Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Отчёт

по учебной (ознакомительной) практике

Студент:

гр. 250501 Хворостова А. Ю.

Руководитель:

старший преподаватель кафедры ЭВМ

Ковальчук А. М.

МИНСК 2023

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_gjdgxs)

[1 ПОСТАНОВКА ЗАДАЧИ 4](#_30j0zll)

[2 ОБЗОР ЛИТЕРАТУРЫ 5](#_1fob9te)

[3 ФУНКЦИОНАЛЬНОЕ ПРОЕКТИРОВАНИЕ 6](#_tyjcwt)

[4 РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ](#_4d34og8) 7

[4.1 Разработка схем алгоритмов](#_2s8eyo1) 7

[4.2 Разработка алгоритмов](#_17dp8vu) 7

[5 РЕЗУЛЬТАТ РАБОТЫ](#_3rdcrjn) 9

[ЗАКЛЮЧЕНИЕ](#_26in1rg) 11

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ](#_lnxbz9) 12

[ПРИЛОЖЕНИЕ А](#_1ksv4uv) 13

[ПРИЛОЖЕНИЕ Б](#_44sinio) 14

[ПРИЛОЖЕНИЕ В](#_2jxsxqh) 15

[ПРИЛОЖЕНИЕ Г](#_z337ya) 16

[ПРИЛОЖЕНИЕ Д](#_3j2qqm3) 17

# ВВЕДЕНИЕ

Язык программирования С был создан в 1972 году Деннисом Ритчи в Bell Labs. Он был разработан как язык системного программирования для Unix, который был в то время распространенной операционной системой. В настоящее время С широко используется во многих областях, включая разработку операционных систем, банковских систем, игр, автомобильных систем, веб-приложений и многих других.

Один из главных принципов С - это близость к машинному коду, что позволяет писать быстрый и эффективный код. Это делает язык С популярным выбором для решения задач, где требуется высокая производительность и низкий уровень абстракции, таких как разработка драйверов, компиляторов и операционных систем. Также язык С очень удобен для решения задач связанных с численными методами, например, им можно эффективно решить сложные математические задачи и моделирование систем.

Практическое применение языка С включает в себя разработку встроенных систем, микроконтроллеров, OpenGL приложений, Unix программ, научных программ и многих других. Серьезный подход к изучению языка С и его техническим аспектам, позволяет программистам создавать мощные инструменты, которые способны выдержать тяжелые нагрузки и решить самые сложные задачи.

В современном мире, где все становится всё более автоматизированным, программируемые приложения играют большую роль в нашей повседневной жизни. Файловый менеджер на языке С - это одно из таких приложений, которое позволяет работать с файлами на компьютере пользователя.

С предназначен для системного программирования и имеет доступ к низкоуровневым функциям операционной системы, таким как работа с файлами и директориями. Файловый менеджер - это программа, которая работает с файловой системой компьютера и позволяет пользователю управлять файлами и директориями. Язык С предоставляет все необходимые инструменты для реализации такой программы, включая функции для работы с файловой системой, пользовательским интерфейсом, обработки ошибок и т.д.

Важность данной работы обусловлена тем, что файловый менеджер может значительно облегчить организацию хранения данных, обеспечивая удобный и быстрый доступ к содержимому файлов. Разработка эффективного алгоритма для создания такой программы является ключевым фактором для обеспечения её надежной и стабильной работы.

1. **ПОСТАНОВКА ЗАДАЧИ**

Файловые менеджеры являются неотъемлемой частью операционных систем и служат для управления файлами и папками на компьютере. Они решают ряд задач, которые связаны с организацией и управлением файловой системы.

В первую очередь файловые менеджеры позволяют пользователю просматривать содержимое каталогов, открывать файлы и папки, копировать, перемещать и удалять их. Они также предоставляют пользователю возможность создавать новые файлы и папки, переименовывать их и изменять атрибуты файлов.

Необходимо разработать алгоритмы хранения данных, обеспечивая удобство пользования, поддержку различных форматов файлов, надежность хранения, возможность изменения данных.

1. **ОБЗОР ЛИТЕРАТУРЫ**

Поставленной задачей является реализация удобного интерфейса для работы с данными. Это включает в себя базовые функции работы с файлами для корректной организации данных: создание нового файла, копирование информации из файла в новый, поиск файлов с определенным расширением, вывод содержимого директории, вывод содержимого файла, удаление файла. Все функции возможны как для текстовых, так и для бинарных. Этот набор функций удовлетворяет всем возможным задачам организации информации.

Для решения задачи был выбран язык программирования С.

Важным практическим применением языка С является работа с файлами. В С файлы могут быть открыты, прочитаны, записаны, закрыты и переименованы. Это делает С идеальным выбором для разработки программ, которые работают с файлами, таких как текстовые редакторы, базы данных и системы управления версиями. Благодаря языку С разработчики могут создавать программы с высокой скоростью и производительностью обработки данных в файловой системе.

В программе осуществляется обработка исключительных ситуаций – механизм, предназначенный для описания реакции программы на ошибки, которые могут возникнуть при выполнении программы и приводят к невозможности (бессмысленности) дальнейшей обработки программой её базового алгоритма

1. **ФУНКЦИОНАЛЬНОЕ ПРОЕКТИРОВАНИЕ**

Входными данными являются начальная рабочая директория, в которой пользователь может исполнять некоторые функции, введенная пользователем команда и аргументы к ней.

Выходными данными являются измененные файлы/директории.

# РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ

**4.1 Разработка схем алгоритмов**

Схема алгоритма программы main.c представлена в приложении А.

Функция copyTextFile() копирует текстовый файл по заданной директории. Схема алгоритма представлена в приложении Б.

**4.2 Разработка алгоритмов**

Функция copyTextFile копирует файл по указанной директории

Шаг 1. Начало.

Шаг 2. Объявление переменных:

char \*\*fileINFO - указатель на массив строк

FILE \*dest - указатель на файл, в который будем копировать информацию

int lines - количество строк в файле

char destFilename[FILENAME] - имя файла, в который будем копировать информацию

Шаг 3. Проверка, что файл содержит информацию. Если нет, выводим сообщение об ошибке и выходим из функции.

Шаг 4. Открытие файла destFilename на запись.

Шаг 5. Проверка, что файл dest был успешно открыт. Если нет, вывод сообщения об ошибке и выход из функции.

Шаг 6. Цикл параметра i по всем строкам в fileINFO:

Шаг 7. Запись i-ой строки из fileINFO в файл dest с использованием функции fwrite.

Шаг 8. Закрытие файла dest.

Шаг 9. Конец функции.

Шаг 10. Конец

Функция findBinFiles возвращает массив названий бинарных файлов.

Передаваемые параметры:

const char \*dir\_path - директория

int \*count - указатель на количество файлов

char \*path - путь к файлу

Шаг 1. Начало.

Шаг 2. Объявление переменных:

char \*\*found - массив названий файлов

DIR \*dir - директория

struct dirent \*entry - доступ к содержимому директории

Шаг 3. Вызов readdir - чтение из директории

Шаг 4. Получить расширение const char \*extension файла из имени

Шаг 5. Если extension = .bin, перейти к шагу 6, иначе переход к шагу 11

Шаг 6. Увеличить значение \*count на 1

Шаг 7. Перевыделить память на fount на (\*count)\*sizeof(char\*) байт

Шаг 8. Выделить память на path

Шаг 9. В path записать абсолютный путь

Шаг 10. Записать в массив имен found путь path

Шаг 11.Переход на шаг 3.

Шаг 12. Закрыть директорию dir.

Шаг 13. Вернуть found.

Шаг 14. Конец.

1. **РЕЗУЛЬТАТ РАБОТЫ**

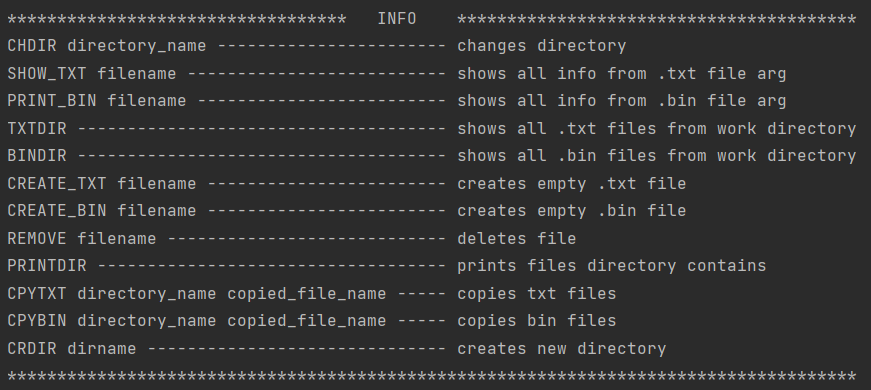


Рисунок 5.1 – Главное меню программы



Рисунок 5.2 – Пример работы функции print\_directory\_contents



Рисунок 5.3 – Пример работы функции createDirectory

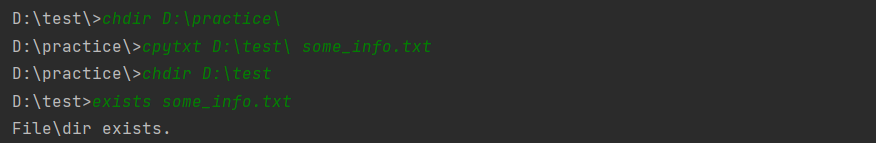


Рисунок 5.4 – Пример работы функции copyTextFile

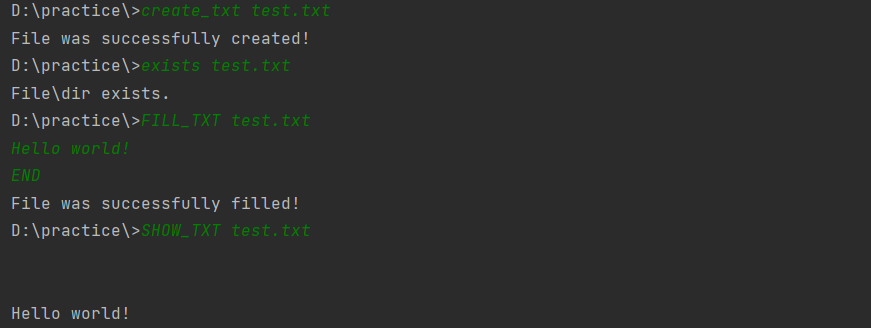


Рисунок 5.5 – Пример работы функций createTextFile и fillTextFile

**ЗАКЛЮЧЕНИЕ**

В данной работе был разработан эффективный файловый менеджер с возможностью обработки файлов в различных форматах на языке С. Разработанный алгоритм позволяет точно и быстро работать с файлами, а также производить различные операции с ними, такие как копирование, перемещение, создание и удаление файлов и папок.

Разработанный менеджер обеспечивает удобное и эргономичное обращение пользователей с файловой системой, что делает его привлекательным и функциональным инструментом для работы с файлами.

Выполнение данной учебной практики позволило не только улучшить навыки работы с файловой системой, но и понять принципы проектирования и разработки программного обеспечения для обработки файлов. Полученные знания и опыт могут быть использованы в дальнейшей работе над созданием и развитием программного обеспечения для обработки и хранения файлов.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Луцик, Ю. А. Основы алгоритмизации и программирования : язык Си : учебно-метод. пособие / Ю. А. Луцик, А. М. Ковальчук, Е. А. Сасин. – Минск : БГУИР, 2015. – 170с. : ил.

2. Основы алгоритмизации и программирования : лаборатор. практикум для студентов специальности 1-40 02 01 «Вычисл. машины, системы и сети» всех форм обучения. В 2 ч. Ч. 2 / сост. Ю. А. Луцик [и др.]. – Минск : БГУИР, 2010. – 36 с. : ил.

3.Википедия [Электронный ресурс]. - Режим доступа: https://docs.google.com/document/d/10\_zHuwyQXVOeqc58\_I6FMTv-mIu9aMaVw6xmHe5HmLk/edit?usp=sharing

4. Демидович, Е. М. Основы алгоритмизации и программирования. Язык Си / Е. М. Демидович. – СПб. : БХВ-Петербург, 2008. – 440 с.

5. Стивен Прата Язык программирования С. Лекции и упражнения, 6-е изд. : Пер. с англ. —М. :ООО “И.Д. Вильямс”, 2015. — 928 с

**ПРИЛОЖЕНИЕ А**

*(обязательное)*

Схема алгоритма программы main.c

**ПРИЛОЖЕНИЕ Б**

*(обязательное)*

Схема алгоритма функции copyTextFile()

**ПРИЛОЖЕНИЕ В**

*(обязательное)*

Код программы