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 ДЛЯ ФОРМАТИРОВАНИЯ И ПРОВЕРКИ SFS

БГУИР КП 1-40 02 01 307 ПЗ

Студент: гр. 350503

Ганецкий В. В.

Руководитель:

Протько М. А.

Минск 2025

Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет

информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Дисциплина: Программирование на языках высокого уровня

УТВЕРЖДАЮ

Заведующий кафедрой ЭВМ

\_\_\_\_\_\_\_\_\_ Б. В. Никульшин

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2025 г.

ЗАДАНИЕ

по курсовой работе студента

Ганецкого Владимира Витальевича

1 Тема работы: «Утилита для форматирования и проверки SFS»

2 Срок сдачи студентом законченной работы: 31 марта 2025 г

3 Исходные данные к работе: язык программирования C, библиотека ncurses

4 Содержание пояснительной записки (перечень подлежащих разработке вопросов):

Введение. 1. Обзор литературы. 2. Функциональное проектирование. 3. Разработка программных модулей. 4. Результаты работы программы. 5. Руководство пользователя. Заключение. Список использованных источников.

5 Перечень графического материала (с точным обозначением обязательных чертежей и графиков)

5.1 Диаграмма классов

5.2 Схема алгоритма метода

5.3 Схема алгоритма метода

КАЛЕНДАРНЫЙ ПЛАН

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование этапов курсовой работы | Объем этапа, % | Срок выполнения этапа | Примечания |
| Выбор задания. Разработка содержания пояснительной записки. Перечень графического материала | 25 | 15.09 – 01.10 |  |
| Обзор литературы. Функциональное проектирование | 20 | 02.10 – 01.11 | С выполнением чертежа |
| Разработка программных модулей. Тестирование программы. Заключение | 40 | 02.11 – 01.12 | С выполнением чертежа |
| Список использованных источников. Приложения. Оформление пояснительной записки и графического материала | 15 | 02.12 – 11.12 |  |

Дата выдачи задания: 24 февраля 2025 г.

Руководитель Протько М. А.

Задание принял к исполнению–––––––\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_–– Ганецкий В. В.

# **ВВЕДЕНИЕ**

Сейчас на любом устройстве, будь то компьютер или телефон, есть файловая система. Файловая система представляет собой систему организации и хранения данных на устройствах, которая позволяет эффективно использовать пространство носителя информации. Она отвечает за то, как данные размещаются в заданных им местах, как они будут распределяться по объему памяти, сколько дополнительной информации они будут содержать для их однозначной классификации и структуризации. Это позволяет операционной системе выгодно использовать пространство памяти и уменьшать нагрузку на рабочие элементы. Кроме того, неоспоримую пользу файловая система приносит для пользователя, предоставляя ему интерфейс для взаимодействия с информацией. Благодаря визуализации всех объектов, хранимых в устройстве, а также оформлении их в виде каталогов, пользователь может с легкостью находить файлы и различать их между собой, не вдаваясь в подробности их предназначения. Это существенно сокращает время на поиск информации и улучшает пользовательский опыт. Также файловая система может предоставлять расширенный функционал для работы с файлами, такой как разграничение доступа или шифрование файлов. Наличие такого рода функций позволяет улучшить безопасность данных и персонализировать представление объектов файловой системы.

С учетом сложности файловых систем, для каждой из них требуется утилита, позволяющая проверять корректность размещенных данных, их целостность и изменять структуру носителя информации, то есть осуществлять форматирование. Форматирование позволяет быстро очищать носитель от ненужной информации, давая возможность заново заполнять чистое пространство новыми данными, а также помогает вернуть структуру в рабочее состояние, если по какой-либо причине в файловой системе произошел сбой и работоспособность была нарушена. Также некоторые утилиты в таком случае позволяют не только отформатировать носитель информации, но и восстановить утраченные или поврежденные данные. Тема данного курсового проекта посвящена разработке именно такой утилите, которая позволит расширить работу SFS (simple file system) и защитить ее от сбоев.

# **1 ОБЗОР ЛИТЕРАТУРЫ**

Для разработки данной утилиты была выбрана библиотека ncurses, предназначенная для управления вводом-выводом на терминал. Она упрощает работу, позволяя использовать готовые функции для создания консольного интерфейса, что дает возможность уделить больше времени на реализацию возможностей утилиты файловой системы. Более того, данная библиотека предоставляет преимущество в виде написания переносимого кода, так как ncurses сама управляет взаимодействием с аппаратными различиями терминалов.

В качестве языка программирования для реализации программного обеспечения был выбран C. Такой выбор обусловлен распространенностью языка в сфере написания операционных систем как средства, которое позволяет писать высокопроизводительный код с удобным уровнем абстракции, позволяющий эффективно создавать требуемый функционал, и при этом не требующее значительного потребления системных ресурсов. С точки зрения темы проекта, это позволяет написать утилиту, быстро работающую с файловой системой и обладающей широким функционалом для взаимодействия с ней.

# **1.1 Анализ существующих аналогов**

Учитывая большой потенциал для разработки мощного программного обеспечения, следует изучить имеющиеся аналоги, составить представление об их возможностях, недостатках, преимуществах и сформировать план по структуре собственной утилиты.

# **TestDisk**

TestDisk позволяет восстанавливать загрузочные разделы, удаленные разделы, фиксировать поврежденные таблицы разделов и восстанавливать данные, а также создавать копии файлов с удаленных/недоступных разделов. При запуске приложения предоставляется список разделов жесткого диска, с которыми можно работать. Выбор доступных действий, осуществляемых в разделах, включает: анализ для корректировки структуры (и последующее восстановление, в случае обнаружения проблемы); изменение дисковой геометрии; удаление всех данных в таблице разделов; восстановление загрузочного раздела; перечисление и копирование файлов; восстановление удаленных файлов; создание снапшота раздела.

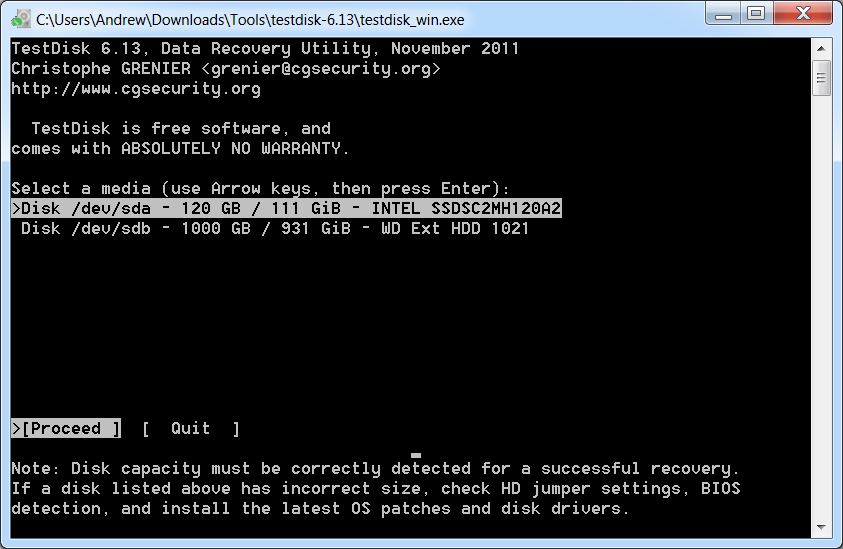


Рисунок 1.1 – скриншот программы TestDisk

# **EaseUS Partition Master**

EaseUS Partition Master – инструмент для работы с разделами жесткого диска. Он позволяет создавать, перемещать, объединять, разделять, форматировать, изменяя их размер и расположение без потери данных. Также помогает восстанавливать удаленные или потерянные данные, проверять разделы, перемещать ОС на другой HDD/SSD и осуществлять другой функционал.

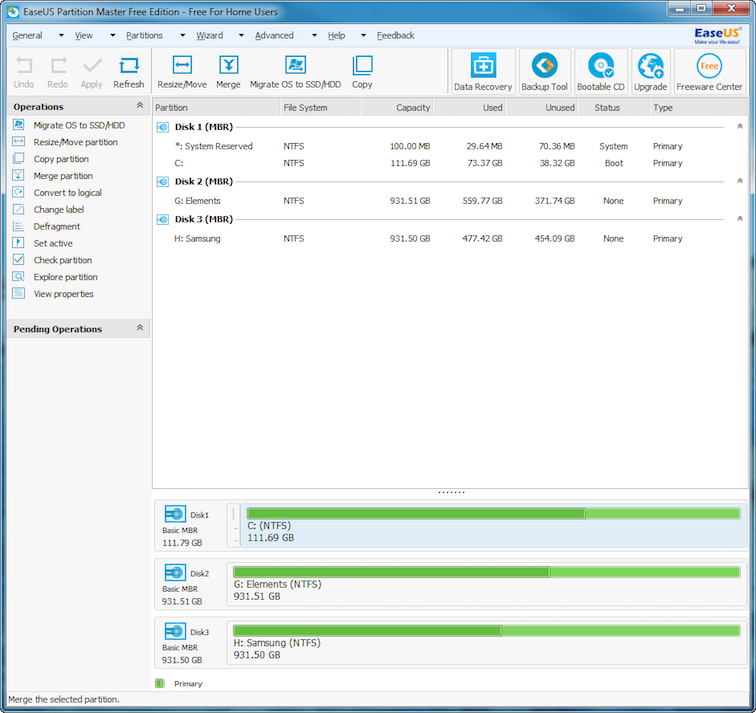


Рисунок 1.2 – скриншот программы EaseUS Partition Master

# **GParted**

GParted – редактор дисковых разделов с открытым исходным кодом. Осуществляет эффективное и безопасное управление разделами (создание, удаление, изменение размера, перемещение, копирование, проверка) без потери данных. Данное приложение позволяет создавать таблицы разделов (MS-DOS или GPT), включать, отключать и изменять атрибуты, выравнивать разделы, восстанавливать данные с поврежденных разделов и выполнять другие важные функции.

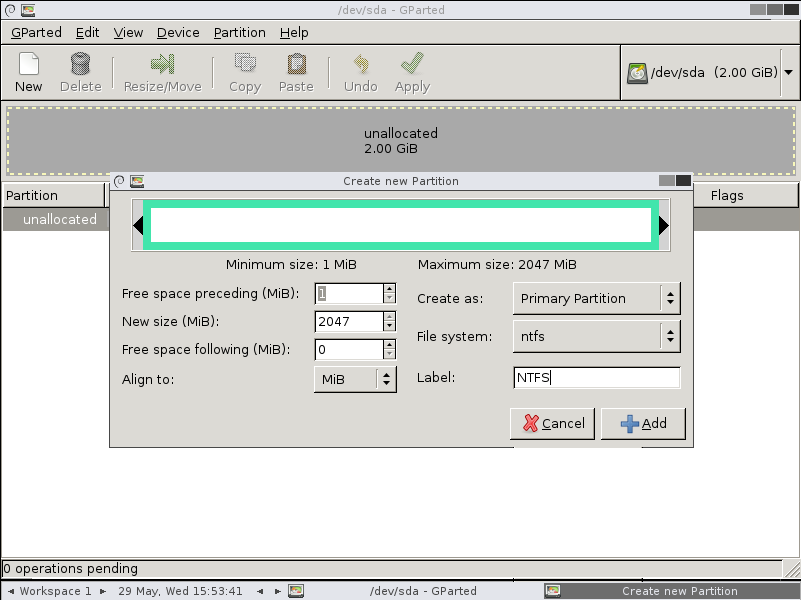


Рисунок 1.3 – скриншот программы GParted

# **1.2 Постановка задачи**

По итогу рассмотрения аналогов можно сказать, что все они обладают разнообразным функционалом, а также большая часть из них имеет полноценный пользовательских интерфейс, что упрощает работу с данными программу. В рамках курсового проекта и выбранной темы реализации подобных функций и внешнего вида будет исчерпывающим, поэтому для реализации программного обеспечения были установлены следующие опорные пункты:

- Программа должна иметь минимальный пользовательских интерфейс для обеспечения удобного взаимодействия с приложением

- Программа должна предоставлять следующие функции: форматирование носителя, проверка файловой системы, шифрование файлов, журналирование, восстановление повреждений

- Программа должна иметь документацию по принципу работы файловой системы (SFS) и самой утилиты

# **2 СИСТЕМНОЕ ПРОЕКТИРОВАНИЕ**

После определения требований к функционалу разрабатываемого приложения его следует разбить на функциональные блоки. Такой подход упростит понимание проекта, позволит устранить проблемы в архитектуре, обеспечит гибкость и масштабируемость программного продукта в будущем путем добавления новых блоков.