Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего

профессионального образования

«Московский государственный технический университет

имени Н.Э. Баумана»

(МГТУ им.Н.Э. Баумана)



Факультет «Информатика и системы управления»

Кафедра «Программное обеспечение ЭВМ и информационные технологии» (ИУ-7)

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе по предмету

«Вычислительные комплексы и сети»

на тему:

«Сравнение двух файлов по контрольной сумме»

Студент\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Мушиц С. Группа ИУ7-71)

Студент\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Переверзев В.А. Группа ИУ7-71)

Руководитель курсовой работы\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Алёшин В.А.)

Москва, 2013 г.

Введение

Цель данной работы – разработать программу, позволяющую просматривать и сравнивать хеш-суммы разных фалов на двух компьютерах.

Исследовательская часть

## Обзор предметной области

При копирование файлов с одного компьютера на другой, иногда возникает необходимость проверить тот ли файл был скопирован и скопирован ли он без ошибок. Такую проверку можно осуществить с помощью сравнения хеш-сумм файла. Хеш-сумму можно вычислить для любого файла и в зависимости от алгоритма количество коллизий можно свести к минимуму.

Такой функционал можно найти в таких программах как Total Commander или WinMerge. Это две наиболее известные программы которые могут предоставить необходимые средства для сравнения файлов на компьютере, но они могут сравнивать файлы на двух разных компьютерах.

## Требования к проекту

В соответствии с заданием на курсовую работу необходимо разработать программный продукт, ориентированный на семейство операционных систем Windows, и позволяющий сравнивать хеш-суммы выбранных файлов.

Требования к программному продукту:

1. Предоставить возможность сравнения выбранных двух хеш-сумм файлов на разных компьютерах;
2. Наличие понятного интерфейса пользователя;
3. Систему авторизации пользователя в системе;
4. Обмен данными происходит по средством протокола TCP/IP.

## Выводы

Путем анализа предметной области и существующих решений проблемы сформулированы требования к реализуемой программе по сравнению файлов.

Конструкторская часть

## Клиент-серверная архитектура

Традиционно взаимодействие с приложениями по сети организуется с использованием архитектуры типа “клиент-сервер”. Данный тип архитектуры можно рассмотреть на примере веб-сайта. При просмотре веб-сайта происходит отправка по интернету соответствующего запроса серверу, который затем возвращает необходимую информацию. Если надо загрузить какой-то файл, это делается напрямую с сервера.

На рисунке 1 представлен простой вариант архитектуры типа “клиент-сервер”.

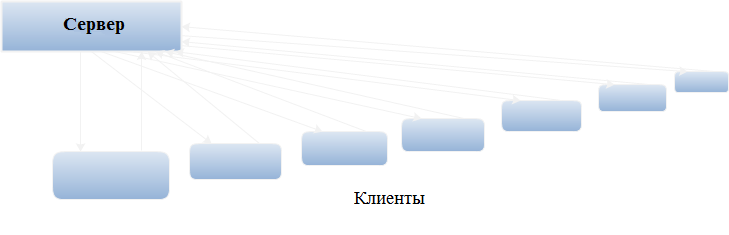


Рис. 1 Архитектура “клиент-сервер”.

Есть одна проблема данной архитектуры – это масштабируемость. При добавлении каждого нового клиента нагрузка на сервер, который должен взаимодействовать с каждым клиентом, будет увеличиваться (см. рис. 2.1).

Есть способы для улучшения работы такой архитектуры, на пример увеличить мощность сервера или устанавливать дублирующие сервера. Но эти способы затраты по времени и дороги.

## Пиринговая архитектура

Чтобы избежать проблему масштабируемости можно использовать другую архитектуру – peer-to-peer (p2p). Чаще всего в такой архитектуре нету четкого выраженного сервера, а каждый узел (peer) является как сервером, так и клиентом. Такая организация сети позволяет сохранять работоспособность сети при любом сочетании доступных узлов. Участники сети являются пиры (см. рис. 3).

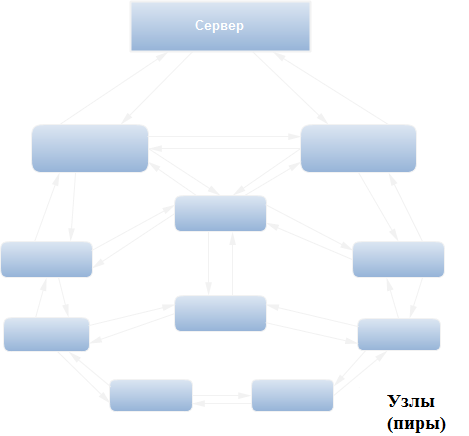


Рис. 3 Архитектура p2p.

Принцип работы такой сети в том, что узлы скачивают нужную информацию не на прямую с сервера, а с таких же узлов как они сами. Необходимый файл делится на куски и каждый кусок может быть скачан с разных узлов, что значительно повышает скорость скачивания, и главный сервер не будет перегружен запросами.

## Алгоритм хеширование

Для вычисление хеша файлов будет использовать 128 битный алгоритм MD5. Обычно представляться как последовательность из 32 шестнадцатеричных цифр. Пример хеша:

MD5("md5") = 1bc29b36f623ba82aaf6724fd3b16718

MD5 хорош тем, что даже небольшое изменение входного сообщения приводит к полному изменению хеша:

MD5("md4") = c93d3bf7a7c4afe94b64e30c2ce39f4f

## Описание протокола передачи данных

Для осуществления передачи данных между клиентом и сервером используется протокол, основанный на транспортном протоколе TCP/IP.

## Общий алгоритм работы программы

Работу программы можно описать следующим алгоритмом рисунке 4.

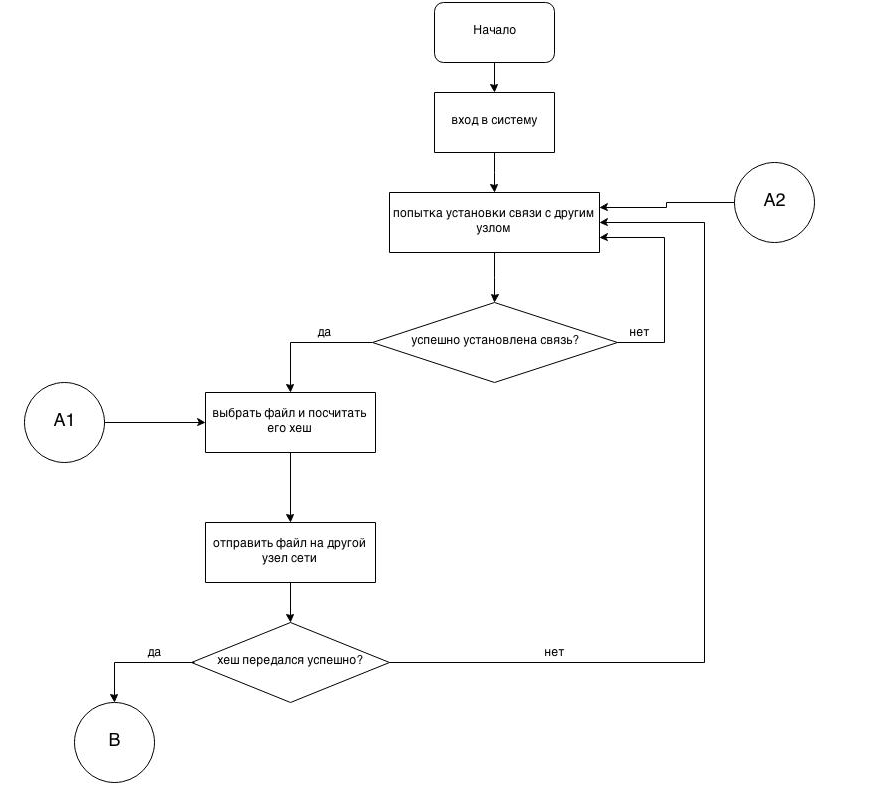


Рис.4 Алгоритм работы программы, часть 1

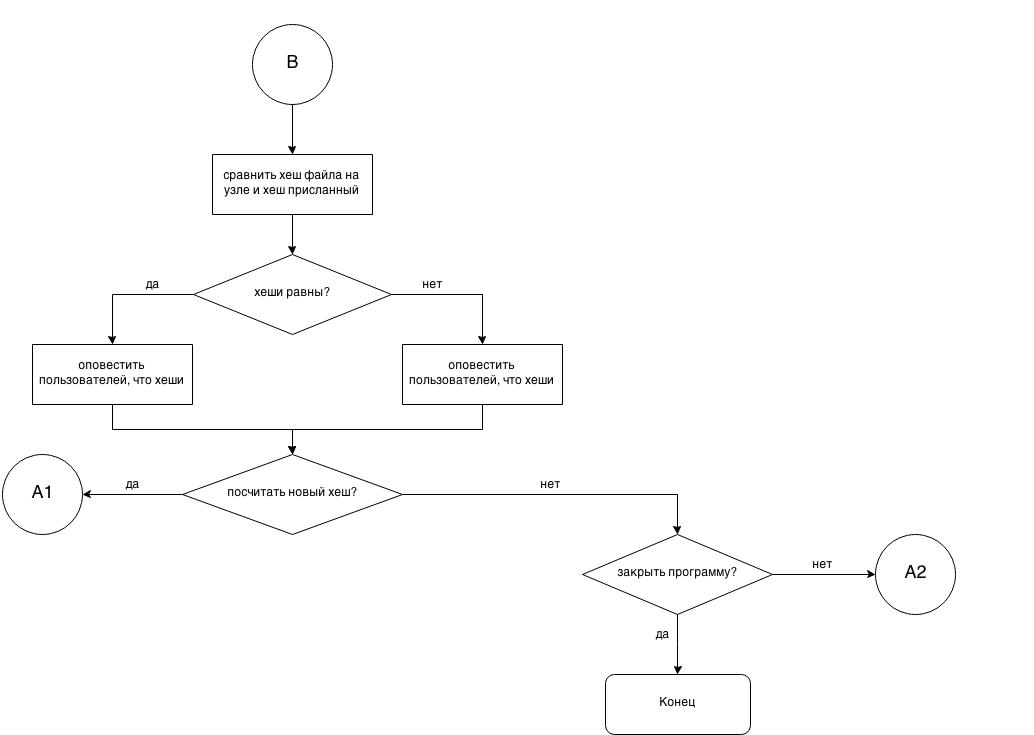


Рис.4 Алгоритм работы программы, часть 2

Технологический раздел

## Язык программирования

Для написания данной программы был выбран язык программирования C#. С# – объектно-ориентированный язык программирования, на котором можно писать приложения использую .NET Framework. Это позволяет использовать такие встроенные библиотеки как Windows Communication Foundation, Windows Presentation Foundation и многими другими, которые упрощают написание кода. При написании программ на языке C# разработчикам предоставляется автоматический сборщик мусора, который контролирует освобождение ресурсов и этим самым позволяет предотвратить утечку памяти. Язык C# позволяет обрабатывать исключение, перехватывая ошибки и обрабатывая их необходимым для разработчика образом, тем самым позволяя избавиться от ошибок реального времени, которые могут привести к закрытию приложения.

## Использование WPF

Для разработки интерфейса была выбрана технология WPF. Windows Presentation Foundation (WPF) — система для построения клиентских приложений Windows с визуально привлекательными возможностями взаимодействия с пользователем, графическая (презентационная) подсистема в составе .NET Framework (начиная с версии 3.0), использующая язык XAML.

WPF позволяет создавать интерфейс программы, сочетающий красочность информативность и удобство в использование.

## Использование WCF

Windows Communication Foundation(WCF) — программный фреймворк, используемый для обмена данными между приложениями, входящий в состав .NET Framework. Благодаря WCF есть возможность построения безопасных и надёжных транзакционных систем через упрощённую унифицированную программную модель межплатформенного взаимодействия. Комбинируя функциональность существующих технологий .NET по разработке распределённых приложений, WCF предоставляет единую инфраструктуру разработки, при умелом применении повышающую производительность и снижающую затраты на создание безопасных, надёжных и транзакционных Web-служб нового поколения.

## Использование NLog

NLog свободная платформа ведения журнала для платформы .NET с широким выбором инструментов и возможностей для гибкой настройки журналирования. NLog может фиксировать любые диагностические сообщения программы, дополнять их необходимой информацией (дата/время, процесс, имя пользователя, имя операции), отформатировать их в соответсвии с требованием пользователя и сохранить либо в БД, либо в файл.

## Скриншоты программы

Приложение разделено на две формы. Первая форма – форма аутентификации пользователя (рис. 5). Вторая форма позволяет пользователю выбирать любой файл на компьютере для вычисление хеша, и для пересылки хеша на другой компьютер (рис. 6).

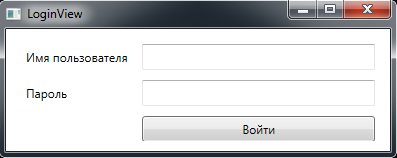


Рис.5 Форма аутентификации

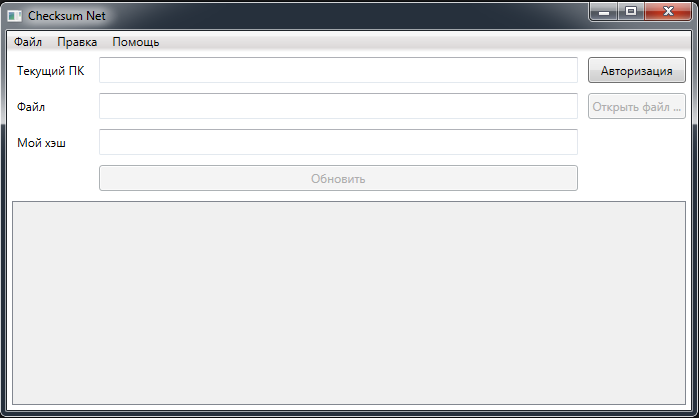


Рис.6 Основная форма программы

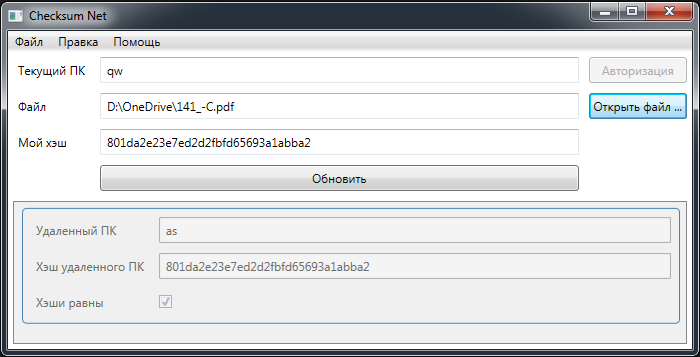


Рис.7 Скриншот работы программы

Выводы