Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Дисциплина: Операционные системы и системное программирование

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту

на тему

Утилита контроля появления дубликатов в файловой системе с заменой их на жесткие ссылки и протоколирования фактов замены

БГУИР КП 1–40 02 01 01 402 ПЗ

Студент: В.С.Моисеев

Руководитель: А.А.Калютчик

МИНСК 2024

Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет: ФКСиС. Кафедра: ЭВМ.

Специальность: 40 02 01 «Вычислительные машины, системы и сети».

Специализация: 40 02 01-01 «Проектирование и применение локальных компьютерных сетей».

УТВЕРЖДАЮ

Заведующий кафедрой ЭВМ

\_\_\_\_\_\_\_\_\_ Б.В.Никульшин

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2024 г.

ЗАДАНИЕ

по курсовому проекту студента

Моисеева Владислава Сергеевича

**1** Тема проекта: «Утилита контроля появления дубликатов в файловой системе с заменой их на жесткие ссылки и протоколирования фактов замены».

**2** Срок сдачи студентом законченного проекта: 20 мая 2024 г.

**3** Исходные данные к проекту:

**3.1** Язык программирования: С++.

**3.2** Среда разработки: Visual Studio Code.

**3.3** Операционная система: Linux (Ubuntu).

**4** Содержание пояснительной записки (перечень подлежащих разработке

вопросов):

Введение 1. Обзор литературы. 2. Системное проектирование.   
3. Функциональное проектирование. 4. Разработка программных модулей.   
5. Программа и методика испытаний. 6. Руководство пользователя. Заключение. Список использованных источников. Приложения.

**5** Перечень графического материала (с точным указанием обязательных

чертежей):

Схема программы.

КАЛЕНДАРНЫЙ ПЛАН

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование этапов  дипломного проекта | Объем  этапа,  % | Срок выполнения этапа | Примечания |
| Подбор и изучение литературы. Сравнение аналогов. | 10 | 23.03 – 05.04 |  |
| Структурное проектирование | 15 | 05.04 – 12.04 |  |
| Функциональное проектирование | 25 | 12.04 – 24.04 |  |
| Разработка программных модулей | 20 | 24.04 – 08.05 |  |
| Программа и методика испытаний | 10 | 8.05 – 15.05 |  |
| Оформление пояснительной записки | 15 | 20.05 – 30.05 |  |

Дата выдачи задания: 22.02.2021

Руководитель А. А. Калютчик

ЗАДАНИЕ ПРИНЯЛ К ИСПОЛНЕНИЮ \_\_\_\_\_\_\_\_

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 5](#_heading=h.3rdcrjn)

[1 ОБЗОР ЛИТЕРАТУРЫ 6](#_heading=h.2jxsxqh)

[1.1 Архитектура и принципы работы УКФС 6](#_heading=h.ai4fkinwi89z)

[1.2 Интерфейсы программирования приложений (API) для работы с УКФС 7](#_heading=h.bn8ykrhvvbq6)

[2. СИСТЕМНОЕ ПРОЕКТИРОВАНИЕ 9](#_heading=h.owqn6d2bx4ju)

[2.1 Структура данных 9](#_heading=h.7sif9i3vzb0m)

[2.2 Алгоритмы работы 9](#_heading=h.j39qo37ipu7a)

[2.3 Поддерживаемые операции 9](#_heading=h.vwdqoi4w3qqu)

[2.4 Выбор языка программирования 10](#_heading=h.qdudp9cbyol3)

[2.5 Инструменты разработки 10](#_heading=h.hez91fa3aapo)

# ВВЕДЕНИЕ

В современном мире компьютеры прочно вошли в нашу жизнь. Мы используем их для работы, учебы, развлечений и хранения информации. Одним из важнейших компонентов любой компьютерной системы является утилита контроля файловой системы. Она отвечает за организацию хранения данных на диске, предоставляя пользователю удобный интерфейс для доступа к ним.

В рамках данного курсового проекта будет рассмотрена утилита контроля появления дубликатов в файловой системе с заменой их на жесткие ссылки и протоколирования фактов замены, предназначенная для работы в пространстве пользователя. Утилита будет реализована на базе операционной системы Linux.

**Целью** проекта является разработка и реализация утилиты контроля появления дубликатов в файловой системе с заменой их на жесткие ссылки, обладающей следующими характеристиками:

* Поиск дубликатов и сравнение содержимого: Это основные функции утилиты, которые позволяют определить файлы с одинаковым содержимым и принять решение о замене их на жесткие ссылки;
* Замена на жесткие ссылки: Это ключевая характеристика, поскольку замена дубликатов на жесткие ссылки позволяет сэкономить место на диске, избегая создания лишних копий файлов;
* Протоколирование: Протоколирование фактов замены не только обеспечивает отслеживание выполненных операций, но и помогает в дальнейшем анализе файловой системы и выявлении паттернов поведения;
* Эффективность и производительность: Утилита должна быть эффективной как в потреблении ресурсов, так и в скорости выполнения, особенно для больших файловых систем. Это обеспечивает удобство использования и экономию времени пользователей.

**Актуальность** темы обуславливается тем, что утилиты контроля файловых систем являются неотъемлемой частью любой компьютерной системы. Разработка и изучение утилиты контроля появления дубликатов в файловой системе позволит глубже понять принципы работы утилит контроля файловых систем, а также приобрести навыки программирования в операционной системе Linux.

# 1 ОБЗОР ЛИТЕРАТУРЫ

Утилита контроля файловой системы — это программное средство, которое предназначено для мониторинга и обнаружения изменений в файловой системе компьютера или сервера. Её основная задача состоит в том, чтобы обеспечить целостность и безопасность файловой системы путём выявления несанкционированных изменений или повреждений файлов и структур каталогов.

## 1.1 Архитектура и принципы работы УКФС

В данном обзоре рассматриваются архитектура и принципы работы трех популярных УКФС:

### 1.1.1 fdupes

fdupes - это утилита командной строки для операционных систем Linux и Unix, предназначенная для поиска и управления дубликатами файлов в файловой системе. Она позволяет сканировать указанные каталоги и сравнивать содержимое файлов, выявляя дубликаты на основе их хеш-сумм или других атрибутов. После обнаружения дубликатов fdupes предоставляет опции для управления ими, включая удаление, перемещение или замену на жесткие ссылки. Это позволяет сократить объем занимаемого дискового пространства, поскольку жесткая ссылка не создает дополнительных копий файлов, а лишь указывает на существующий экземпляр. Fdupes обладает гибкими возможностями настройки, позволяя пользователю выбирать критерии сравнения файлов и определять режимы работы утилиты в зависимости от конкретных потребностей. Она является популярным инструментом среди администраторов и пользователей Linux и Unix для оптимизации использования дискового пространства и поддержания порядка в файловой системе.

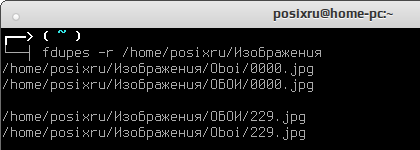


Рисунок 1.1 - Иллюстрация структуры fdupes

### 1.1.2 Duplicate Cleaner

Duplicate Cleaner - это мощная утилита для операционной системы Windows, разработанная для поиска и удаления дубликатов файлов на компьютере. Она обеспечивает пользователей возможностью осуществлять более эффективное управление файлами, оптимизируя использование дискового пространства и улучшая организацию данных.

Одной из ключевых особенностей Duplicate Cleaner является его способность находить дубликаты файлов не только на основе их имен, но и на основе содержимого. Это позволяет обнаруживать файлы с разными именами, но с одинаковым содержимым, что является частым случаем при копировании или сохранении файлов на диске.

Утилита предлагает гибкие опции поиска, включая возможность настройки критериев сравнения, таких как размер файлов, типы файлов, диапазон дат изменения и другие параметры. После завершения поиска Duplicate Cleaner предоставляет пользователю список обнаруженных дубликатов, с возможностью просмотра информации о каждом файле.

Одним из важных функций утилиты является возможность замены дубликатов на жесткие ссылки. Пользователь может выбрать, какие дубликаты заменять, а какие удалять или перемещать. Это позволяет существенно сэкономить место на диске, не создавая лишних копий файлов.

Duplicate Cleaner также обладает удобным интерфейсом, который делает процесс поиска и управления дубликатами файлов интуитивно понятным и простым в использовании. Благодаря своим функциям и возможностям настройки, утилита стала популярным выбором среди пользователей Windows для оптимизации и организации своих файлов.

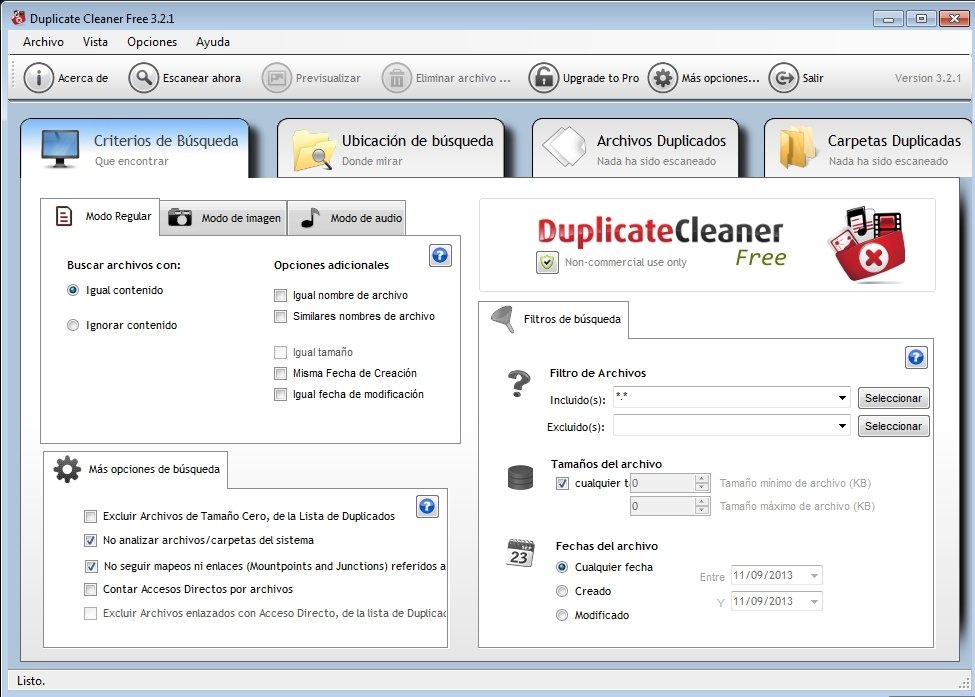


Рисунок 1.4 - Иллюстрация структуры Duplicate Cleaner

### 1.1.3 AllDup

AllDup - это бесплатная утилита для операционной системы Windows, предназначенная для поиска и удаления дубликатов файлов. Она обладает широким спектром функций, которые делают процесс поиска и управления дубликатами удобным и эффективным. Одной из ключевых особенностей AllDup является его способность проводить поиск дубликатов на основе различных критериев, таких как имя файла, размер, дата изменения, содержимое и другие характеристики. Это позволяет обнаруживать как точные дубликаты файлов, так и файлы с схожим содержимым.

Утилита предоставляет гибкие возможности настройки параметров поиска, позволяя пользователю определить критерии сравнения и исключить из поиска определенные типы файлов или директории. После завершения поиска AllDup предоставляет подробный отчет о найденных дубликатах, который включает информацию о размере файлов, их местоположении и другие атрибуты.

Одной из привлекательных возможностей AllDup является его способность заменять дубликаты файлов на жесткие ссылки. Это позволяет сократить объем используемого дискового пространства, не удаляя лишние копии файлов, а просто заменяя их на ссылки, указывающие на один и тот же файл. Пользователь имеет возможность выбирать, какие дубликаты заменять, а какие оставлять или удалить.

AllDup обладает интуитивно понятным и удобным интерфейсом, что делает процесс поиска и удаления дубликатов файлов простым и доступным даже для неопытных пользователей. Благодаря своим функциям и простоте использования, AllDup является популярным выбором среди пользователей Windows для оптимизации и организации своих файлов.

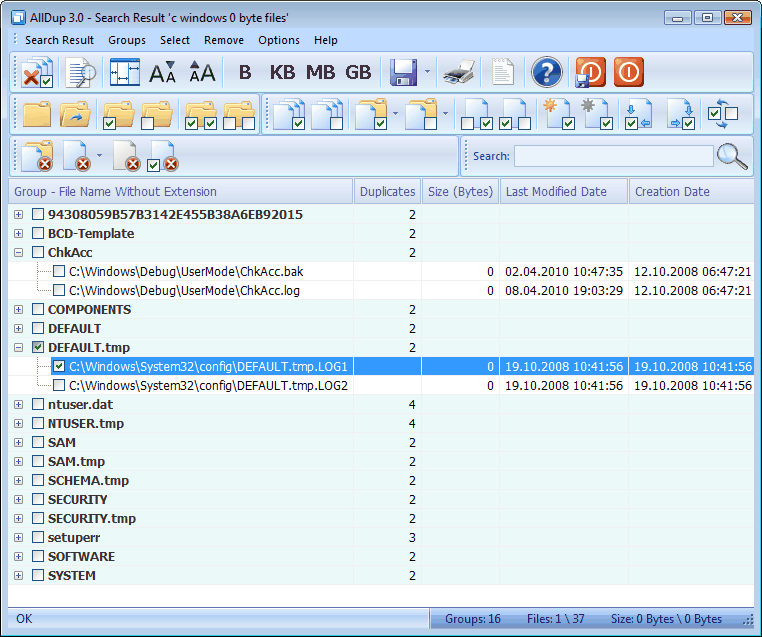


Рисунок 1.5 - Иллюстрация структуры ALLDup

## 1.2 Интерфейсы программирования приложений (API) для работы с УКФС

Утилита контроля появления дубликатов в файловой системе API для контроля появления дубликатов: предоставляет методы для обнаружения и управления дубликатами файлов.

Поиск, замена, протоколирование: Основные операции с дубликатами файлов. Переносимость кода: Возможность работы на разных операционных системах. Функции доступа к файловой системе: Интеграция с системными вызовами для работы с файлами и директориями.

Гибкость настроек: Параметры поиска и замены дубликатов, а также форматы журналирования, могут быть настроены в соответствии с требованиями пользователя.

Эффективность и производительность: Оптимизированные алгоритмы поиска и замены, чтобы обеспечить быструю и эффективную обработку больших объемов данных.

# 2. СИСТЕМНОЕ ПРОЕКТИРОВАНИЕ

Данный раздел посвящен системному проектированию утилиты контроля появления дубликатов в файловой системе с заменой их на жесткие ссылки и протоколирования фактов замены. Утилита контроля появления дубликатов в файловой системе с заменой на жесткие ссылки и протоколированием фактов замены - это программа, которая сканирует файловую систему, выявляет дубликаты файлов, заменяет их на жесткие ссылки для экономии места на диске и записывает информацию о замене в журнал.

## 2.1 Структура данных

* Журнал замен: Это файл или база данных, в которой записывается информация о каждой замене дубликата на жесткую ссылку. Каждая запись в журнале может содержать следующую информацию: путь к исходному файлу дубликата, путь к оригинальному файлу, на который была создана жесткая ссылка, дата и время замены.
* Структура для хранения информации о файлах и дубликатах: массив или список файлов, включая их атрибуты (например, путь, имя, размер, хеш-сумма и т. д.), для каждого файла может быть отдельная структура, которая содержит информацию о его дубликатах.
* Структура для представления жестких ссылок: массив или список жестких ссылок, каждая из которых содержит информацию о своем пути и целевом файле.
* Дерево файловой системы: структура данных, которая отражает иерархию файлов и директорий в файловой системе. Это может быть представлено в виде древовидной структуры, где каждый узел представляет файл или директорию, а дочерние узлы - их содержимое.
* Структуры для операций поиска и замены: для удобства реализации алгоритмов поиска дубликатов и замены на жесткие ссылки могут использоваться различные структуры данных, такие как хеш-таблицы, деревья поиска или очереди.
* Структура для конфигурации и параметров утилиты: Структура данных, которая содержит параметры, указанные пользователем для настройки поведения утилиты, например, пути к директориям для сканирования, условия сравнения файлов и т. д.

## 2.2 Алгоритмы работы

Утилита контроля появления дубликатов в файловой системе с заменой их на жесткие ссылки и протоколирования фактов замены реализует следующие алгоритмы:

* Сканирование файловой системы: Проход по всей файловой системе для поиска файлов и составления списка файлов с их атрибутами (путь, имя, размер, хеш-сумма и т. д.).
* Выявление дубликатов: сравнение файлов по их атрибутам или содержимому для выявления дубликатов. При обнаружении дубликатов файлов запись информации о них в структуру данных или массив.
* Замена дубликатов на жесткие ссылки: для каждого обнаруженного дубликата: создание жесткой ссылки на оригинальный файл. Удаление дубликата или изменение его имени/расположения (по выбору пользователя). Запись информации о замене в журнал.
* Протоколирование фактов замены: Запись информации о каждой замене в журнал, включая путь к исходному файлу дубликата, путь к оригинальному файлу и дату/время замены.
* Операции завершения: отчет об успешном завершении операции сканирования и замены. Возможность отмены операции или повторного запуска с другими параметрами.

## 2.3 Поддерживаемые операции

Утилита контроля появления дубликатов в файловой системе с заменой их на жесткие ссылки и протоколирования фактов замены поддерживает следующие операции:

* Сканирование файловой системы.
* Выявление дубликатов.
* Замена на жесткие ссылки.
* Протоколирование замены.
* Управление параметрами и настройками.

## 2.4 Выбор языка программирования

C был выбран в качестве языка программирования утилиты контроля появления дубликатов в файловой системе по ряду причин:

* Производительность: C обеспечивает высокую производительность, что важно для системных программ.
* Переносимость: C является переносимым языком, что позволяет SFS работать на разных платформах.
* Удобство разработки: C обладает удобными инструментами и библиотеками для разработки системного ПО.
* Поддержка: C имеет обширную документацию и сообщество разработчиков.

## 2.5 Инструменты разработки

Для разработки SFS будут использоваться следующие инструменты:

* Компилятор: GCC
* Отладчик: GDB