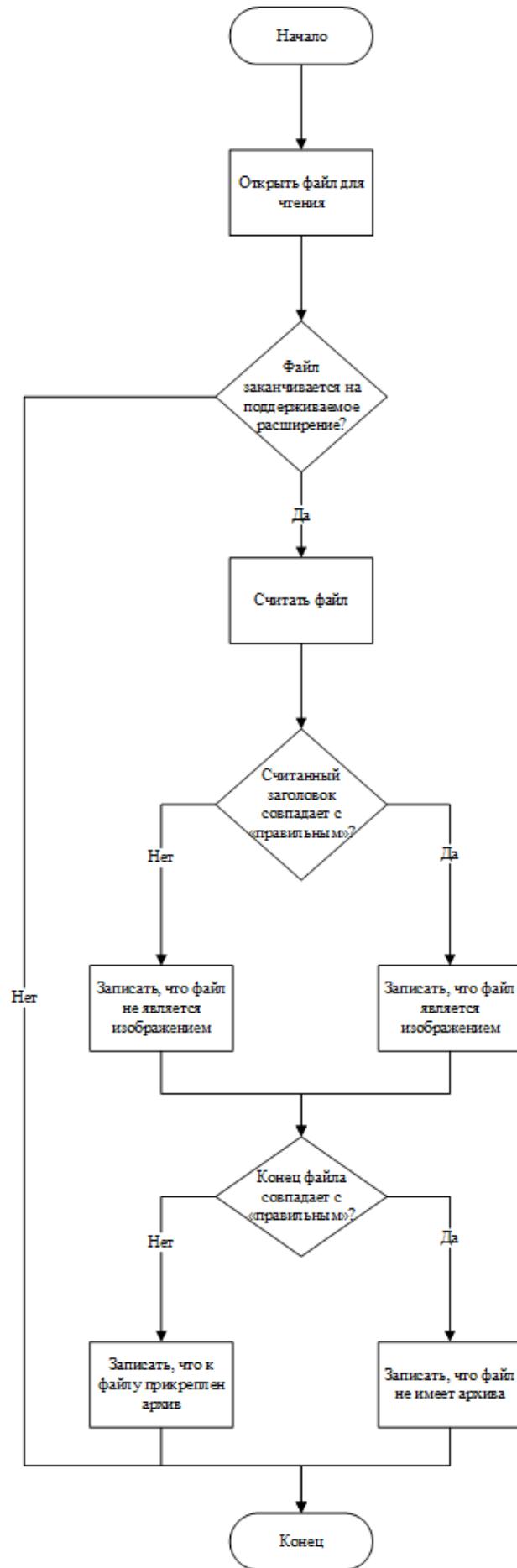


Рисунок 5.28 – Блок-схема алгоритма программы, проверяющей заголовки и концы файлов

```

<?xml version="1.0" encoding="UTF-8"?>
- <add>
  + <doc>
  - <doc>
    <field name="id">gif_ed20a76b8f247693dd1c394eb9bec245</field>
    <field name="application">images</field>
    <field name="doc_type">image</field>
    <field name="image_result">correct</field>
    <field name="contains_archive">false</field>
    <field name="image_path">C:/Users/Tapyc/Documents/build-image_
    <field name="image_datecreate">Вс дек 14 21:03:31 2014</field>
    <field name="image_datemodify">Вс дек 14 21:03:31 2014</field>
  </doc>
  - <doc>
    <field name="id">jpg_d41d8cd98f00b204e9800998ecf8427e</field>
    <field name="application">images</field>
    <field name="doc_type">image</field>
    <field name="image_result">correct</field>
    <field name="contains_archive">false</field>
    <field name="image_path">C:/Users/Tapyc/Documents/build-image_
    <field name="image_datecreate">Вс дек 14 21:01:28 2014</field>
    <field name="image_datemodify">Вс дек 14 21:01:28 2014</field>
  </doc>
  + <doc>
  - <doc>
    <field name="id">png_d41d8cd98f00b204e9800998ecf8427e</field>
    <field name="application">images</field>
    <field name="doc_type">image</field>
    <field name="image_result">correct</field>
    <field name="contains_archive">false</field>
    <field name="image_path">C:/Users/Tapyc/Documents/build-image_
    <field name="image_datecreate">Вс дек 14 21:02:07 2014</field>
    <field name="image_datemodify">Вс дек 14 21:02:07 2014</field>
  </doc>

```

Рисунок 5.29 – Структура XML-файла

Пример структуры XML-файла представлен на рисунке 5.29.

Значение поля id — «формат изображения»_«MD5 сумма от файла». Поля image_result и contains_archive непосредственно указывают на то, является ли файл изображением, и прикреплен ли к нему архив соответственно. Также имеется полный путь до файла, дата создания и дата изменения. В программу были введены следующие тесты: 5 файлов изображений для базовой проверки, 5 текстовых файлов с расширением изображений, и 2 файла изображений с прикрепленным к ним архивом. Все файлы программа распознала корректно и вывела в XML файл.

Библиотеки Qt, использованные для написания программного модуля:

- QFile — для открытия и чтения всего файла;
- QString — для работы со строками в программе;
- QDebug — для вывода в консоль в целях отладки;
- QDataStream — для считывания отдельных байтов из потока;
- QXmlStreamWriter — для вывода в XML;
- QFileInfo — для считывания информации о файле (даты и т.д.);
- QDateTime — для работы с форматом даты, в который записывается информация о файле;
- QCryptographicHash — для получение md5 суммы.

Результат работы программы представлен на рисунке 5.30.

5.5.4 Задачи на следующий семестр

В будущем в программе будут реализованы:

- поддержка новых расширений;
- возможность проверить наличие архива для всех расширений;
- рекурсивный обход директорий.

```
D:\Soft\Qt\Tools\QtCreator\bin\qtcreator_process_stub.exe

file name = "pic.bmp.bmp" header = "424d" image_result = "correct" contains_archive= "false"
file name = "pic.gif.gif" header = "474946" image_result = "correct" contains_archive= "false"
file name = "pic.jpg.jpg" header = "ffd8" image_result = "correct" contains_archive= "false"
file name = "pic.tiff.tif" header = "49492a" image_result = "correct" contains_archive= "false"
file name = "pic.png.png" header = "89504e47" image_result = "correct" contains_archive= "false"
file name = "text.bmp.bmp" header = "7465" image_result = "incorrect" contains_archive= "false"
file name = "text.gif.gif" header = "746573" image_result = "incorrect" contains_archive= "false"
file name = "text.jpg.jpg" header = "7465" image_result = "incorrect" contains_archive= "false"
file name = "text.tiff.tif" header = "746573" image_result = "incorrect" contains_archive= "false"
file name = "text.png.png" header = "74657374" image_result = "incorrect" contains_archive= "false"
file name = "archive.jpg.jpg" header = "ffd8" image_result = "correct" contains_archive= "true"
file name = "archive.png.png" header = "89504e47" image_result = "correct" contains_archive= "true"
Press <RETURN> to close this window...
```

Рисунок 5.30 – Результат работы программы

5.6 Сбор и анализ информации из реестра ОС MS Windows

Реестр Windows или системный реестр — иерархически построенная база данных параметров и настроек в большинстве операционных систем Microsoft Windows. Реестр содержит информацию и настройки для аппаратного обеспечения, программного обеспечения, профилей пользователей, предустановки. Большинство изменений в Панели управления, ассоциации файлов, системные политики, список установленного ПО фиксируются в реестре. И именно поэтому получение его данных является крайне важной задачей для дальнейшего исследования в ходе компьютерной экспертизы.

Конечной целью работы является разработка программного модуля (winreg) для конвертирования данных из реестра в подходящий для дальнейшего исследования формат. Для достижения данной цели были установлены следующие задачи, разбитые на три этапа.

Этап первый — «введение»:

- ознакомление с софтом — системой для комплексного анализа использования операционных систем семейств Microsoft Windows;
- ознакомление с открытой средой разработки QtCreator;
- ознакомление со системой верстки документов Latex;
- ознакомление со системой контроля версий Git, обучение основные ее возможностям.

Этап второй — «исследование»:

- исследование модели формирования и структуры реестра Windows;
- исследование готовых решений для работы реестром Windows.

Этап третий — «разработка»:

- выбор метода получения информации из реестра;
- разработка приложения.

5.6.1 Этап второй — «исследование». Структура и модель формирования

Как и было сказано ранее, реестр Windows — иерархически построенная база данных, формируемая на основе конфигурации ПК и пользовательских или программных настроек. В собранном виде он имеет древовидный вид с пятью корневыми разделами (табл. 5.6).

Таблица 5.6 – Корневые разделы реестра и их краткое описание

HKEY_CLASSES_ROOT	Подраздел HKEY_LOCAL_MACHINE\Software
HKEY_CURRENT_USER	Корневой для данных конфигурации пользователя; подраздел HKEY_USERS
HKEY_LOCAL_MACHINE	Параметры конфигурации ПК
HKEY_USERS	Все активные профили пользователей
HKEY_CURRENT_CONFIG	Информации об оборудовании ПК

Каждый раздел состоит из некоторого количества подразделов, отвечающих за различные настройки программного обеспечения, самой операционной системы или служебную информацию.

Реестр Windows формируется в момент загрузки операционной системы

из некоторых исходных файлов, список которых может быть найден по пути: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\hivelist (рис. 5.31).

Имя	Тип	Значение
\b\ (По умолчанию)	REG_SZ	(значение не присвоено)
\b\REGISTRY\MACHINE\BCD00000000	REG_SZ	\Device\HarddiskVolume5\EFI\Microsoft\Boot\BCD
\b\REGISTRY\MACHINE\HARDWARE	REG_SZ	
\b\REGISTRY\MACHINE\SAM	REG_SZ	\Device\HarddiskVolume7\Windows\System32\config\SAM
\b\REGISTRY\MACHINE\SECURITY	REG_SZ	\Device\HarddiskVolume7\Windows\System32\config\SECURITY
\b\REGISTRY\MACHINE\SOFTWARE	REG_SZ	\Device\HarddiskVolume7\Windows\System32\config\SOFTWARE
\b\REGISTRY\MACHINE\SYSTEM	REG_SZ	\Device\HarddiskVolume7\Windows\System32\config\SYSTEM
\b\REGISTRY\USER\DEFAULT	REG_SZ	\Device\HarddiskVolume7\Windows\System32\config\DEFAULT
\b\REGISTRY\USER\S-1-5-19	REG_SZ	\Device\HarddiskVolume7\Windows\ServiceProfiles\LocalService\NTUSER.DAT
\b\REGISTRY\USER\S-1-5-20	REG_SZ	\Device\HarddiskVolume7\Windows\ServiceProfiles\NetworkService\NTUSER.DAT
\b\Registry\User\S-1-5-21-3981227007-...	REG_SZ	\Device\HarddiskVolume7\Users\Oleg\NTUSER.DAT
\b\Registry\User\S-1-5-21-3981227007-...	REG_SZ	\Device\HarddiskVolume7\Users\Oleg\AppData\Local\Microsoft\Windows\UsrClass.dat

Рисунок 5.31 – Пример содержимого hivelist

Непосредственно сам механизм сбора на данном этапе перестал быть интересен, поскольку интерес теперь представляют исходные файлы (далее «сырые» файлы реестра).

5.6.2 Этап второй — «исследование». Исследование готовых решений

Поскольку система комплексного анализа использования Windows не предполагает непосредственного запуска самой исследуемой ОС, дальнейшую работу необходимо вести имя лишь доступ к файловой системе, что, в свою очередь, предполагает работу лишь с «сырыми» файлами реестра. Для этого необходимо специальное ПО.

Были рассмотрены два приложения для работы с исходными файлами реестра: Fred (Forensic Registry EDitor) и chntpw.

Chntpw — утилита, позволяющая работать с «сырыми» файлами реестра. Изначально она была написана лишь для сбора паролей Windows, но в последствии была расширена для редактирования всего реестра (рис. 5.32).

```
b4el@b4el-VirtualBox:~$ chntpw
chntpw version 0.99.6 110511 , (c) Petter N Hagen
chntpw: change password of a user in a Windows SAM file,
or invoke registry editor. Should handle both 32 and 64 bit windows and
all version from NT3.x to Win7
chntpw [OPTIONS] <samfile> [systemfile] [securityfile] [otherreghive] [...]
-h      This message
-u <user> Username to change, Administrator is default
-l      list all users in SAM file
-i      Interactive. List users (as -l) then ask for username to change
-e      Registry editor. Now with full write support!
-d      Enter buffer debugger instead (hex editor),
-v      Be a little more verbose (for debugging)
-L      For scripts, write names of changed files to /tmp/changed
-N      No allocation mode. Only same length overwrites possible (very safe
mode)
-E      No expand mode, do not expand hive file (safe mode)
See readme file on how to get to the registry files, and what they are.
Source/binary freely distributable under GPL v2 license. See README for details.
NOTE: This program is somewhat hackish! You are on your own!
```

Рисунок 5.32 – Параметры запуска chntpw

Плюсы и минусы использования данной утилиты приведены в таблице 5.7. Из-за ограничений данной утилиты, ее использование в проекте невозможно.

Таблица 5.7 – Корневые разделы реестра и их краткое описание

Плюсы	Минусы
Есть в стандартном репозитории Ubuntu Возможна работа с bash	Ограничения использования GPLv2

Forensic Registry EDitor (fred) — свободно распространяемый, с открытым исходным кодом редактор реестра Windows (табл. 5.8).

Таблица 5.8 – Корневые разделы реестра и их краткое описание

Плюсы	Минусы
Открытый исходный код Публичные репозитории	Последняя версия данного редактора была написана на QT4 версии, а система работает на 5-ой Ограничение лицензии GNU v3 запрещает использование исходного кода в коммерческих проектах

Вид окна QTCreator с ошибкой при попытке компиляции в QT5 приведен на рисунке 5.33.

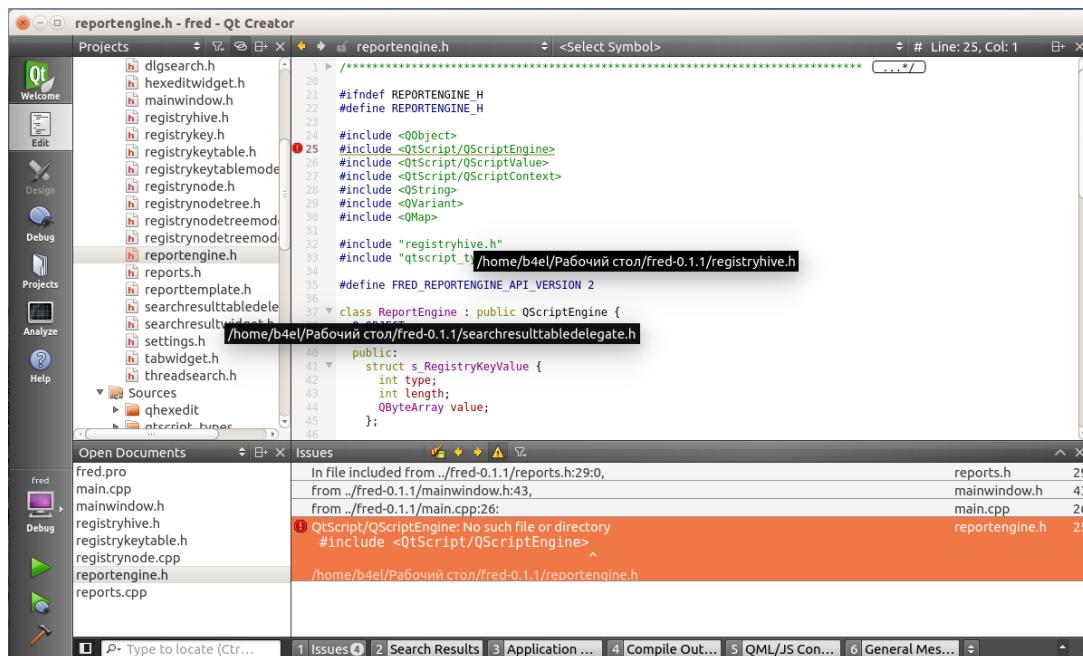


Рисунок 5.33 – Ошибка при попытке компиляции в QT5

Ошибка отсутствия в Qt5 необходимых библиотек приведена на рисунке 5.34.

Из-за сложности переноса утилиты и ограничений её использования, было принято решение о частичном заимствовании логики при написании самостоятельного плагина к системе.

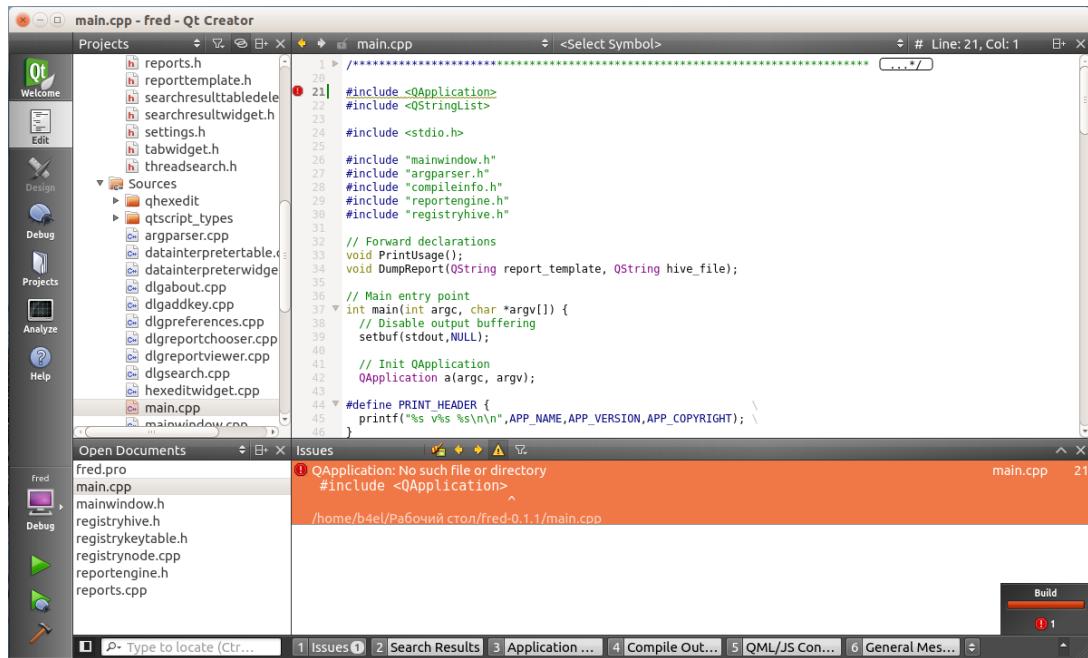


Рисунок 5.34 – Отсутствие в Qt5 необходимых библиотек

5.6.3 Этап третий — «разработка». Выбор метода получение информации из реестра

Рассмотренные выше утилиты позволяют работать с реестром лишь через себя. Использование Cntrpw как полностью, так и частично невозможно из-за ограничений лицензий. Утилита fred также ограничена в использовании, но у него имеются открытые исходные коды в свободном доступе, а значит, на них можно опираться при разработке своей программы.

5.6.4 Этап третий — «разработка». Разработка приложения

Для получения данных из реестра, фактически, нужно проводить реверс формата исходных файлов реестра Windows. Первым шагом стоит определения структуры файла. Он (файл) оказался бинарным (рис. 5.35).

Для работы с бинарными файлами используется библиотека QDataStream. Пример работы первой версии программы, выводящей строковые переменные, представлен ниже (рис. 5.36).

Процесс извлечения строковых переменных из реестра можно увидеть на рисунке 5.37.

5.6.5 Задачи на следующий семестр

В ходе дальнейшей разработки планируется определить условную разметку файла, которая определяет принадлежность каждого раздела к другому, а также получить данные из разделов-листьев и перевести эти данные в xml-представление.

5.6.6 Ссылки на Интернет-ресурсы

Ссылки:

- страница утилиты fred — <https://www.pinguin.lu/fred>;
- пример использования cntrpw — <http://habrahabr.ru/post/94764>;
- страница утилиты chntpw — <http://pogostick.net/pnh/ntpasswd>;

/home/b4el/Рабочий стол/reg/DEFAULT				0x0000000000	0%
00000000	72	65	67	66	DD 00 00 00
00000010	15	E5	CF	01	01 00 00 00
00000020	01	00	00	00	20 00 00 00
00000030	74	00	65	00	6D 00 52 00
00000040	53	00	79	00	73 00 74 00
00000050	5C	00	43	00	6F 00 6E 00
00000060	44	00	45	00	46 00 41 00
00000070	E8	D2	CE	6C	01 6E DE 11
00000080	E8	D2	CE	6C	01 6E DE 11
00000090	00	00	00	00	E9 D2 CE 6C
000000A0	0B	CD	18	24	72 6D 74 6D
000000B0	00	00	00	00	00 00 00 00
000000C0	00	00	00	00	00 00 00 00
000000D0	00	00	00	00	00 00 00 00
000000E0	00	00	00	00	00 00 00 00
000000F0	00	00	00	00	00 00 00 00
00000100	00	00	00	00	00 00 00 00
00000110	00	00	00	00	00 00 00 00
00000120	00	00	00	00	00 00 00 00
00000130	00	00	00	00	00 00 00 00
00000140	00	00	00	00	00 00 00 00
00000150	00	00	00	00	00 00 00 00

Рисунок 5.35 – Представление файла DEFAULT в HEX редакторе

```

main.cpp - binatyexp - Qt Creator
Projects Sources main.cpp HexToString(QDataStream &stream)
Qt Welcome binatyexp binatyexp.pro
Edit Sources main.cpp
Design main.cpp
Debug Open Documents binatyexp.pro main.cpp
Analyze
Help
binatyexp
Type to locate (Ctrl+Shift+F) Issues Sea... App... Co... QM... Gen...

```

```

1 #include <QCoreApplication>
2 #include <iostream>
3 #include <QDir>
4 #include <QString>
5 #include < QMap>
6 #include <QFile>
7 #include <QTextStream>
8
9 QString HexToString(QDataStream &stream)
10 {
11     QString res = "";
12     while(true || !stream.atEnd())
13     {
14         char data[2];
15         stream.readRawData(data, 2);
16
17         if (data[0] == 0x00)
18             break;
19         else
20             res += QChar::fromLatin1(data[0]);
21
22     }
23     return res;
24 }
25
26 int main(int argc, char *argv[])
27 {
28     QCoreApplication a(argc, argv);
29     QString r;
30     QFile *m_file = new QFile ("..\\rawdata");
31     bool m_bOpen = m_file->open(QIODevice::ReadOnly);
32     if (!m_bOpen)
33     {

```

Рисунок 5.36 – Исходный код программы

- информация по реестру — http://www.forensicswiki.org/wiki/Windows_Registry;
- описание видов лицензий — <http://www.gnu.org/licenses/licenses.html>;
- описание xml — <https://ru.wikipedia.org/wiki/XML>.

```
DateTime  
16AD5458A96CD7C3p  
Count  
DateTime  
1735E6BCA8F46527p  
Count  
DateTime  
6522F3EEDAE37075p  
Count  
DateTime  
DateTime  
hbin  
FAF44B824535C819  
KA #  
Time  
3ACCADB0850D2E2Bx  
Countr  
DateTime  
Time  
B51B73020ADA099p  
Counts  
DateTime  
Time  
5500687DEAC1EBE6p
```

Рисунок 5.37 – Извлечение строковых переменных из реестра

Заключение

В данном семестре нашей группой была выполнена часть работы по созданию автоматизированного программного комплекса для проведения компьютерной экспертизы, проанализированы дальнейшие перспективы и поставлены цели для дальнейшего развития проекта.

Список использованных источников

- 1 Федотов Николай Николаевич. Фorenзика - компьютерная криминалистика. Юрид. мир, 2007. 432 с.
- 2 Scott Chacon. Pro Git : professional version control. 2011. URL: <http://progit.org/ebook/progit.pdf>.
- 3 С.М. Львовский. Набор и вёрстка в системе L^AT_EX. МЦНМО, 2006. С. 448.
- 4 И. А. Чеботаев, П. З. Котельников. L^AT_EX 2_ε по-русски. Сибирский Хронограф, 2004. 489 с.
- 5 Qt Documentation [Электронный ресурс] // qt-project.org:[сайт]. 2013. URL: <http://qt-project.org/doc>.
- 6 Всё о кроссплатформенном программировании - Qt [Электронный ресурс] // doc.crossplatform.ru:[сайт]. 2013. URL: <http://doc.crossplatform.ru/qt>.
- 7 Справочник по XML-стандартам [Электронный ресурс] // msdn.microsoft.com:[сайт]. URL: [http://msdn.microsoft.com/ru-ru/library/ms256177\(v=vs.110\).aspx](http://msdn.microsoft.com/ru-ru/library/ms256177(v=vs.110).aspx).

Приложение А
(Обязательное)
Компакт-диск

Компакт-диск содержит:

- электронную версию пояснительной записи в форматах *.tex и *.pdf;
- актуальную версию программного комплекса для проведения компьютерной экспертизы;
- тестовые данные для работы с программным комплексом.