

Инженерная школа информационных технологий и робототехники  
 Направление подготовки 15.03.04 «Автоматизация технологических процессов и производств»  
 Отделение автоматизации и робототехники

### БАКАЛАВРСКАЯ РАБОТА

Тема работы
Автоматизация распознавания типов файловых систем по их сигнатурам, при их частичном повреждении.

УДК 004.651:004.93

Студент

Группа	ФИО	Подпись	Дата
158Т61	Ван Лянь		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОАР ИШИТР	Аврамчук Валерий Степанович	к.т.н., доцент		

### КОНСУЛЬТАНТЫ ПО РАЗДЕЛАМ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОСГН УОД	Конотопский Владимир Юрьевич	к.т.н., доцент		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Профессор ОКД, ИШНКБ	Сечин Александр Иванович	д.т.н., профессор		

### ДОПУСТИТЬ К ЗАЩИТЕ:

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОАР ИШИТР	Громаков Евгений Иванович	к.т.н., доцент		

## ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ООП

По направлению подготовки 15.03.04 Автоматизация технологических процессов и производств

Код результата	Результат обучения (выпускник должен быть готов)
<i>Профессиональные компетенции</i>	
P1	Демонстрировать базовые естественнонаучные и математические знания для решения научных и инженерных задач в области анализа, синтеза, проектирования, производства и эксплуатации систем автоматизации технологических процессов и производств. Уметь сочетать теорию, практику и методы для решения инженерных задач, и понимать область их применения
P2	Иметь осведомленность о передовом отечественном и зарубежном опыте в области теории, проектирования, производства и эксплуатации систем автоматизации технологических процессов и производств.
P3	Применять полученные знания для определения, формулирования и решения инженерных задач при разработке, производстве и эксплуатации современных систем автоматизации технологических процессов и производств с использованием передовых научно–технических знаний и достижений мирового уровня, современных инструментальных и программных средств.
P4	Уметь выбирать и применять соответствующие аналитические методы и методы проектирования систем автоматизации технологических процессов и обосновывать экономическую целесообразность решений.
P5	Уметь находить необходимую литературу, базы данных и другие источники информации для автоматизации технологических процессов и производств.
P6	Уметь планировать и проводить эксперимент, интерпретировать данные и их использовать для ведения инновационной инженерной деятельности в области автоматизации технологических процессов и производств.
P7	Уметь выбирать и использовать подходящее программно–техническое оборудование, оснащение и инструменты для решения задач автоматизации технологических процессов и производств.
<i>Универсальные компетенции</i>	
P8	Владеть иностранным языком на уровне, позволяющем работать в интернациональной среде с пониманием культурных, языковых и социально – экономических различий.
P9	Эффективно работать индивидуально, в качестве члена и руководителя группы с ответственностью за риски и работу коллектива при решении инновационных инженерных задач в области автоматизации технологических процессов и производств, демонстрировать при этом готовность следовать профессиональной этике и нормам.
P10	Иметь широкую эрудицию, в том числе знание и понимание современных общественных и политических проблем, вопросов безопасности и охраны здоровья сотрудников, юридических аспектов, ответственности за инженерную деятельность, влияния инженерных решений на социальный контекст и окружающую среду.
P11	Понимать необходимость и уметь самостоятельно учиться и повышать квалификацию в течение всего периода профессиональной деятельности.

Министерство науки и высшего образования Российской Федерации  
 федеральное государственное автономное  
 образовательное учреждение высшего образования  
 «Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа – Инженерная школа информационных технологий и роботехники  
 Направление подготовки – 15.03.04 Автоматизация технологических процессов и производств  
 Уровень образования - бакалавр  
 Отделение школы (НОЦ) – Отделение автоматизации и робототехники  
 Период выполнения – весенний семестр 2020 учебного года

Форма представления работы:

бакалаврская работа

### КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН выполнения выпускной квалификационной работы

Срок сдачи студентом выполненной работы:

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
	Основная часть	70
	Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	15
	Социальная ответственность	15

**СОСТАВИЛ:**

**Руководитель ВКР**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОАР ИШИТР	Аврамчук Валерий Степанович	к.т.н., доцент		

**СОГЛАСОВАНО:**

**Руководитель ООП**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОАР ИШИТР	Громаков Евгений Иванович	к.т.н., доцент		

Министерство науки и высшего образования Российской Федерации  
 федеральное государственное автономное  
 образовательное учреждение высшего образования  
 «Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа - Инженерная школа информационных технологий и робототехники  
 Направление подготовки - 15.03.04 Автоматизация технологических процессов и производств  
 Отделение - Отделение автоматизации и робототехники

УТВЕРЖДАЮ:  
 Руководитель ООП  
 \_\_\_\_\_ Громаков Е. И.  
 (Подпись)    (Дата)    (Ф.И.О.)

### ЗАДАНИЕ на выполнение выпускной квалификационной работы

В форме:

бакалаврская работа
---------------------

Студенту:

Группа	ФИО
158Т61	Ван Лянь

Тема работы:

Автоматизация распознавания типов файловых систем по их сигнатурам, при их частичном повреждении.
---

Утверждена приказом директора (дата, номер)	№ 162-14/с от 10.06.2020
---	--------------------------

Срок сдачи студентом выполненной работы:	
--	--

#### ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

<p><b>Исходные данные к работе</b>  <i>(наименование объекта исследования или проектирования; производительность или нагрузка; режим работы (непрерывный, периодический, циклический и т. д.); вид сырья или материал изделия; требования к продукту, изделию или процессу; особые требования к особенностям функционирования (эксплуатации) объекта или изделия в плане безопасности эксплуатации, влияния на окружающую среду, энергозатратам; экономический анализ и т. д.).</i></p>	<p>Объектом проектирования является Алгоритм автоматического определения типа файловой системы по сигнатуре.</p> <p>Алгоритм имеет следующие требования:          Просмотр типа файловой системы логического диска; Просмотр начальной позиции и размера логического диска.</p>
<p><b>Перечень подлежащих исследованию, проектированию и разработке вопросов</b>  <i>(аналитический обзор по литературным источникам с целью выяснения достижений мировой науки техники в рассматриваемой области; постановка задачи исследования, проектирования, конструирования; содержание процедуры исследования, проектирования, конструирования; обсуждение результатов выполненной работы; наименование дополнительных разделов,</i></p>	<p>Узнавание структуру файловых систем FAT и NTFS, узнавание структуру главной загрузочной области(MBR) жесткого диска, В Delphi7 разработан алгоритм для автоматического определения типа файловой системы, На языке. С разработан алгоритм автоматического определения типа файловой системы по сигнатуре.</p>

<i>подлежащих разработке; заключение по работе).</i>	
<b>Перечень графического материала</b> <i>(с точным указанием обязательных чертежей)</i>	Алгоритм работы диаграммы Delphi7, алгоритм работы диаграммы языка С, сравнительная таблица WinHex

<b>Консультанты по разделам выпускной квалификационной работы</b> <i>(с указанием разделов)</i>	
<b>Раздел</b>	<b>Консультант</b>
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Конотопский Владимир Юрьевич
Социальная ответственность	Сечин Александр Иванович

<b>Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику</b>	
---	--

**Задание выдал руководитель:**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОАР ИШИТР	Аврамчук Валерий Степанович	к.т.н., доцент		

**Задание принял к исполнению студент:**

Группа	ФИО	Подпись	Дата
158Т61	Ван Лянь		

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА  
«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСООБЪЕКТИВНОСТЬ И  
РЕСУРСОСБЕРЕЖЕНИЕ»**

Студенту:

<b>Группа</b>	<b>ФИО</b>
158Т61	Ван Лянь

<b>Школа</b>	<b>ИШИТР</b>	<b>Отделение Школа</b>	<b>Автоматизации и робототехники</b>
<b>Уровень образования</b>	Бакалавриат	<b>Направление /специальность</b>	15.03.04 Автоматизация технологических процессов и производств

**Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:**

1. <i>Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих</i>	<i>Использовать действующие ценники и договорные цены на потребленные материальные и информационные ресурсы, а также указанную в МУ величину тарифа на эл. энергию</i>
2. <i>Нормы и нормативы расходования ресурсов</i>	—
3. <i>Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования</i>	<i>Действующие ставки единого социального налога и НДС (см. МУ, ставка дисконтирования <math>i=0.1</math>)</i>

**Перечень вопросов, подлежащих исследованию, проектированию и разработке:**

1. <i>Оценка коммерческого потенциала, перспективности и альтернатив проведения НИ с позиции ресурсоэффективности и ресурсосбережения</i>	<i>Оценка готовности полученного результата к выводу на целевые рынки, краткая характеристика этих рынков</i>
2. <i>Планирование и формирование бюджета научных исследований</i>	<i>Построение плана-графика выполнения ВКР, составление соответствующей сметы затрат, расчет величины НДС и цены результата ВКР</i>
3. <i>Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования</i>	<i>Качественная и количественная характеристика экономического и др. видов эффекта от внедрения результата, определение эффективности внедрения</i>

**Перечень графического материала**

1. Оценка конкурентоспособности ИП
2. Матрица SWOT
3. Диаграмма Ганта – <u>выполнить</u>
4. Бюджет НИ – <u>выполнить</u>
5. Основные показатели эффективности НИ

<b>Дата выдачи задания для раздела по линейному графику</b>	26.02.2020 г.
---	---------------

**Задание выдал консультант:**

<b>Должность</b>	<b>ФИО</b>	<b>Ученая степень, звание</b>	<b>Подпись</b>	<b>Дата</b>
Доцент ОСГН УОД	Конотопский Владимир Юрьевич	к.т.н., доцент		26.02.2020 г.

**Задание принял к исполнению студент:**

<b>Группа</b>	<b>ФИО</b>	<b>Подпись</b>	<b>Дата</b>
158Т61	Ван Лянь		

## ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

<b>Группа</b>	<b>ФИО</b>
158Т61	Ван Лянь

<b>Школа</b>	<b>ИШИТР</b>	<b>Отделение школы (НОЦ)</b>	<b>Автоматизации и робототехники</b>
Уровень образования	Бакалавриат	Направление/специальность	15.03.04 Автоматизация технологических процессов и производств

Тема ВКР:

**Автоматизация распознавания типов файловых систем по их сигнатурам, при их частичном повреждении.**

**Исходные данные к разделу «Социальная ответственность»:**

1. Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения	Алгоритм автоматической распознавания типа файловой системы по их сигнатурам. Работы проводились на обеспечить необходимую информацию для будущей потери таблицы разделов из-за сбой
--	---

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

<p><b>1. Правовые и организационные вопросы обеспечения безопасности:</b></p> <ul style="list-style-type: none"> <li>– специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства;</li> <li>– организационные мероприятия при компоновке рабочей зоны.</li> </ul>	<ul style="list-style-type: none"> <li>– Основные проводимые правовые и организационные мероприятия по обеспечению безопасности трудящихся на рабочем месте согласно СанПиН 2.2.2/2.4.1340-03, ФЗ – 197 и ГОСТ 12.2.032-78 ССБТ.</li> </ul>
<p><b>2. Производственная безопасность:</b></p> <p>2.1. Анализ выявленных вредных и опасных факторов</p> <p>2.2. Обоснование мероприятий по снижению воздействия</p>	<p>Анализ выявленных вредных и опасных факторов:</p> <ul style="list-style-type: none"> <li>– электромагнитное излучение,</li> <li>– Отклонение показателей микроклимата,</li> <li>– Превышение уровня шума,</li> <li>– Недостаточная освещенность рабочей зоны,</li> </ul>
<p><b>3. Экологическая безопасность:</b></p>	<ul style="list-style-type: none"> <li>– Наличие промышленных отходов (бумага-черновики, вторцвет- и</li> </ul>

	чермет, пластмасса, перегоревшие люминесцентные лампы, оргтехника, обрезки монтажных проводов, бракованная строительная продукция) и способы их утилизации;
<b>4. Безопасность в чрезвычайных ситуациях:</b>	<ul style="list-style-type: none"> <li>– Рассмотрены 2 ситуации ЧС:</li> <li>– 1) природная – сильные морозы зимой, (аварии на электро-, тепло-коммуникациях, водоканале, транспорте);</li> <li>– 2) Анализ пожарной безопасности.</li> </ul>

Дата выдачи задания для раздела по линейному графику	26.02.2020
--	------------

**Задание выдал консультант:**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Профессор ОКД, ИШНКБ	Сечин Александр Иванович	д.т.н., Профессор		

**Задание принял к исполнению студент:**

Группа	ФИО	Подпись	Дата
158Т61	Ван Лянь		26.02.2020

## Реферат

Бакалаврская работа содержит 85 страницы, 26 таблиц, 25 рисунков, 21 источник.

Ключевые слова: файловая система, файловая система FAT32, файловая система NTFS, главная загрузочная запись, сигнатура.

Объектом исследования является алгоритм автоматического определения типа файловой системы по сигнатуре.

Целью данной работы является разработка алгоритма определения NTFS, FAT32 и других типов.

В процессе исследования проводились: узнавание структуру файловых систем FAT и NTFS; узнавание структуру главной загрузочной области(MBR) жесткого диска; В Delphi 7 разработан алгоритм для автоматического определения типа файловой системы; На языке C разработан алгоритм автоматического определения типа файловой системы по сигнатуре.

В результате исследования разработан алгоритм автоматического определения типа файловой системы по сигнатуре.

В рамках разработки проекта в будущем планируется оптимизация алгоритма и введение в эксплуатацию.

## Содержание

Определения, обозначения и сокращения.....	12
Введение.....	13
<b>1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ.....</b>	<b>14</b>
1.1 Файловая система.....	14
1.2 Классификация файловой системы.....	15
1.2.1 Файловая система FAT32.....	15
1.2.2 Файловая система exFAT.....	20
1.2.3 Файловая система NTFS.....	21
1.3 Технологии восстановления данных: поиск по сигнатурам.....	26
1.4 Определение главной загрузочной записи (MBR).....	28
<b>2 ПРОЕКТНАЯ ЧАСТЬ.....</b>	<b>31</b>
2.1 Delphi 7 распознает дисковую файловую систему.....	31
2.1.1 Функции, используемые в эксперименте.....	31
2.1.2 Алгоритм получения типа файловой системы в Delphi 7.....	31
2.1.3 Анализ результатов.....	32
2.2 Программная реализация алгоритма чтения файла с логического раздела с помощью языка. С.....	33
2.2.1 Функции, используемые в эксперименте.....	33
2.2.2 Исходный код.....	36
2.2.3 Анализ результатов.....	45
<b>3 Финансовый менеджмент, ресурсоэффективность и ресурсосбережение....</b>	<b>51</b>
Введение.....	51
3.1 Оценка коммерческого потенциала и перспективности проведения исследований с позиции ресурсоэффективности и ресурсосбережения... 51	51
3.1.1 Анализ конкурентных технических решений.....	51
3.1.2 SWOT-анализ.....	53
3.2 Планирование научно-исследовательских работ.....	56
3.2.1 Структура работ в рамках научного исследования.....	56
3.2.2 Разработка графика проведения научного исследования.....	57

3.2.3 Бюджет научно-технического исследования.....	59
3.3 Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования	63
Выводы по разделу.....	65
4 Социальная ответственность.....	67
Введение.....	67
4.1 Правовые и организационные вопросы обеспечения безопасности....	68
4.2 Производственная безопасность.....	69
4.2.1 Анализ условий труда на рабочем месте.....	69
4.2.2 электромагнитное излучение.....	71
4.2.3 Отклонение показателей микроклимата.....	72
4.2.4 Превышение уровня шума.....	72
4.3 Экологическая безопасность.....	76
4.4 Безопасность в чрезвычайных ситуациях.....	78
4.4.1 Природная чрезвычайная ситуация.....	78
4.4.2 Анализ пожарной безопасности.....	78
Выводы по разделу.....	81
Заключение.....	82
Список используемых источников.....	83

## Определения, обозначения и сокращения

В данной работе применены следующие термины с соответствующими определениями:

FAT: Классическая архитектура файловой системы;

FAT32: Это файловая система, разновидность FAT;

ExFAT: иногда называется FAT64 — проприетарная файловая система, предназначенная главным образом для флэш-накопителей. [3];

NTFS: это файловая система журнала;

MBR: Главная загрузочная запись [5];

GPT: Таблица разделов;

EBR: Расширенная загрузочная запись, сектор на жёстком диске (512 байт).

## Введение

Моей темой является автоматизация распознавания типов файловых систем по их сигнатурам, при их частичном повреждении.

Моя тема актуальна, потому что всем интересна тема, которую мы изучаем. Актуальность данной работы определяется необходимостью разработки алгоритма автоматического определения типа файловой системы по сигнатуре. В будущем разработанный алгоритм в случае сбоя сможет предоставить необходимую информацию для восстановления потерь таблиц разделов.

Объектом исследования является алгоритм автоматического определения типа файловой системы по сигнатуре.

Целью данной работы является разработка алгоритма определения NTFS, FAT32 и других типов.

Задачи моей ВКР:

- рассмотрена структура файловых систем FAT и NTFS;
- рассмотрена структура главной загрузочной области(MBR) жесткого диска;
- разработан алгоритм для автоматического определения типа файловой системы в Delphi7;
- разработан алгоритм автоматического определения типа файловой системы по сигнатуре на языке C.

Наиболее приемлемым способом определения типа файловой системы по сигнатуре является алгоритм, который написан при помощи языка delphi 7, языка C и Winhex.

# 1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

## 1.1 Файловая система

Прежде чем раздел или диск можно будет использовать в качестве файловой системы, его необходимо инициализировать и записать структуру данных на диск. Этот процесс называется установлением файловой системы.

Файловая система — это метод и структура данных, используемые операционной системой для уточнения файлов на диске или разделе, то есть метод организации файлов на диске. Он решает проблему эффективного хранения данных на макете хранилища данных, управление пространством, именовании файлов и управлении безопасностью. С помощью файловой системы он гарантирует, что данные, записанные на диск, соответствуют считываемым данным. Реализует упрощение и безопасность чтения и записи данных.

В дополнение к хранению и управлению данными, хранящимися в файлах, файловая система также хранит на диске файлы и некоторую важную информацию о самой файловой системе, такую как права доступа к файлу, размер, дата изменения, владелец и место хранения и т. Д. Это метаданные файловой системы. Файловая система является мостом между операционной системой и дисковым устройством. Благодаря файловой системе достигается разумная организация и эффективный доступ к данным. Производительностью операционной системы является управление файлами и каталогами.

Функции файловой системы включают в себя:

- управление и планирование пространства хранения файла;
- обеспечение логической структуры; физической структуры и способа хранения файла;
- реализацию преобразования файла из идентификации в фактический адрес;

- реализацию операции управления и операции доступа к файлу и реализацию информации о файле;
- обмениваться и предоставлять надежные меры конфиденциальности и защиты файлов;
- обеспечивать меры безопасности файлов.

## **1.2 Классификация файловой системы**

Файловая система обеспечивает способ организации дисков. Он определяет, как данные хранятся на диске, и какие типы информации могут быть прикреплены к имени файла, разрешениям и другим атрибутам. Windows поддерживает три разных файловых системы. *NTFS* — самая продвинутая файловая система. Windows использует *NTFS* для своих системных дисков, и по умолчанию она используется для большинства несъемных дисков. *FAT32* — это старая файловая система, не такая эффективная, как *NTFS*, и она не поддерживает так много функций, но имеет большую совместимость с другими операционными системами. *ExFAT* — это современная альтернатива *FAT32*, которая поддерживает больше устройств и операционных систем, чем *NTFS*, но не так обширна, как *FAT32*.

### **1.2.1 Файловая система FAT32**

*FAT32* относится к таблице размещения файлов — это метод управления дисковыми файлами, который использует 32-разрядное двоичное управление записями, потому что ядром файловой системы FAT является таблица размещения файлов, и имя получено из нее [2].

Файловая система FAT поддерживает только четыре атрибута файла: ‘системный’; ‘скрытый’; ‘только чтение’ и ‘архивный’.

Чтобы обеспечить наиболее полную совместимость с существующими программами, сетями и драйверами устройств, в составе файловой системы *FAT32* были реализованы только минимально необходимые изменения архитектуры *Windows*, внутренних структур данных, интерфейсов API (*Application Programming Interface*) и дискового формата. Как правило,

существующие приложения и драйверы продолжают беспрепятственно работать на дисках FAT32.

В FAT32 отсутствуют разрешения и другие функции безопасности, встроенные в более современную файловую систему NTFS. Кроме того, современные версии *Windows* больше нельзя устанавливать на диск, отформатированный в FAT32; они должны быть установлены на диски, отформатированные в NTFS.

FAT32 идеально подходит для использования на съемных дисках, если нужна максимальная совместимость с самыми разными устройствами, но при условии, что на диске нет файлов размером 4 Гб или больше.

Структура раздела FAT32 показана на рисунке 1 [4].



Рисунок 1 – Структура раздела FAT32

Для каждого логического диска, использующего файловую систему FAT32, внутреннее пространство можно разделить на три части, за которыми следуют область загрузки (область BOOT), область таблицы размещения файлов (область FAT) и область данных (область DATA). Область загрузки и область таблицы размещения файлов также совместно называются системной областью, занимающей небольшое пространство в передней части всего логического диска и хранящей связанную управляющую информацию.

Область данных — это область, где логический диск используется для хранения содержимого файла, и эта область используется в кластерах в качестве единицы выделения.

### **Загрузочный сектор (область BOOT)**

Первый сектор файловой системы FAT32 представляет собой загрузочный сектор, который содержит много конкретной информации о файловой системе. Такой как количество таблиц FAT, размер каждой таблицы FAT (количество секторов) и слова в каждом секторе, Количество разделов, количество секторов, включенных в каждый кластер, количество зарезервированных секторов, размер файловой системы (количество секторов), начальное количество кластеров в корневом каталоге и другая дополнительная информация. Загрузочная область (область BOOT) использует три сектора (номер логического сектора 0) в первом секторе, в котором хранится количество байтов каждого сектора логического диска, количество секторов, соответствующих каждому кластеру, и Другие важные параметры и руководства записи. После этого есть несколько зарезервированных секторов, включая резервные копии от одного до трех секторов.

### **Область расположения файла [4]**

Область таблицы размещения файлов (область FAT) является наиболее важной областью для управления дисковым пространством и файлами файловой системы FAT. Он хранит информацию об использовании каждого кластера в области данных логического диска. Управление объемом памяти и свободного места, занимаемого файлами, осуществляется с помощью FAT. В области FAT хранятся в общей сложности две идентичные таблицы размещения файлов, поэтому, когда первый файл поврежден, второй доступен. Размер таблицы FAT определяется количеством кластеров в области данных логического диска и использует целое число секторов. Вы можете найти использование каждого кластера в области данных, посмотрев на значение заполнения в соответствующем месте в таблице FAT. В таблице

FAT32 каждый кластер занимает четыре байта (32 бита), указывая, что первые 8 байтов (байты 0H-07H) используются для хранения типа диска, поэтому эффективный номер кластера начинается с 02H. Кластер использования 02H представлен 32-битным двоичным числом, содержащим байты 08H-0BH, кластер использования 03H представлен 32-битным двоичным числом, содержащим байты 0CH-0FH, и так далее. Записать соответствующую позицию неиспользованного и восстановленного кластера в ноль, заполнить соответствующую позицию поврежденного кластера конкретным значением 0FFFFFF7H и заполнить соответствующую позицию выбранного кластера ненулевым значением. Особенно если кластер является файлом для последнего кластера значение заполнения равно 0FFFFFFFH. Если кластер не является последним кластером файла, значение заполнения — это номер кластера следующего кластера, занятого файлом. Поэтому кластеры, занимаемые файлом, образуют цепочку кластеров и хранятся в таблице FAT32.

### **Область данных**

Область данных используется для хранения пользовательских данных. После FAT2 область данных также разделяется на кластеры, начиная с номера 2. Начальная позиция кластера 2 является реальной начальной позицией области данных.

### **Корневой каталог**

Обычно корневой каталог находится в кластере № 2, но в принципе корневой каталог в файловой системе FAT32 может быть расположен где угодно в области данных. Область корневого каталога хранит записи каталога каждого файла в корневом каталоге, и каждая запись каталога занимает 32 байта. В файловой системе FAT32 корневой каталог используется как часть области данных, аналогично подкаталогам.

### **Подкаталог**

В файловой системе FAT32, за исключением корневого каталога, все подкаталоги создаются в соответствии с конкретными потребностями во

время использования. Если в корневом каталоге создается новый подкаталог, подкаталог называется подкаталогом корневого каталога, а корневой каталог называется родительским каталогом подкаталога. При создании нового подкаталога создайте запись каталога в кластере, назначенном его родительскому каталогу. Позиция каталога описывает номер начального кластера каталога, назначьте кластер свободному пространству и очистите его до нуля. Номер кластера записывается в его записи каталога. При создании записей каталога для подкаталогов первые две записи каталога используются в кластерах, назначенных подкаталогам, для записи их взаимосвязи с соответствующим родительским каталогом.

### **Форматы имен файлов [4]**

Файловая система FAT32 включает поддержку длинных имен файлов (LFN). Для хранения длинных имен используются элементы каталога, смежные с основным элементом. Имя файла записывается не в символах ASCII, а в *Unicode*. Вы можете сохранить фрагменты длиной до 13 символов Unicode в записи каталога. Неиспользованная часть последнего фрагмента заполнена кодом 0xFFFF. В таблице 1 показана структура элементов каталога с длинными именами файлов.

Таблица 1 – Структура элемента каталога для длинного имени файла

Смещение	Размер (байт)	Содержание
0x00	1	Номер фрагмента
0x01	10	Символы 1-5 имени файла в <i>Unicode</i>
0x0B	1	Атрибуты файла
0x0C	1	Байт флагов
0x0D	1	Контрольная сумма короткого имени
0x0E	12	Символы 6-11 имени файла в <i>Unicode</i>
0x1A	2	Номер первого кластера (заполняется нулями)
0x1C	4	Символы 12-13 имени файла в <i>Unicode</i>

Длинное имя сначала записывается в каталог, а последовательность фрагментов меняется на противоположную, начиная с конца. После длинного

(полного) имени помещается стандартный дескриптор файла, содержащий вариант этого имени, который можно сократить с помощью специального алгоритма. На Рисунке 2 показаны все записи папки файла Thequi ~ 1.fox, длинное имя файла – The quick brown.fox. Длинные имена используют Unicode, поэтому каждый символ в имени использует два байта в записи папки. Значение поля атрибута записи длинного имени – 0x0F. Поле атрибута для краткости 0x20, как показано на рисунке 2.

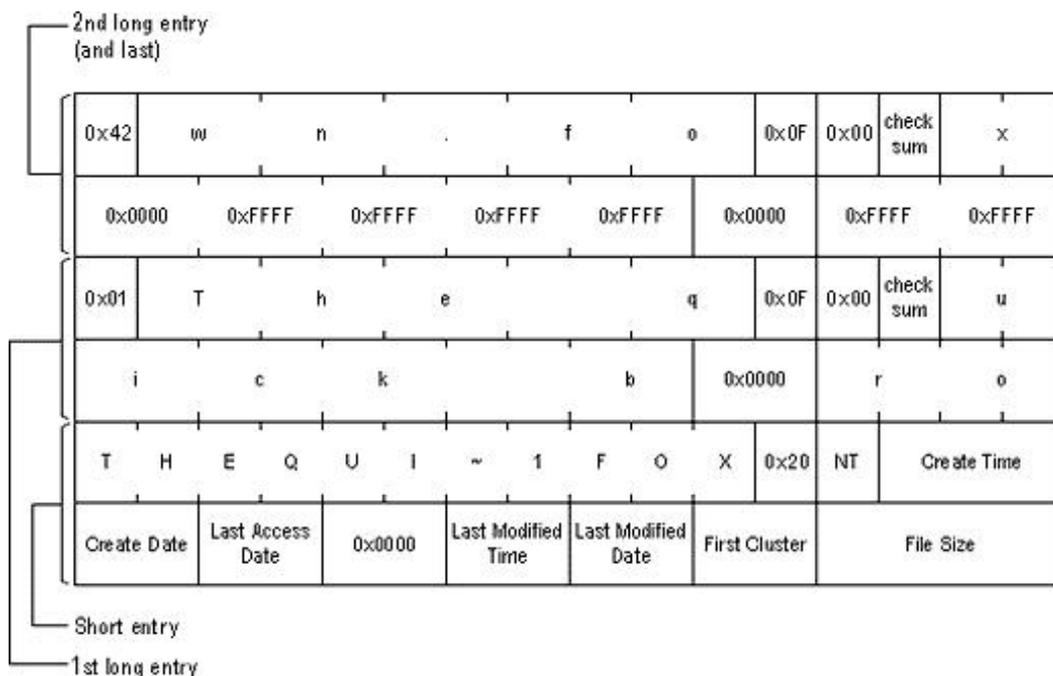


Рисунок 2 – Пример записи папки с длинным именем файла

На логическом диске с организацией FAT32 дополнительно присутствует структура *FSInfo*, размещаемая в первом секторе резервной области. Эта структура содержит информацию о количестве свободных кластеров на диске и о номере первого свободного кластера в таблице FAT.

### 1.2.2 Файловая система exFAT

ExFAT-это формат файловой системы USB-накопителя. ExFAT-это расширение файловой системы fat и имеет расширенные таблицы размещения файлов. Это файловая система, более подходящая для флэш-памяти, предназначенная для решения ограничений fat16 / 32. Появление этой файловой системы связано недостатками, обнаруженными в файловой

системе fat. ExFAT не имеет столько флэш-памяти, как формат ntfs, и из-за этого в системе не будет использоваться больше функций. И именно поэтому файловая система формата exfat больше подходит для различных мобильных карт памяти и U-дисков.

ExFAT также обладает лучшей совместимостью с различными операционными системами, по сравнению с NTFS, а устройства с этой файловой системой отлично читаются и перезаписываются как в Windows, так и в Mac OS, а также в Linux.

Преимущества формата ExFAT [3]:

1. Позволяет совместимость настольных компьютеров и мобильных устройств;
2. Позволяет количество перезаписей того же сектора. На флэш-накопителе после определенного количества операций записи блок памяти изнашивается необратимо;
3. Максимальный размер одного файла до 16 *EB*;
4. Размер кластера может быть до 32 *MB*;
5. Максимальное количество файлов в одном каталоге может достигать 65 536;
6. Позволяет распределение свободного места, введя растровые изображения свободного пространства, которые могут уменьшить фрагментацию диска.

### **1.2.3 Файловая система NTFS**

NTFS (файловая система новой технологии) — это формат диска, специально разработанный для сетевых и дисковых квот, шифрования файлов и других функций безопасности управления, поддерживаемых серией операционных систем ядра *Windows NT*. Он обеспечивает длинные имена файлов, защиту и восстановление данных, и к ним можно получить доступ через каталоги и разрешения файлов для обеспечения безопасности и поддержки между разделами.

## Физическая структура NTFS

Как и любая другая система, NTFS делит все полезное место на кластеры — блоки данных, используемые единовременно. NTFS поддерживает почти любые размеры кластеров — от 512 байт до 64 Кбайт, неким стандартом же считается кластер размером 4 Кбайт. Диск NTFS условно делится на две части. Первые 12% диска отводятся под так называемую MFT зону — пространство, в которое растет метафайл MFT. Запись каких-либо данных в эту область невозможна. MFT-зона всегда держится пустой — это делается для того, чтобы самый главный, служебный файл (MFT) не фрагментировался при своем росте. Остальные 88% диска представляют собой обычное пространство для хранения файлов, как показано на рисунке 3 [1].



Рисунок 3 – Структура Диска NTFS

Однако доступное дисковое пространство включает в себя все физически доступное пространство — также включена незаполненная часть области MFT. Механизм использования области MFT заключается в следующем: когда файл больше не может быть записан в обычное пространство, необходимо уменьшить только область MFT, тем самым освобождая место для записи файла. Когда пространство в нормальной области MFT освобождается, область может быть снова расширена. В то же время, не исключено, что в этой области хранятся обычные файлы: никакой аномалии тут нет.

### **MFT и его структура**

Файловая система NTFS представляет собой выдающееся достижение структуризации: каждый элемент системы представляет собой файл — даже

служебная информация. Самый главный файл на NTFS называется MFT, или *Master File Table* — общая таблица файлов. Именно он размещается в MFT зоне и представляет собой централизованный каталог всех остальных файлов диска, и как не парадоксально, себя самого. MFT поделен на записи фиксированного размера (обычно 1 Кбайт), и каждая запись соответствует какому-либо файлу (в общем смысле этого слова). Первые 16 файлов несут служебный характер и недоступны операционной системе — они называются метафайлами, причем самый первый метафайл — сам MFT. Эти первые 16 элементов MFT — единственная часть диска, имеющая фиксированное положение. Интересно, что вторая копия первых трех записей, для надежности — они очень важны — хранится ровно посередине диска. Остальной MFT-файл может располагаться, как и любой другой файл, в произвольных местах диска — восстановить его положение можно с помощью его самого, «зацепившись» за самую основу — за первый элемент MFT.

### **Метафайлы**

Метафайлы находятся в корневом каталоге диска NTFS - они начинаются с символов «\$», хотя сложно получить какую-либо информацию о них стандартным способом [1]. В следующей таблице 2 показаны используемые в настоящее время метафайлы и их назначение.

Таблица 2 – Метафайлы [1]

<i>\$MFT</i>	Сам <i>MFT</i>
<i>\$LogFile</i>	Это эквивалентная запись в файле журнала. Эта запись содержит последовательность шагов обработки для реализации восстанавливаемых функций NTFS. Размер записи зависит от размера раздела диска до 4 МБ. В случае сбоя системы (например, если произошел неожиданный сбой питания) NTFS будет проверять его содержимое и структуру на основе журналов при следующей загрузке, чтобы сохранить его согласованность с последним завершением работы.

## Продолжение таблицы 2 – Метафайлы [1]

\$Uppcase	Это запись таблицы символов ( <i>Uppcase table</i> ). Функция этой записи заключается в преобразовании символов нижнего регистра в символы верхнего регистра, соответствующие <i>Unicode</i> . NTFS поддерживает <i>Unicode</i> через него.
\$AttrDef	Это запись определения атрибута, содержащая имя атрибута, номер атрибута и таблицу описания атрибута для всех файлов и каталогов. Поскольку NTFS считает, что каждый файл состоит из набора атрибутов, вся информация, относящаяся к файлу, рассматривается как определенный тип атрибута, что означает, что тип атрибута также определяет некоторые характеристики файла, такие как атрибут дескриптора безопасности, который определяет владение файлом в многопользовательской среде. Таким образом, NTFS будет различать операции в зависимости от определения атрибута во время любой операции файла.
\$Secure	Это запись файла безопасности, которая содержит инструкции по безопасности для всех файлов в этом разделе. Он взаимодействует со свойствами файла для реализации функций безопасности, таких как назначение разрешений в операционных системах семейства NT.
\$Extend	Это запись файла расширения NTFS, которая используется несколькими функциями расширения NTFS, такими как дисковые квоты, точки перезагрузки и целевые идентификаторы. Многие обновления NTFS 5.0 относительно предыдущей версии достигаются путем изменения этой записи.
\$.	Корневой каталог

### Файлы

Имя файла может содержать любые символы, включая полный набор национальных алфавитов, так как данные представлены в *Unicode* — 16-битном представлении, которое дает 65535 разных символов. Максимальная длина имени файла — 255 символов.

### Каталоги

Каталог в NTFS — это специальный файл, который хранит ссылки на другие файлы и каталоги, создавая иерархическую структуру данных на диске. Файл каталога разделен на блоки, каждый из которых содержит имя файла, основные атрибуты и элементы MFT и уже содержит ссылки на полную информацию об элементах каталога. Внутренняя структура каталога

представляет собой двоичное дерево. Что это означает: для поиска файла с заданным именем в линейном каталоге (например, FAT) операционная система должна просматривать все элементы в каталоге, пока не будет найден правильный файл. Бинарное дерево, однако, упорядочивает имена файлов таким образом, что поиск файла происходит быстрее — путем получения двузначных ответов на вопросы о положении файла [1], как показано на рисунке 4.

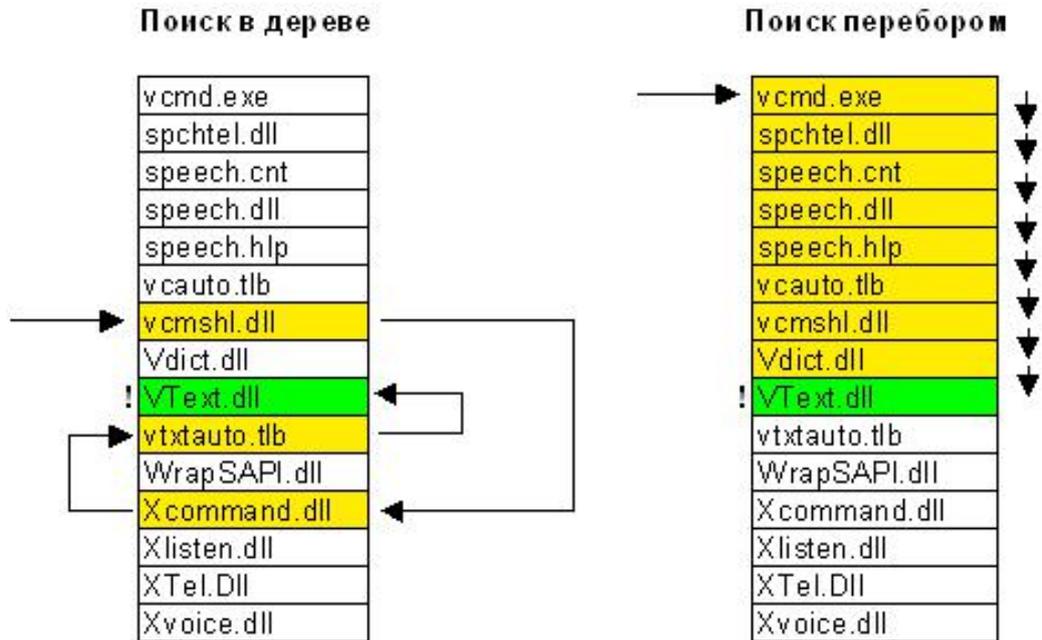


Рисунок 4 – поиск файл в линейном каталоге и в двоичном дереве

NTFS — отказоустойчивая система, которая вполне может привести себя в корректное состояние при практически любых реальных сбоях. Любая современная файловая система основана на таком понятии, как транзакция — действие, совершаемое целиком и корректно или не совершаемое вообще. У NTFS просто не бывает промежуточных (ошибочных или некорректных) состояний — квант изменения данных не может быть поделен на до и после сбоя, принося разрушения и путаницу — он либо совершен, либо отменен.

### Жесткие ссылки [1]

Эта вещь существовала в NTFS давно, но она используется редко: жесткие ссылки предполагают, что один и тот же файл имеет два имени (несколько указателей на каталоги файлов или несколько каталогов на одну и ту же запись MFT) Имена 1.txt и 2.txt: если пользователь удаляет файл 1,

файл 2 сохраняется. Если он удаляет 2, файл 1 остается, то есть оба имени с момента создания полностью совпадают. Файл физически стирается только после удаления его имени.

### **1.3 Технологии восстановления данных: поиск по сигнатурам**

Продаваемый под разными именами, поиск по сигнатурам файлов (*Signature-search*) является одной из важнейших технологий восстановления данных, известных сегодня. Эта технология позволяет находить информацию по содержанию, которое больше не отображается в файловой системе даже под знаком «удален», то есть фактически позволяет пользователю восстанавливать файлы с отформатированных, перераспределенных и поврежденных носителей информации. Однако, как и многие другие технологии, поиск по сигнатурам имеет свои ограничения. Попробуем разобраться в реальных преимуществах и недостатках метода. [7]

#### **Как работает поиск по сигнатурам**

##### **Определение файла**

Основной принцип работы алгоритмов сигнатурного поиска такой же, как у самых первых антивирусов. Как антивирус сканирует файл в поисках участков данных, совпадающих с известными фрагментами кода вирусов, так и алгоритмы сигнатурного поиска, используемые в программах для восстановления данных, считывают информацию с поверхности диска в надежде встретить знакомые участки данных. Заголовки многих типов файлов содержат характерные последовательности символов. К примеру, файлы в формате \*.jpeg содержат последовательность символов «JFIF», \*.bmp изображения начинаются с «BM», архивы \*.zip начинаются с символов «PK», \*.dbf базы данных содержат сигнатуру «OPLDatab», а документы PDF начинаются с символов «%PDF-».

Некоторые файлы (к примеру, текстовые и HTML файлы) не обладают характерными сигнатурами, но могут быть определены по косвенным признакам, т.к. содержат только символы из таблицы ASCII.

### **Определение размера файла [7]**

Для восстановления файла мало найти его начало, нужно также определить его конец. Конец файла можно найти, зная размер и адрес начала файла. Размер файла определяется либо анализом заголовка (\*.zip, \*.jpeg, \*.avi, \*.psd, \*.pst, \*.rar, \*.tiff и т.п.), либо считыванием и анализом секторов диска, идущих сразу за заголовком. К примеру, концом текстового или HTML файла алгоритм будет считать первый же сектор, который будет содержать символы, не входящие в таблицу ASCII.

### **Ограничения**

Поиск подписи не панацея. Перезапись содержимого файлов и фрагментирование дисков (особенно больших файлов) негативно влияют на возможность восстановления информации.

### **Гибридные алгоритмы**

Современные программы восстановления данных используют гибридный подход к анализу диска, стараются считывать как можно больше информации из файловой системы и только в экстренных случаях анализировать содержимое диска на наличие сигнатур (если файловая система повреждена или потеряна), а также защищают уже удаленные файлы.

### **Преимущества алгоритмов поиска по сигнатурам [7]**

Преимущества использования средств восстановления, оснащенных алгоритмами поиска по сигнатурам, очевидны – это гораздо более эффективное восстановление наиболее важных файлов на диске. Алгоритмы поиска по сигнатурам могут обнаруживать и качественно восстанавливать офисные документы (DOC, XLS, PPT, RTF и другие), сжатые архивы, цифровые фотографии, сообщения электронной почты, файлы баз данных, аудио и видео контента. С *Signature*-поиском могут быть найдены даже более ранние (но, возможно, лучше сохранившиеся) версии документов. Наконец,

поиск по сигнатурам эффективен даже там, где другие методы не срабатывают: на отформатированных, поврежденных и перераспределенных носителях, а также дисках с поврежденной и недоступной файловой системой.

### **Ограничения алгоритмов поиска по сигнатурам [7]**

Восстановление данных с помощью *Signature*-поиска может иметь дело только с известными типами файлов. Конечно, большинство видов файлов хранятся в базах данных сигнатур, однако есть много других видов информации, не содержащихся в них. Поиск по сигнатурам по своей природе ограничен только определенными типами данных. Вот почему *Signature*-поиск считается важным методом восстановления данных, но не может полностью заменить традиционного сканирования файловых систем. Вторым наиболее важным недостатком алгоритмов поиска по сигнатурам является то, что они должны детально просканировать весь диск (быстрой навигации по файловой системе для них достаточно), а это может занять немало времени.

### **1.4 Определение главной загрузочной записи (MBR)**

В главной загрузочной записи записывается информация о самом жестком диске, а также о размере и расположении каждого раздела жесткого диска. В случае повреждения основная информация о структуре данных на жестком диске будет потеряна.

Главной загрузочной записи (MBR), также известная как главный загрузочный сектор, является первым сектором, который необходимо прочитать, когда компьютер включен для доступа к жесткому диску. При подробном обсуждении внутренней структуры главного загрузочного сектора иногда первые 446 байтов контента специально обозначаются как «Главная загрузочная запись» (MBR), за которыми следуют четыре 16-байтовые «таблицы разделов диска» (DPT), и 2-байтовая конечная метка (55AA). Как показано на рисунке, структура записи раздела показана в таблице 3.

## Структура MBR [5]

Таблица 3 – Классическая структура главной загрузочной записи (MBR)

Смещение	Длина, байт	Описание	
0000h	446	Код загрузчика	
01BEh	16	Раздел 1	<i>Таблица разделов</i>
01CEh	16	Раздел 2	
01DEh	16	Раздел 3	
01EEh	16	Раздел 4	
01FEh	2	Сигнатура (55h AAh)	

### Код загрузчика

После завершения процедуры POST в ОЗУ по физическому адресу 0x7C00 записывается код загрузчика (первые 446 байт из нулевого сектора диска), после чего ему передаётся управление. Задача этого кода — проанализировать таблицу разделов (таблица 4) жёсткого диска, затем передать управление второму загрузочному коду, который может находиться или в начале активного раздела, или на специальной области диска (эта область не занята файловыми системами и обычно является группой секторов № 1-№ 63, разделы на диске обычно начинаются с сектора № 64). Второй загрузчик уже умеет читать хотя бы одну файловую систему, и его задача — передать управление файлам из файловой системы ОС для запуска ОС. [5]

Таблица 4 – Формат таблицы разделов

Байт хранения	Содержание и значение
1-й байт	Загрузочный знак. Значение 80H указывает активный раздел, а значение 00H указывает неактивный раздел.
2,3,4-й байт	Исходный номер головки, номер сектор, цилиндр номер этого раздела из них:
	Исходный номер головки — 2-й байт номер сектор — Младшие 6 бит третьего байта

Продолжение таблицы 4 – Формат таблицы разделов

Байт хранения	Содержание и значение
2, 3, 4-й байт	цилиндр номер — Это старшие 2 бита 3-го байта + 8 бит 4-го байта
5-й байт	Коды типов разделов:
	00h   Пустая запись (свободное место)
	06h   FAT-16B
	0Bh   FAT-32
	05h   Расширенный раздел
	07h   NTFS
6,7,8-й байт	0Fh   Расширенный раздел LBA (то же что и 05h, с использованием LBA)
	Номер концевой головки, номер сектора, номер цилиндра этой перегородки. Из них:
	Головной номер — 6-й байт; номер сектора — младшие 6 бит 7-го байта
9,10,11,12-й байт	Номер цилиндра — верхние 2 бита 7-го байта + 8-й байт.
	Количество секторов, которые использовались ранее в этом разделе

В случае если используется расширенный раздел, координаты начала раздела указывают на EBR, как показано на Таблице 5[6].

Таблица 5 Структура EBR

Смещение	Длина	Описание
1BEh	16	Указатель на раздел
1CEh	16	Указатель на следующий EBR
1DEh	32	Не используется (должно быть заполнено нулями)
1FEh	2	Сигнатура (55h AAh)

## 2 ПРОЕКТНАЯ ЧАСТЬ

### 2.1 Delphi 7 распознает дисковую файловую систему

#### 2.1.1 Функции, используемые в эксперименте

*GetVolumeInformation()* определяет серийный номер, метку диска и его файловую систему.

```
BOOL GetVolumeInformation(  
LPCTSTR lpRootPathName, //Корневой каталог, в котором  
находится том, например: «C: \», если NULL, это означает текущий  
каталог  
LPTSTR lpVolumeNameBuffer, //буфер имен томов  
DWORD nVolumeNameSize, //длина буфера имени тома  
LPDWORD lpMaximumComponentLength, //Максимальная длина  
имени файла, зависит от файловой системы  
LPDWORD lpFileSystemFlags, //параметры файловой системы  
LPTSTR lpFileSystemNameBuffer, //объясняет, что такое  
файловая система, например NTFS, FAT и т. Д.  
DWORD nFileSystemNameSize); //длина буфера имени файловой системы
```

#### 2.1.2 Алгоритм получения типа файловой системы в Delphi 7

```
procedure TForm1.Button1Click(Sender: TObject);
```

```
var
```

```
RootPath:array [0..20] of Char ;
```

```
VolName: array [0..255] of Char;
```

```
SerialNumber: DWORD;
```

```
MaxCLength: DWORD;
```

```
FileSysFlag: DWORD;
```

```
FileSysName: array[0..255] of Char;
```

```
begin
```

```
StrCopy(@RootPath,PChar(Edit1.Text));
```

```
GetVolumeInformation(RootPath,VolName,255,
```

```
@SerialNumber,MaxCLength,FileSysFlag,
```

```

FileSysName,255);
Memo1.Lines.Add('RootPath = '+RootPath);
Memo1.Lines.Add('VolumeName = '+VolName);
Memo1.Lines.Add('FSName = '+FileSysName);
end;
procedure TForm1.Edit1Click(Sender: TObject);
begin
Edit1.Text:="";
end;

```

### 2.1.3 Анализ результатов

На рисунке 5 представлены результаты, полученные после запуска программы, а на рисунке 6 представлена фактическая информация.

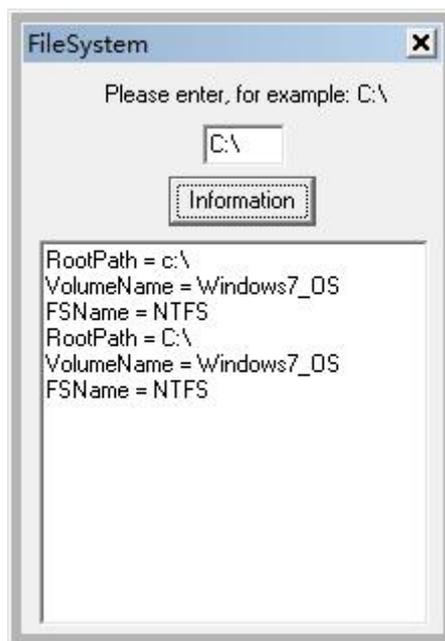


Рисунок 5 – Результат операции

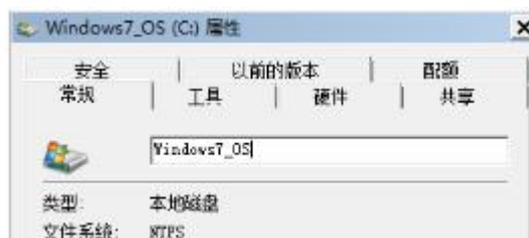
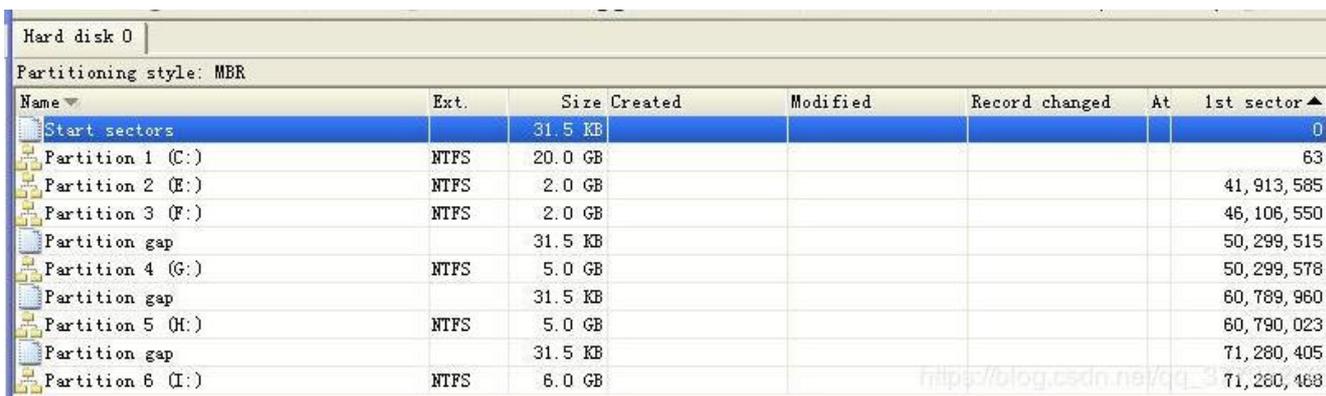


Рисунок 6 – фактическая информация

Из рисунка видно, что результаты экспериментальной программы верны.

## 2.2 Программная реализация алгоритма чтения файла с логического раздела с помощью языка. С

Эксперимент проводился на виртуальной машине winXP, а стиль раздела MBR. В виртуальной машине было выделено 40 ГБ дискового пространства. Три диска в главном разделе С, Е и F и три логических диска G, H и I в расширенном разделе показаны на рисунке 7.



Name	Ext.	Size	Created	Modified	Record changed	At	1st sector
Start sectors		31.5 KB					0
Partition 1 (C:)	NTFS	20.0 GB					63
Partition 2 (E:)	NTFS	2.0 GB					41,913,585
Partition 3 (F:)	NTFS	2.0 GB					46,106,550
Partition gap		31.5 KB					50,299,515
Partition 4 (G:)	NTFS	5.0 GB					50,299,578
Partition gap		31.5 KB					60,789,960
Partition 5 (H:)	NTFS	5.0 GB					60,790,023
Partition gap		31.5 KB					71,280,405
Partition 6 (I:)	NTFS	6.0 GB					71,280,468

Рисунок 7 – Информация о диске видна на Winhex

### 2.2.1 Функции, используемые в эксперименте

A. *CreateFile()* Функция, чтобы открыть устройство.

HANDLE *CreateFile*(

LPCSTR *lpFileName*,

DWORD *dwDesiredAccess*,

DWORD *dwShareMode*,

LPSECURITY\_ATTRIBUTES *lpSecurityAttributes*,

DWORD *dwCreationDisposition*,

DWORD *dwFlagsAndAttributes*,

HANDLE *hTemplateFile*);

*lpFileName* – Это имя файла или устройства, которое будет создано или открыто. Текст диска ("\\.\PhysicalDrive0"). Если есть два диска, второй диск-1.

*dwDesiredAccess* – это запрос на доступ к файлу или устройству, а общие значения-GENERIC\_READ, GENERIC\_WRITE или оба (GENERIC\_READ / GENERIC\_WRITE).

*dwShareMode* – Это режим общего доступа к файлам или устройствам,

который может читать, записывать, удалять и удалять запросы.

*lpSecurityAttributes*, который является указателем на структуру SECURITY\_ATTRIBUTES. Если этот параметр имеет значение NULL, дескриптор, возвращаемый CreateFile, не может наследоваться ни одним дочерним процессом, который может создать приложение, с использованием атрибутов безопасности по умолчанию.

*dwCreationDisposition*, операция, выполняемая с файлом или устройством, который существует или не существует. Для устройств, отличных от файлов, этот параметр обычно имеет значение OPEN\_EXISTING.

*dwFlagsAndAttributes*, который представляет собой атрибуты файла или устройства и флаги.

*hTemplateFile*, действительный дескриптор файла шаблона с доступом GENERIC\_READ. Этот параметр может быть НЕДЕЙСТВИТЕЛЕН.

Возвращаемое значение: если функция завершается успешно, возвращаемое значение является открытым дескриптором указанного файла, устройства, именованного канала или почтового слота. Если функция завершается ошибкой, возвращаемое значение INVALID\_HANDLE\_VALUE. Чтобы получить код ошибки, можно вызвать функцию GetLastError ().

Б. *DeviceIoControl()* Функция, возвращающая информацию об устройстве диска.

```
BOOL DeviceIoControl(  
    HANDLE hDevice,  
    DWORD dwIoControlCode,  
    LPVOID lpInBuffer,  
    DWORD nInBufferSize,  
    LPVOID lpOutBuffer,  
    DWORD nOutBufferSize,  
    LPDWORD lpBytesReturned,  
    LPOVERLAPPED lpOverlapped);
```

*hDevice* — дескриптор устройства, с которым нужно работать. В этом эксперименте это дескриптор устройства, возвращаемый функцией `CreateFile()`.

*DwIoControlCode* — это управляющий код операции, который определяет конкретную операцию и тип устройства, которое ее выполняет. Необходимо использовать `IOCTL_DISK_GET_DRIVE_GEOMETRY`, чтобы получить информацию о диске.

*lpInBuffer* — указатель на входной буфер, который может быть НЕДЕЙСТВИТЕЛЕН.

*nInBufferSize* — размер входного буфера в байтах.

*lpOutBuffer*, указатель на выходной буфер, используемый для получения данных, возвращаемых операцией.

*nOutBufferSize* — размер выходного буфера в байтах.

*lpBytesReturned* — указатель на переменную, которая получает размер (в байтах) данных, хранящихся в выходном буфере. Если выходной буфер слишком мал для приема каких-либо данных, вызов завершается неудачно, `GetLastError()` возвращает `ERROR_INSUFFICIENT_BUFFER`, а `lpBytesReturned` равно нулю.

*lpOverlapped*, указатель на структуру `OVERLAPPED`. Если `hDevice` открывается без указания `FILE_FLAG_OVERLAPPED`, `lpOverlapped` игнорируется.

**Возвращаемое значение:** если операция завершается успешно, возвращаемое значение не равно нулю.

В. *SetFilePointer()* Функция, чтобы установить место для чтения информации о диске.

```
DWORD SetFilePointer(  
HANDLE hFile,  
LONG lDistanceToMove,  
PLONG lpDistanceToMoveHigh,
```

DWORD dwMoveMethod

);

*hFile* — это дескриптор файла.

*lDistanceToMove* — это младшие 32 бита числа со знаком, указывающего расстояние, используемое для указания байтового смещения указателя файла.

*lpDistanceToMoveHigh* — старшие 32 бита расстояния.

*DwMoveMethod* — это начальная точка перемещения указателя файла.

Г. *ReadFile()* Функция для чтения содержимого диска.

BOOL ReadFile(  
HANDLE hFile,  
LPVOID lpBuffer,  
DWORD nNumberOfBytesToRead,  
LPDWORD lpNumberOfBytesRead,  
LPOVERLAPPED lpOverlapped  
);

*HFile* — дескриптор файла / устройства.

*lpBuffer* — указатель на буфер, используемый для получения прочитанных данных.

*nNumberOfBytesToRead* — максимальное количество байтов для чтения.

*lpNumberOfBytesRead*, используется для получения количества прочитанных байтов.

*lpOverlapped* может быть NULL.

### 2.2.2 Исходный код

```
#include <windows.h>
```

```
#include <winioctl.h> // Разработка и управление драйвером DDK //
```

```
#include <stdio.h>
```

```
#include <math.h>
```

```

#include <stdlib.h>
#include <stdint.h>
#define BufferLength 1024
// позволяет преобразовать значение, хранящееся в четырех
// последовательных байтах, в тип int //
uint32_t transtoint(unsigned char a[]){
uint32_t sum = 0UL;
uint32_t temp = 0UL;
for (int i = 3; i>=0; i--){
temp = a[i];
temp = temp << (8 * (3-i));
sum = sum | temp;}
return sum;}
// Шестнадцатеричный вывод //
bool HexOutput (char* buf, size_t len,bool ismbr, ULONGLONG *
baseaddr,ULONGLONG * nextaddr,int EBRnum){
bool mbrflag=1;// При чтении MBR можно определить, является ли запись
основным разделом или расширенной записью раздела. //
unsigned char a = buf[0];
if(ismbr)
printf("-----");
else
printf("-----%d-EBR-----",EBRnum);
if(ismbr)
printf("    первая часть(MBR): \n\n");
else
printf("    первая часть(EBR): \n\n");
int flag = 0;
for (size_t i = 0; i < len; ++i){
unsigned char c = buf[i]; // должен использовать unsigned char для вывода

```

```

значения > 128 //
flag++;
if (c < 16)
printf("0%x ", c);
else
printf("%x ", c);
if (i == 445){
flag = 0;
printf("\n\n      Вторая часть (таблица разделов): \n");}
if (i == 509){
flag = 0;
printf("\n\n      третья часть(Конец знака): \n");}
if ((flag) % 16 == 0)
printf("\n");}
printf("\n\n Адрес и размер раздела: \n\n");
int limit=ismbr?509:477; // Если вы уже знаете, что это расширенная часть
EBR, вам не нужно слишком много читать, rank = 1 – таблица разделов, rank
= 2 – следующая //
for (int m = 445, rank = 1; m < limit&&rank<=4; m += 16, rank++){
unsigned char fifth = buf[m + 5]; // Получить пятый бит флага файловой
системы, 05H или 0FH шестнадцатеричный расширенный раздел //
if(fifth==0x5||fifth==0xf) //Это расширенная запись раздела, не нужно читать
ее позже //
{printf("Это расширенная часть!\n");
mbrflag=0;
rank = 4;}//Не mbr это расширенный раздел EBR //
if (fifth < 16) // Отрегулируйте выходной формат //
printf("%d-й флаг таблицы разделов: 0%x\n", rank, fifth);
else
printf("%d-й флаг таблицы разделов: %x\n", rank, fifth);

```

```

if (fifth == 0x00) // Когда пятый бит (флаг) равен 00, это означает, что
информация таблицы разделов пуста и разделов нет //
{printf(" Таблица разделов пуста\n\n");//Не нужно продолжать читать}
else { // Запись раздела //
unsigned char offsetadd[4] = { 0 };
for (int n = m + 12, t = 0; n > m + 8, t < 4; n--, t++){
unsigned char temp = buf[n];
if (temp < 16)
printf(" 0%x ", temp);
else
printf(" %x ", temp);
offsetadd[t] = buf[n];}
//Рассчитать адрес, преобразовать в десятичный номер сектора LBA //
printf("\n");
uint32_t tempadd = transtoint(offsetadd);
printf("\n Начальный адрес: %I64x", (ULONGLONG)tempadd *
(ULONGLONG)512 + *baseaddr);
if(ismbr&&!mbrflag)// если в mbr и получил расширенную запись, EBR в
relsecor + nowbase (0)
{*baseaddr=(ULONGLONG)tempadd* (ULONGLONG)512 + *baseaddr;//only
change once
*nextaddr = (ULONGLONG)0UL;}
else if (!mbrflag)//if it's the extend entry //
{*nextaddr = (ULONGLONG)tempadd * (ULONGLONG)512;}
printf("\n\n");
printf("Размер: ");
for (int p = m + 16, w = 0; p > m + 12, w < 4; p--, w++){
unsigned char temp1 = buf[p];
if (temp1 < 16)
printf(" 0%x ", temp1);

```

```

else
printf(" %x ", temp1);
offsetadd[w] = buf[p];}
//Рассчитать размер и преобразовать в ГБ
printf("\n");
uint32_t tempsize = transtoint(offsetadd);
if(ismbr && !mbrflag)
printf("\nОбщий размер диска расширения: %lu Секторов = %lf ГБ \n",
tempsize, ((double)tempsize / 2.0 / 1024.0 / 1024.0));
else if(!ismbr)
{printf("\nРазмер %d-ого логического диска: : %lu Секторов = %lf ГБ
\n",EBRnum, tempsize, ((double)tempsize / 2.0 / 1024.0 / 1024.0));}
else
printf("\n Размер диска : %lu Секторов = %lf ГБ \n", tempsize,
((double)tempsize / 2.0 / 1024.0 / 1024.0));}
printf("\n\n");
return (mbrflag);}
// Функция: Анализ основной таблицы разделов, чтобы получить адрес
смещения и размер раздела каждого раздела. //
BOOL GetDriveGeometry(DISK_GEOMETRY *pdg, int addr){
HANDLE hDevice;
BOOL bResult; // флаг результатов
DWORD junk; // отменить результаты
char lpBuffer[BufferLength] = { 0 };
//Используйте CreateFile, чтобы получить дескриптор устройства //
hDevice = CreateFile(TEXT("\\\\.\\PhysicalDrive0"), // Название устройства,
здесь относится к первому жесткому диску //
GENERIC_READ, // нет доступа к диску //
FILE_SHARE_READ | FILE_SHARE_WRITE, // режим обмена //
NULL, // атрибуты безопасности по умолчанию //

```

```

OPEN_EXISTING, // disposition //
0, // атрибуты файла //
NULL); // не копировать атрибуты файла //
if (hDevice == INVALID_HANDLE_VALUE){ // не могу открыть диск //
printf("Creatfile error!May be no permission!ERROR_ACCESS_DENIED! \n");
return (FALSE);}
// IO с устройством через функцию DeviceIoControl //
bResult = DeviceIoControl(hDevice,
IOCTL_DISK_GET_DRIVE_GEOMETRY, // Контрольный код с указанием
типа устройства //
NULL,
0, // Нет входного буфера //
pdg,sizeof(*pdg),
&junk, // #возвращено байтов //
(LPOVERLAPPED)NULL); // синхронный ввод / вывод //
LARGE_INTEGER offset; // давно подписано //
offset.QuadPart = (ULONGLONG)addr * (ULONGLONG)512;
SetFilePointer(hDevice, offset.LowPart, &offset.HighPart,
FILE_BEGIN);//Читать MBR от 0 //
if(GetLastError())
printf("Код типа ошибки : %ld\n\n", GetLastError());//Если что-то пойдет не
так //
DWORD dwCB;
BOOL bRet = ReadFile(hDevice, lpBuffer, 512, &dwCB, NULL);
bool finished=0;
int EBRnum=0;
ULONGLONG *baseaddr=new ULONGLONG,*nextaddr= new
ULONGLONG;//Расширенный начальный адрес раздела, адрес EBR //
*baseaddr = (ULONGLONG)0;
*nextaddr = (ULONGLONG)0;

```

```

finished=HexOutput(lpBuffer, 512,true,baseaddr,nextaddr,EBRnum);//Сначала
прочитать MBR //
if(finished)
CloseHandle(hDevice);
else
{do{
EBRnum++;
memset(lpBuffer, 0, sizeof(lpBuffer));
offset.QuadPart = (ULONGLONG)(*baseaddr + *nextaddr);// найти EBR //
SetFilePointer(hDevice, offset.LowPart, &offset.HighPart, FILE_BEGIN);//
Читает EBR //
if(GetLastError())
printf("Код типа ошибки : %ld\n\n", GetLastError());// Если что-то пойдет не
так //
ReadFile(hDevice, lpBuffer, 512, &dwCB, NULL);
}while(!HexOutput(lpBuffer, 512,false,baseaddr,nextaddr,EBRnum));
CloseHandle(hDevice);}
delete baseaddr;
delete nextaddr;
return bResult;
}
extern int add[20];
extern int disknum;
int main(){
DISK_GEOMETRY pdg; // Структура для хранения параметров диска //
BOOL bResult; // флаг общих результатов //
ULONGLONG DiskSize; // размер накопителя, в байтах //
bResult = GetDriveGeometry(&pdg, 0);
if (bResult){
DiskSize = pdg.Cylinders.QuadPart * (ULONG)pdg.TracksPerCylinder *

```

```

(ULONG)pdg.SectorsPerTrack * (ULONG)pdg.BytesPerSector;
printf("Общий размер диска = %I64d (Bytes) = %I64d (ГБ)\n", DiskSize,
DiskSize / (1024 * 1024 * 1024));}
else
{printf("GetDriveGeometry failed. Error %ld.\n", GetLastError());}
return ((int)bResult);}

```

### 2.2.3 Графическое представление алгоритмов (блок-схема)

Чтобы облегчить читателю понимание того, как работает программа, здесь представлено графическое представление (блок-схема) алгоритма.

Основная программа начинается с определения необходимых переменных. Затем вызовете подпрограмму (1), чтобы получить возвращаемое значение *bResult*. Определяет, является ли возвращаемое значение *bResult* равным 1, вычисляет размер диска, если это так, и выдает ошибку. Если это не так. Наконец, основная программа закончилась, как показано на рисунке 8.

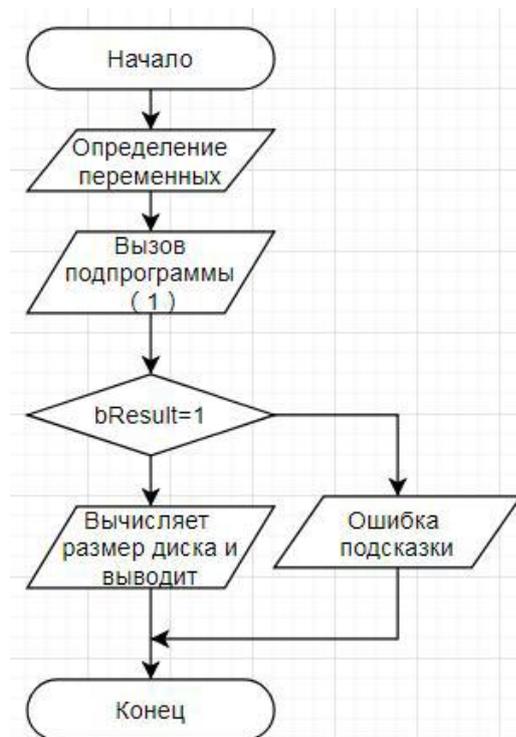


Рисунок 8 – Основная программа

Подпрограмма начинается с получения устройства ввода-вывода с помощью этой функции *CreateFile()*. Если не получается, Подскажет ошибку.

Если можно получить, просто можно прочитать и пошиться через функцию *DeviceIoControl()*. Затем можно установить адрес для чтения информации о диске с помощью функции *SetFilePointer()*. Затем можно прочитать содержимое диска с помощью функции *ReadFile()*. Итак, часть чтения – это MBR? Если это так, вызов подпрограмму (2), а затем искать EBR. Если нет, то часть, которая читается, – это EBR. Если часть чтения – это EBR, вызывающая подпрограмму (2). Наконец, удалить адресную константу и вернуть основную программу, как показано на рисунке 9.

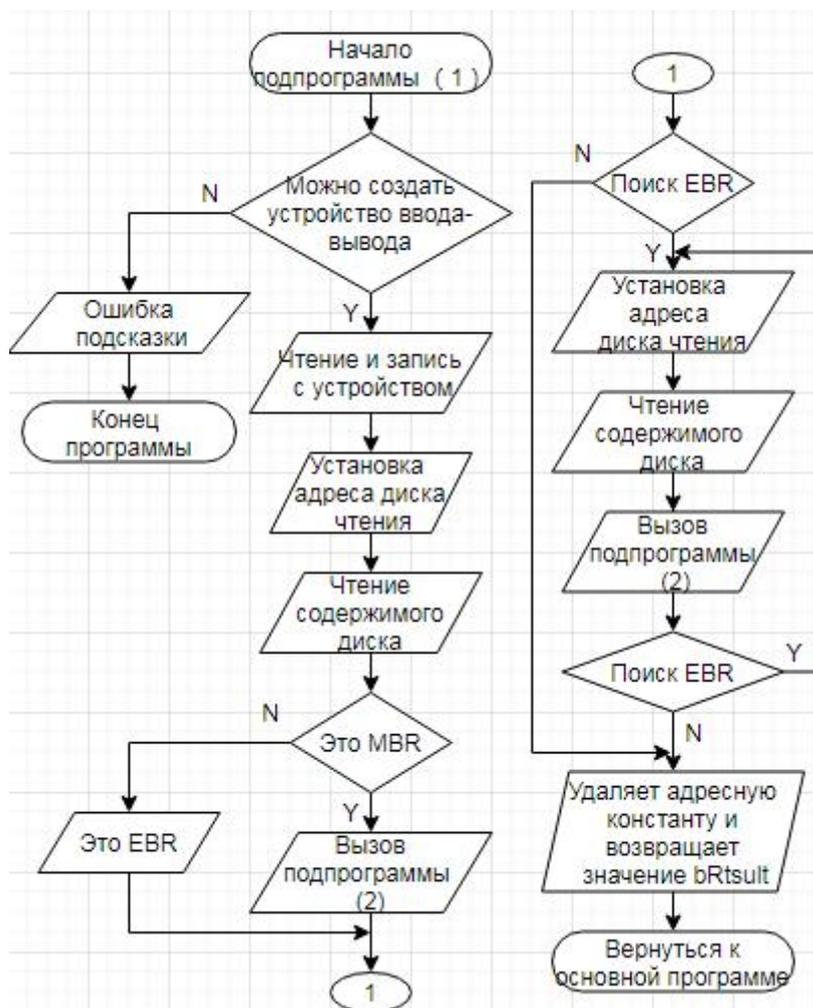


Рисунок 9 – Подпрограмма (1)

Подпрограмма (2) начинается с определения, является ли это MBR. Если да, то выходная информация: это MBR. Если нет, то выводит информацию: это EBR. Затем можно определить, является ли пятый байт 0x05 или 0x0F. Если они равны, это означает, что раздел является EBR в MBR. Определяет, равен ли пятый байт 0x00, если он не равен. Если он равен

0x00, это означает, что раздел пуст и возвращает подпрограмму (1). Если он не равен 0x00, выводится пятый байт. Затем можно прочитать исходный адрес и вывести его. Преобразует начальный адрес из шестнадцатеричного в десятичный и выводит его снова. Затем прочитайте этот размер раздела и выведите его. Преобразование размера раздела из шестнадцатеричного в десятичный вывод снова. Наконец, возвращает подпрограмму (1), как показано на рисунке 10.

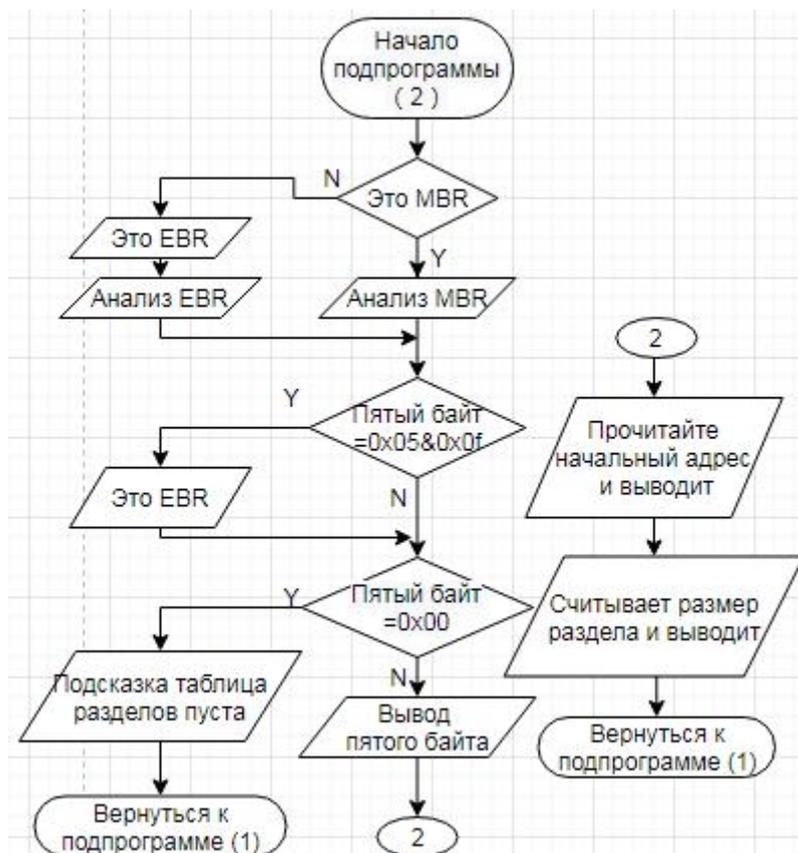


Рисунок 10 – Подпрограмма (2)

### 2.2.3 Анализ результатов

Результат запуска программы для анализа информации о диске согласуется с использованием *Winhex* для просмотра расчетов.

MBR часть показана на рисунке 11:

0000001B0	00 00 00 00 00 2C 44 63 32 7B 32 7B 00 00 80 01
0000001C0	01 00 07 FE FF FF 3F 00 00 00 B2 8C 7F 02 00 FE
0000001D0	FF FF 07 FE FF FF F1 8C 7F 02 C5 FA 3F 00 00 FE
0000001E0	FF FF 07 FE FF FF B6 87 BF 02 C5 FA 3F 00 00 FE
0000001F0	FF FF 0F FE FF FF 7B 82 FF 02 AA 53 00 02 55 AA
000000200	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Рисунок 11 – Таблица разделов

При запуске программы получены результаты, представленные на

рисунке 12.

Адрес и размер раздела:

1-й флаг таблицы разделов: 07

00 00 00 3f

Начальный адрес: 7e00

Размер: 02 7f 8c b2

Размер диска: 41913522 Секторов = 19,985925 ГБ

3-й флаг таблицы разделов: 07

02 bf 87 b6

Начальный адрес: 57f0f6c00

Размер: 00 3f fa c5

Размер диска: 4192965 Секторов = 1,999362 ГБ.

Это расширенная часть!

2-й флаг таблицы разделов: 07

02 7f 8c f1

Начальный адрес: 4ff19e200

Размер: 00 3f fa c5

Размер диска: 4192965 Секторов = 1,999362 ГБ.

4-й флаг таблицы разделов: 0f

02 ff 82 7b

Начальный адрес: 5ff04f600

Размер: 02 00 53 aa

Общий размер диска расширения: 33575850 Секторов = 16,010213 ГБ.

### Рисунок 12 – Результаты эксперимента

Мы сравниваем данные, полученные из экспериментальных результатов с реальными данными, как показано на рисунках 13, 14, 15, 16.

Partition	File System	Size
Partition 1 (C:)	NTFS	20.0 GB
Partition 2 (E:)	NTFS	2.0 GB
Partition 3 (F:)	NTFS	2.0 GB
Partition gap		31.5 KB
Partition 4 (G:)	NTFS	5.0 GB
Partition gap		31.5 KB
Partition 5 (H:)	NTFS	5.0 GB
Partition gap		31.5 KB
Partition 6 (I:)	NTFS	6.0 GB

Offset: 0 1 2 3 4 5 6 7 8 9 A B C D E F  
000007E00 EB 52 90 4E 54 46 53 20 20 20 20 00 02 08 00 00 @RNTFS

Рисунок 13 – Проверка информации о диске C, начальный адрес 7E00, размер: 20G

Partition	File System	Size
Partition 2 (E:)	NTFS	2.0 GB
Partition 3 (F:)	NTFS	2.0 GB
Partition gap		31.5 KB
Partition 4 (G:)	NTFS	5.0 GB
Partition gap		31.5 KB
Partition 5 (H:)	NTFS	5.0 GB
Partition gap		31.5 KB
Partition 6 (I:)	NTFS	6.0 GB

Offset: 0 1 2 3 4 5 6 7 8 9 A B C D E F  
4FF19E200 EB 52 90 4E 54 46 53 20 20 20 20 00 02 04 00 00 @RNTFS

Рисунок 14 – Проверка информации о диске E, начальный адрес 4FF19E200, размер: 2G

Partition 3 (F:)	NTFS	2.0 GB
Partition gap		31.5 KB
Partition 4 (G:)	NTFS	5.0 GB
Partition gap		31.5 KB
Partition 5 (H:)	NTFS	5.0 GB
Partition gap		31.5 KB
Partition 6 (I:)	NTFS	6.0 GB

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
57F0F6C00	EB	52	90	4E	54	46	53	20	20	20	20	00	02	04	00	00	#PNTFS

Рисунок 15 – Проверка информации о диске F, начальный адрес 57F0F6C00, размер: 2G

Partition gap		31.5 KB
Partition 4 (G:)	NTFS	5.0 GB
Partition gap		31.5 KB
Partition 5 (H:)	NTFS	5.0 GB
Partition gap		31.5 KB
Partition 6 (I:)	NTFS	6.0 GB

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
5FF04F600	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	

Рисунок 16 – Проверка EBR расширенного раздела, соответствующего четвертому DPT. Адрес расширенного раздела EBR: 5FF04F600

Содержание DPT первого EBR, как показано на рисунке 17:

5FF04F7B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	FE
5FF04F7C0	FF	FF	07	FE	FF	FF	3F	00	00	00	0E	12	A0	00	00	FE
5FF04F7D0	FF	FF	05	FE	FF	FF	4D	12	A0	00	4D	12	A0	00	00	00
5FF04F7E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
5FF04F7F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	55 AA

Рисунок 17 – Содержание DPT первого EBR

00 00 00 3F в адресной информации первой записи равно 63, указывая на первый логический диск. Начальный адрес этого логического диска рассчитывается как  $63 * 512 = 7E00H$ , необходимо добавить  $5FF04F600H$ , чтобы получить  $5FF057400H$ , это начальная позиция диска G. Информация о размере диска G равна  $0E 12 A0 00$ , то есть  $10490382$  сектора,  $10490382 * 512/1024/1024/1024 = 5$  ГБ, как показано на рисунке 18.

Partition 4 (G:)	NTFS	5.0 GB
Partition gap		31.5 KB
Partition 5 (H:)	NTFS	5.0 GB
Partition gap		31.5 KB
Partition 6 (I:)	NTFS	6.0 GB

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
5FF057400	EB	52	90	4E	54	46	53	20	20	20	20	00	02	08	00	00	#PNTFS

Рисунок 18 – Информация о размере диска G

Вторая запись — это 5-й байт, а содержимое - 05H, что указывает на то, что это расширенный раздел. (5-й байт 4-го сектора в предыдущем MBR равен 0FH, что также означает, что это расширенный раздел). Адресной информацией расширенного раздела является  $A0124DH$ , то есть 10490445-й

сектор (замененный байтом 140249A00H) относительно начальной позиции расширенного раздела 5FF04F600H хранит MBR следующего расширенного раздела. Эта позиция 140249A00H + 5FF04F600H = 73F299000H. В этой позиции действительно находится следующий логический диск, EBR диска H. Можно рассчитать таким же образом.

Сравнивая экспериментальный результат логического диска G и последующего расширенного раздела с реальными данными, как показано на рисунке 19, 20. вы обнаружите, что полученный результат верен.

Partition gap		31.5 KB
Partition 5 (H:)	NTFS	5.0 GB
Partition gap		31.5 KB
Partition 6 (I:)	NTFS	6.0 GB

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
73F299000	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Рисунок 19 – расширенный раздел (MBR)

1-й флаг таблицы разделов: 07

00 00 00 3f

Начальный адрес: 5ff057400

Размер: 00 a0 12 0e

Размер 1-ого логического диска: 10490382 Секторов = 5,002204 ГБ

Это расширенная часть!

2-й флаг таблицы разделов: 05

00 a0 12 4d

Начальный адрес: 73f299000

Размер: 00 a0 12 4d

Размер 1-ого логического диска: 10490445 Секторов = 5,002234 ГБ

Рисунок 20 – Результаты эксперимента

Второй контент EBR показано на рисунке 21:

73F2991B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	FE
73F2991C0	FF	FF	07	FE	FF	FF	3F	00	00	00	0E	12	A0	00	00	FE
73F2991D0	FF	FF	05	FE	FF	FF	9A	24	40	01	10	2F	CO	00	00	00
73F2991E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
73F2991F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	55	AA

Рисунок 21 – Второй контент EBR

Сравнивая экспериментальный результат логического диска H и

последующего расширенного раздела с реальными данными, как показано на рисунке 22, вы обнаружите, что полученный результат верен.

Адрес и размер раздела:

1-й флаг таблицы разделов: 07

00 00 00 3f

Начальный адрес: 5ff057400

Размер: 00 a0 12 0e

Размер второго логического диска: 10490382 секторов = 5,002204 ГБ.

Это расширенная часть!

2-й флаг таблицы разделов: 05

01 40 24 9a

Начальный адрес: 87f4e2a00

Размер: 00 c0 2f 10

Размер второго логического диска: 12594960 секторов = 6,005745 ГБ.

Рисунок 22 – Результаты эксперимента

Логическая схема разбиения показана на рисунке 23:

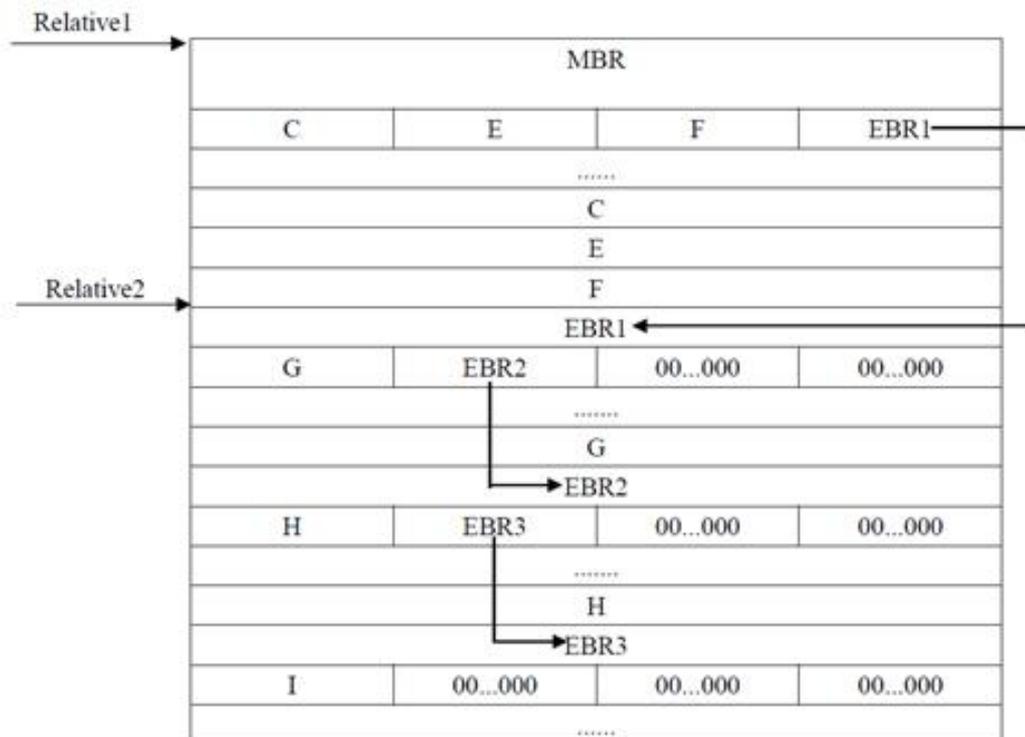


Рисунок 23 – Логическая схема разбиения

В результатах эксперимента с этими данными имеется информация о флаге таблицы разделов. Если эти данные равны 07, это означает, что файловой системой раздела диска является NTFS. Если эти данные равны 05 и 0F, это означает, что это расширенный раздел; если эти данные равны 0B, это FAT32.

### **3 Финансовый менеджмент, ресурсоэффективность и ресурсосбережение**

#### **Введение**

Цель данного раздела ВКР заключается в оценке перспективности разработки и планировании финансовой и коммерческой ценности конечного продукта, предлагаемого в рамках НИ. Коммерческая ценность определяется не только наличием более высоких технических характеристик над конкурентными разработками, но и тем, насколько быстро разработчик сможет ответить на следующие вопросы: будет ли продукт востребован на рынке, какова будет его цена, каков бюджет научного исследования, какое время будет необходимо для продвижения разработанного продукта на рынок.

Данный раздел, предусматривает рассмотрение следующих задач:

- оценка коммерческого потенциала разработки;
- планирование научно-исследовательской работы;
- расчет бюджета научно-исследовательской работы;
- определение ресурсной, финансовой, бюджетной эффективности исследования.

Цель данной ВКР – автоматизация распознавания типов файловых систем по их сигнатурам, при их частичном повреждении.

#### **3.1 Оценка коммерческого потенциала и перспективности проведения исследований с позиции ресурсоэффективности и ресурсосбережения**

##### **3.1.1 Анализ конкурентных технических решений**

В процесс исследования рассматривались две конкурирующие разработки:

- 1) Сканирование диска, чтобы найти точку повреждения файловой системы и восстановить.
- 2) форматирование диска и устанавливание файловой системы и восстановить.

В таблице 6 представлено сравнение разработок-конкурентов и разработки данного НИ с точки зрения технических и экономических критериев оценки эффективности.

Таблица 6 – Сравнение конкурентных технических решений (разработок)

Критерии оценки	Вес критерия	Баллы			Конкуренто-способность		
		Б <sub>ф</sub>	Б <sub>к1</sub>	Б <sub>к2</sub>	К <sub>ф</sub>	К <sub>к1</sub>	К <sub>к2</sub>
1	2	3	4	5	6	7	8
<b>Технические критерии оценки ресурсоэффективности</b>							
1. Проводить время	0,04	5	3	3	0,2	0,12	0,12
2. Простое и удобное управление	0,13	5	4	2	0,65	0,52	0,26
3. информационная безопасность	0,05	4	4	5	0,2	0,2	0,25
4. Тратить деньги	0,08	5	2	3	0,4	0,16	0,24
5. Объем восстановления данных	0,03	5	5	5	0,15	0,15	0,15
6. Эффективность работы	0,06	5	2	2	0,3	0,12	0,12
<b>Экономические критерии оценки эффективности</b>							
1. Реализация продукта	0,1	5	5	5	0,5	0,5	0,5
2. Уровень охвата рынка	0,03	4	5	3	0,12	0,15	0,09
3. Предполагаемая цена	0,08	5	5	3	0,4	0,4	0,24
4. Предполагаемый срок эксплуатации	0,08	5	3	4	0,4	0,24	0,32
5. Финансирование научной разработки конкурентных товаров и разработок	0,03	5	4	5	0,15	0,12	0,15
6. Срок выхода на рынок	0,03	5	5	5	0,15	0,15	0,15
7. Наличие сертификации разработки	0,03	5	4	4	0,15	0,12	0,12
<b>Итого</b>	<b>1</b>	<b>82</b>	<b>67</b>	<b>64</b>	<b>4,89</b>	<b>3,9</b>	<b>3,59</b>

Расчет конкурентоспособности, на примере стабильности срабатывания, определяется по формуле:

$$K = \sum B_i \cdot B_i = 0,1 \cdot 3 = 0,3,$$

где  $K$  – конкурентоспособность проекта;  $B_i$  – вес показателя (в долях единицы);  $B_i$  – балл показателя.

В результате проведенного анализа можно сделать вывод, что предлагаемое в данной ВКР решение обладает конкурентоспособностью.

### 3.1.2 SWOT-анализ

Для исследования внешней и внутренней среды проекта, в этой работе проведен SWOT-анализ с детальной оценкой сильных и слабых сторон исследовательского проекта, а также его возможностей и угроз.

На первом этапе была составлена SWOT-матрица, в которой были описаны слабые и сильные стороны проекта, а также выявлены возможности и угрозы для реализации проекта, которые возникли или могут возникнуть во внешней среде, приведены в таблице 7..

Таблица 7 – Матрица SWOT-анализа

<b>Сильные стороны</b>	<b>Слабые стороны</b>
С1. Тратить меньше денег	Сл1. Недостаток понимания компьютерного диска
С2. Восстановите информацию быстрее	Сл2. Долгосрочное устранение ошибок в программе
С3. Меньше ущерба для файловой структуры	Сл3. Высокие требования к знаниям программирования
С4. Высокая безопасность данных	Сл4. Отсутствие ссылок и материалов для соответствующих научных исследований.
С5. Очистить процесс	Сл5. Вероятность получения брака.
<b>Возможности</b>	<b>Угрозы</b>
В1. Помогите другим лучше понять информацию о диске	У1. Снижение стоимости разработок конкурентов.
В2. Помогите другим лучше выбрать файловую систему диска	У2. Появление зарубежных аналогов и более ранний их выход на рынок.
В3. Легко работать на разделе диска	
В4. Появление потенциального спроса на новые разработки.	

Таблица 8 – Интерактивная матрица проекта «Возможности проекта и сильные стороны»

<b>Сильные стороны проекта</b>						
<b>Возможности проекта</b>		C1	C2	C3	C4	C5
	B1	+	+	-	-	-
	B2	-	-	+	-	-
	B3	-	-	-	-	+
	B4	-	-	-	+	-

Таблица 9 – Интерактивная матрица проекта «Возможности проекта и слабые стороны»

<b>Слабые стороны проекта</b>						
<b>Возможности проекта</b>		Сл1	Сл2	Сл3	Сл4	Сл5
	B1	+	-	-	+	+
	B2	-	-	-	-	-
	B3	-	-	-	-	-
	B4	-	-	-	-	-

Таблица 10 – Интерактивная матрица проекта «Угрозы проекта и сильные стороны»

<b>Сильные стороны проекта</b>						
<b>Угрозы проекта</b>		C1	C2	C3	C4	C5
	У1	+	+	-	-	-
	У2	-	-	-	-	-

Таблица 11 – Интерактивная матрица проекта «Угрозы проекта и слабые стороны»

<b>Слабые стороны проекта</b>						
<b>Угрозы проекта</b>		Сл1	Сл2	Сл3	Сл4	Сл5
	У1	-	-	+	+	-
	У2	-	-	-	-	-

Результаты выполнения SWOT-анализа представлены в таблице 12.

Таблица 12 – Результаты SWOT-анализа

	<p><b>Сильные стороны научно-исследовательского проекта</b>  С1. Тратить меньше денег  С2. Восстановите информацию быстрее  С3. Меньше ущерба для файловой структуры  С4. Высокая безопасность данных  С5. Очистить процесс</p>	<p><b>Слабые стороны научно-исследовательского проекта</b>  Сл1. Недостаток понимания компьютерного диска  Сл2. Долгосрочное устранение ошибок в программе  Сл3. Высокие требования к знаниям программирования  Сл4. Отсутствие ссылок и материалов для соответствующих научных исследований.  Сл5. Вероятность получения брака.</p>
<p><b>Возможности</b>  В1. Помогите другим лучше понять информацию о диске  В2. Помогите другим лучше выбрать файловую систему диска  В3. Легко работать на разделе диска  В4. Появление потенциального спроса на новые разработки.</p>	<p><b>Направления развития</b>  В1С1С2. Хорошее понимание информации на диске может сократить расходы на исследования и быстрее восстановить информацию.  В2С3. Выбор правильной файловой системы может предотвратить повреждение файла.  В3С5. Легко работать в разделе диска может эффективно сократить рабочее время.  В4С4. Может удовлетворить потребность в личной информационной безопасности.</p>	<p><b>Сдерживающие факторы</b>  В1Сл1Сл4Сл5. Лучшее понимание информации о диске может удовлетворить потребности исследовательских материалов и предотвратить появление брака.</p>
<p><b>Угрозы</b>  У1. Снижение стоимости разработок конкурентов.  У2. Появление зарубежных аналогов и более ранний их выход на рынок.</p>	<p><b>Угрозы развития</b>  У1С1С2. Лучшее понимание информации о диске и повышение уровня программирования могут эффективно сократить расходы на исследования</p>	<p><b>Уязвимости:</b>  У1Сл3Сл4. Доступ к учебным и исследовательским материалам для улучшения дизайна программы может эффективно сократить расходы на исследования</p>

Результаты проведенного SWOT-анализа учтены в процессе дальнейшей разработки структуры работ, которые необходимо выполнить в научно-исследовательском проекте.

### 3.2 Планирование научно-исследовательских работ

#### 3.2.1 Структура работ в рамках научного исследования

Планирование работ предполагало определение структуры работ по проведению научного исследования, определение участников каждого вида работ, установление продолжительности работ, построение графика проведения исследований. Перечень этапов, работ и распределение исполнителей по выполнению НТР представлены в таблице 13.

Таблица 13 – Перечень этапов, работ и распределение исполнителей

Основные этапы	№ раб	Содержание работ	Должность исполнителя
Разработка технического задания	1	Составление и утверждение технического задания, утверждение плана-графика	Научный руководитель
	2	Календарное планирование выполнения ВКР	Студент-инженер, научный руководитель
Выбор способа решения поставленной задачи	3	Обзор научной литературы	Студент-инженер
	4	Выбор методов исследования	Студент-инженер
Теоретические и экспериментальные исследования	5	Планирование эксперимента	Студент-инженер, научный руководитель
	6	Проведение эксперимента	Студент-инженер
Обобщение и оценка результатов	7	Анализ полученных результатов НИР	Студент-инженер
	8	Оценка эффективности результатов	Научный руководитель
Оформление отчета по НИР (комплекта документации по ОКР)	9	Составление пояснительной записки	Студент-инженер
	10	Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Студент-инженер
	11	Социальная ответственность	Студент-инженер

### 3.2.2 Разработка графика проведения научного исследования

Определение трудоемкости выполнения научного исследования проведено экспертным путем в человеко-днях. Для определения ожидаемого (среднего) значения трудоемкости  $t_{ожі}$  использована следующая формула:

$$t_{ожі} = \frac{3t_{\min i} + 2t_{\max i}}{5}, \quad (1)$$

где  $t_{ожі}$  – ожидаемая трудоемкость выполнения  $i$ -ой работы, человеко-дни;

$t_{\min i}$  – минимально возможная трудоемкость выполнения заданной  $i$ -ой работы, человеко-дни;

$t_{\max i}$  – максимально возможная трудоемкость выполнения заданной  $i$ -ой работы, человеко-дни.

Исходя из рассчитанной ожидаемой трудоемкости работ, была определена продолжительность каждого этапа работы (в рабочих днях  $T_p$ ), учитывающая параллельность выполнения работ несколькими исполнителями, а также возможность выполнения нескольких видов работ в один временной промежуток. Далее с помощью формулы 2 рассчитана продолжительность одной работы в рабочих днях:

$$T_{p_i} = \frac{t_{ожі}}{Ч_i}, \quad (2)$$

где  $T_{p_i}$  – продолжительность одной работы, рабочие дни;

$t_{ожі}$  – ожидаемая трудоемкость выполнения одной работы, человеко-дни;

$Ч_i$  – численность исполнителей, выполняющих одновременно одну и ту же работу на данном этапе, чел.

Расчеты временных показателей проведения научного исследования обобщены в таблице 14.

Таблица 14 – Временные показатели проведения научного исследования

Название работы	Трудоёмкость работ						Длительность работ в рабочих днях $T_{pi}$	Длительность работ в календарных днях $T_{ki}$
	$t_{min}$ , чел-дни		$t_{max}$ , чел-дни		$t_{ож}$ , чел-дни			
	Исп.1	Исп.2	Исп.1	Исп.2	Исп.1	Исп.2		
1	2	3	4	5	6	7	8	9
1. Составление и утверждение технического задания, утверждение плана-графика	1		4		2,2		2,2	3
2. Календарное планирование выполнения ВКР	1	1	4	4	2,2	2,2	2,2	3
3. Обзор научной литературы		5		10		7	7	11
4. Выбор методов исследования		3		5		3,8	3,8	6
5. Планирование эксперимента		25		30		27	27	41
6. Проведение эксперимента		20		30		24	24	36
7. Анализ полученных результатов НИР		4		8		5,6	5,6	8
8. Оценка эффективности результатов	4		8		5,6		5,6	8
9. Составление пояснительной записки		4		10		6,4	6,4	10
<b>Итого</b>	6	62	16	97	10	76	83,8	126

Примечание: Исп. 1 – научный руководитель, Исп. 2 – студент-инженер.

На основе таблицы составлен календарный план-график выполнения проекта с использованием диаграммы Ганта (таблица 15).

Таблица 15 – Диаграмма Ганта

№	Вид работ	Исп	Т <sub>кi</sub> , кал. дн.	Продолжительность работ														
				февр			март			апр			май					
				1	2	3	1	2	3	1	2	3	1	2	3			
1	Составление и утверждение технического задания, утверждение плана-графика	Исп1	3	■														
2	Календарное планирование выполнения ВКР	Исп1 Исп2	3	■	■													
3	Обзор научной литературы	Исп2	11		■	■												
4	Выбор методов исследования	Исп2	6			■												
5	Планирование эксперимента	Исп2	41			■	■	■	■	■								
6	Проведение эксперимента	Исп2	36							■	■	■	■	■				
7	Анализ полученных результатов НИР	Исп2	8										■	■				
8	Оценка эффективности результатов	Исп1	8											■	■			
9	Составление пояснительной записки	Исп2	10												■			

■ Примечание:– Исп 1 (научный руководитель), ■ – Исп 2 (студент-инженер)

### 3.2.3 Бюджет научно-технического исследования

При планировании бюджета научно-технического исследования учитывались все виды расходов, связанных с его выполнением.

#### Расчет амортизации специального оборудования

Расчет амортизации производится на находящееся в использовании оборудование. В итоговую стоимость проекта входят отчисления на амортизацию за время использования оборудования в статье накладных расходов (таблица 16).

Таблица 16 – Затраты на оборудование

№	Наименование оборудования	Кол-во, шт.	Срок полезного использования, лет	Цена единицы оборудования, тыс. руб.	Общая стоимость оборудования, тыс. руб.
1	Lenovo Компьютер	1	10	25	25
<b>Итого:</b>					25 тыс. руб.

Расчет амортизации проводится следующим образом:

Норма амортизации:

$$H_A = \frac{1}{n}, \quad (3)$$

где  $n$  – срок полезного использования в количестве лет.

Амортизация:

$$A = \frac{H_A I}{12} \cdot m, \quad (4)$$

где  $I$  – итоговая сумма, тыс. руб.;  $m$  – время использования, мес.

Рассчитаем амортизацию для Компьютера, с учётом, что срок полезного использования 10 лет:

$$H_A = \frac{1}{n} = \frac{1}{10} = 0,1. \quad (5)$$

Общую сумму амортизационных отчислений находим следующим образом:

Компьютер:

$$A_1 = \frac{H_A I}{12} \cdot m = \frac{0,1 \cdot 25000}{12} \cdot 4 = 834 \text{ руб.} \quad (6)$$

Суммарные затраты амортизационных отчислений:

$$A = A_1 = 834 \text{ руб.} \quad (7)$$

### **Основная заработная плата исполнителей темы**

В данном разделе рассчитывается заработная плата инженера и руководителя, помимо этого необходимо рассчитать расходы по заработной плате, определяемые трудоемкостью проекта и действующей системой оклада.

Основная заработная плата  $Z_{осн}$  одного работника рассчитывается по следующей формуле:

$$Z_{осн} = Z_{дн} \cdot T_p, \quad (8)$$

где  $Z_{дн}$  – среднедневная заработная плата, руб.;  $T_p$  – продолжительность работ, выполняемых работником, раб.дн. (таблица 8).

Среднедневная заработная плата рассчитывается по формуле:

Для шестидневной рабочей недели (рабочая неделя руководителя):

$$Z_{дн} = \frac{Z_m \cdot M}{F_d} = \frac{51285 \cdot 10,3}{246} = 2147,3 \text{ руб.}, \quad (9)$$

где  $Z_m$  – должностной оклад работника за месяц;  $F_o$  – действительный годовой фонд рабочего времени исполнителей, раб.дн.;  $M$  – количество месяцев работы без отпуска в течение года:

– при отпуске в 28 раб. дня –  $M = 11,2$  месяца, 5-дневная рабочая неделя;

– при отпуске в 56 раб. дней –  $M = 10,3$  месяца, 6-дневная рабочая неделя.

Для пятидневной рабочей недели (рабочая неделя инженера):

$$Z_{он} = \frac{Z_m \cdot M}{F_o} = \frac{33150 \cdot 11,2}{213} = 1743,1 \text{ руб.} \quad (10)$$

Должностной оклад работника за месяц:

– для руководителя:

$$Z_m = Z_{мс} \cdot (1 + k_{пр} + k_o) k_p = 26300 \cdot (1 + 0,3 + 0,2) \cdot 1,3 = 51285 \text{ руб.} \quad (11)$$

– для инженера:

$$Z_m = Z_{мс} \cdot (1 + k_{пр} + k_o) k_p = 17000 \cdot (1 + 0,3 + 0,2) \cdot 1,3 = 33150 \text{ руб.}, \quad (12)$$

где  $Z_{мс}$  – заработная плата, согласно тарифной ставке, руб.;  $k_{пр}$  – премиальный коэффициент, равен 0,3;  $k_o$  – коэффициент доплат и надбавок, равен 0,2;  $k_p$  – районный коэффициент, равен 1,3 (для г. Томска).

Таблица 17 – Баланс рабочего времени исполнителей

Показатели рабочего времени	Руководитель	Инженер
Календарное число дней	365	365
Количество нерабочих дней	52/14	104/14
- выходные дни		
- праздничные дни		
Потери рабочего времени	48/5	24/10
- отпуск		
- невыходы по болезни		
Действительный годовой фонд рабочего времени	246	213

Таблица 18 – Расчет основной заработной платы исполнителей

Исполнители НИ	$Z_{nc}, руб$	$k_{np}$	$k_{\partial}$	$k_p$	$Z_m, руб$	$Z_{\partial n}, руб$	$T_p, раб.дн.$	$Z_{осн}, руб$
Руководитель	26300	0,3	0,2	1,3	51285	2147,3	13	27914,9
Инженер	17000	0,3	0,2	1,3	33150	1743,1	47	81925,7
Итого:								109840,6

Дополнительная заработная плата определяется по формуле:

– для руководителя:

$$Z_{\partial n} = K_{\partial n} * Z_{осн} = 0,15 * 27914,9 = 4187,24 \text{ руб.} \quad (13)$$

– для инженера:

$$Z_{\partial n} = K_{\partial n} * Z_{осн} = 0,15 * 81925,7 = 12288,86 \text{ руб.} \quad (14)$$

где  $k_{\partial n}$  – коэффициент дополнительной заработной платы (на стадии проектирования принимаем равным 0,15).

### Отчисления во внебюджетные фонды (страховые отчисления)

Отчисления во внебюджетные фонды определяется по формуле:

Для руководителя:

$$Z_{внеб} = K_{внеб} (Z_{осн} + Z_{\partial n}) = 0,3 * (27914,9 + 4187,24) = 9630,6 \text{ руб.} \quad (15)$$

Для инженера:

$$Z_{внеб} = K_{внеб} (Z_{осн} + Z_{\partial n}) = 0,3 * (81925,7 + 12288,86) = 28264,37 \text{ руб.} \quad (16)$$

где  $k_{внеб}$  – коэффициент отчислений на уплату во внебюджетные фонды (пенсионный фонд, фонд ОМС и социальное страхование). Общая ставка взносов составляет в 2020 году – 30% (ст. 425, 426 НК РФ).

### Накладные расходы

Накладными расходами учитываются прочие затраты организации, такие как: печать и ксерокопирование проектировочных документов, оплата услуг связи.

Накладные расходы в целом:

$$\begin{aligned} Z_{накл} &= (\text{сумма статей} / 4) * k_{np} = \\ &= (834 + 109840,6 + 16476,1 + 37894,97) * 0,2 = 33009,13 \text{ руб.} \end{aligned} \quad (17)$$

где  $k_{np}$  – коэффициент, учитывающий накладные расходы. Величина коэффициента принимается равной 0,2.

На основании полученных данных по отдельным статьям затрат составляется калькуляция плановой себестоимости НИ «Автоматизация распознавания типов файловых систем по их сигнатурам, при их частичном повреждении» по форме, приведенной в таблице 19.

Таблица 19 – Группировка затрат по статьям

Амортизация	Основная заработная плата	Дополнительная заработная плата	Отчисления на социальные нужды	Итого без накладных расходов	Накладные расходы	Итого бюджетная стоимость
834	109840.6	16476.1	37894,97	165045,67	33009,13	198054,8

### 3.3 Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования

Для определения эффективности исследования рассчитан интегральный показатель эффективности научного исследования путем определения интегральных показателей финансовой эффективности и ресурсоэффективности.

Интегральный показатель финансовой эффективности научного исследования получен в ходе оценки бюджета затрат трех вариантов исполнения научного исследования. Для этого наибольший интегральный показатель реализации технической задачи принят за базу расчета (как знаменатель), с которым соотносятся финансовые значения по всем вариантам исполнения.

В качестве аналогов данной НТР рассмотрены:

- 1) Сканирование диска, чтобы найти точку повреждения файловой системы и восстановить.
- 2) форматирование диска и переустановка файловой системы и восстановить.

Интегральный финансовый показатель вариантов выполнения проектируемого объекта определялся по формуле:

$$I_{\text{финр}}^{\text{Вар.}i} = \frac{\Phi_{pi}}{\Phi_{\text{max}}}, \quad (18)$$

где  $I_{\text{финр}}^{\text{Вар.}i}$  – интегральный финансовый показатель разработки;

$\Phi_{pi}$  – стоимость  $i$ -го варианта исполнения;

$\Phi_{\max}$  – максимальная стоимость исполнения научно-исследовательского проекта.

$$\Phi_{\text{текущ.проект}} = 198054,8 \text{ руб}, \Phi_{\text{исп.1}} = 231090,4 \text{ руб}, \Phi_{\text{исп.2}} = 209841,5 \text{ руб}$$

$$\begin{aligned} I_{\text{финр}}^{\text{тек.про}} &= \frac{\Phi_{\text{тек.про.}}}{\Phi_{\max}} = \frac{198054,8}{231090,4} = 0,86; \\ I_{\text{финр}}^{\text{исп.2}} &= \frac{\Phi_{\text{исп.2}}}{\Phi_{\max}} = \frac{231090,4}{231090,4} = 1; \\ I_{\text{финр}}^{\text{исп.3}} &= \frac{\Phi_{\text{исп.3}}}{\Phi_{\max}} = \frac{209841,5}{231090,4} = 0,91. \end{aligned} \quad (19)$$

В результате расчета консолидированных финансовых показателей по трем вариантам разработки вариант 1 (текущий проект) с меньшим перевесом признан считается более приемлемым с точки зрения финансовой эффективности.

**Интегральный показатель ресурсоэффективности** вариантов выполнения НТР ( $I_{pi}$ ) определен путем сравнительной оценки их характеристик, распределенных с учетом весового коэффициента каждого параметра (таблица 20).

Таблица 20 – Сравнительная оценка характеристик вариантов НТР

Объект исследования Критерии	Весовой коэффициент параметра	Текущий проект	Вар.2	Вар.3
1. Удобство в эксплуатации	0,3	4	3	4
2. Стабильность работы	0,25	5	5	5
3. Стоимость исследования	0,25	5	3	4
4. Ремонтпригодность	0,2	4	4	4
ИТОГО	1	4,5	3,7	4,25

$$I_{p1} = 0,3*4 + 0,25*5 + 0,25*5 + 0,2*4 = 4,5;$$

$$I_{p1} = 0,3*3 + 0,25*5 + 0,25*3 + 0,2*4 = 3,7; \quad (20)$$

$$I_{p1} = 0,3*4 + 0,25*5 + 0,25*4 + 0,2*4 = 4,25.$$

На основании полученных интегрального финансового показателя и интегрального показателя ресурсоэффективности был рассчитан интегральный показатель эффективности вариантов исполнения разработки ( $I_{вари}$ ) по формуле:

$$I_{исп.i} = \frac{I_{p-исп.i}}{I_{финр}^{исп.i}} \quad (21)$$

$$I_{вар1} = \frac{4,5}{0,86} = 5,23, \quad I_{вар2} = \frac{3,7}{1} = 3,7, \quad I_{вар3} = \frac{4,25}{0,91} = 4,67. \quad (22)$$

Далее интегральные показатели эффективности каждого варианта НТР сравнивались с интегральными показателями эффективности других вариантов с целью определения сравнительной эффективности проекта (таблица 21).

Таблица 21 – Сравнительная эффективность разработки

№ п/п	Показатели	Текущий проект	Вар. 2	Вар. 3
1	Интегральный финансовый показатель разработки	0,86	1	0,91
2	Интегральный показатель ресурсоэффективности разработки	4,5	3,7	4,25
3	Интегральный показатель эффективности	5,23	3,7	4,67
	ИТОГО	10,59	8,4	9,83
4	Сравнительная эффективность вариантов исполнения	Вар.1/Вар.1 = 1,0	Вар. 2/Вар. 1 = 0,79	Вар. 3/Вар. 1 = 0,93

Сравнение среднего интегрального показателя сопоставляемых вариантов позволило сделать вывод о том, что наиболее финансово- и ресурсоэффективным является вариант 1 (Текущий проект), т. к. показатель его сравнительной эффективности по отношению к каждому из сравниваемых вариантов больше 1.

### Выводы по разделу

В результате выполнения целей раздела можно сделать следующие выводы:

1. Результатом проведенного анализа конкурентных технических решений является выбор одного из вариантов реализации НТР как наиболее предпочтительного и рационального по сравнению с остальными.

2. При проведении планирования был разработан план-график выполнения этапов работ для руководителя и инженера, позволяющий

оценить и спланировать рабочее время исполнителей. Определено следующее: общее количество календарных дней для выполнения работы – 126 дней; общее количество календарных дней, в течение которых работал инженер, – 104; общее количество календарных дней, в течение которых работал руководитель, – 22;

3. Составлен бюджет проектирования, позволяющий оценить затраты на реализацию проекта, которые составляют 198054,8 руб;

4. По факту оценки эффективности ИР, можно сделать выводы:

1) значение интегрального финансового показателя ИР составляет 0,86, что является показателем того, что ИР является финансово выгодной по сравнению с аналогами;

2) значение интегрального показателя ресурсоэффективности ИР составляет 4,5, по сравнению с 3,7 и 4,25;

3) значение интегрального показателя эффективности ИР составляет 5,23, по сравнению с 3,7 и 4,67, и является наиболее высоким, что означает, что техническое решение, рассматриваемое в ИР, является наиболее эффективным вариантом исполнения.

## **4 Социальная ответственность**

### **Введение**

Социальная ответственность – ответственность отдельного ученого и научного сообщества перед обществом. Первостепенное значение при этом имеет безопасность применения технологий, которые создаются на основе достижений науки, предотвращение или минимизация возможных негативных последствий их применения, обеспечение безопасного как для испытуемых, как и для окружающей среды проведения исследований.

В ходе данной работы разработка алгоритма автоматического определения типа файловой системы по сигнатуре. Работа выполнялась в общежитии. Все работы выполнялись с использования компьютера. Раздел также включает в себя оценку условий труда на рабочем месте, анализ вредных и опасных факторов труда, разработку мер защиты от них.

#### **4.1 Правовые и организационные вопросы обеспечения безопасности**

Согласно СанПиН 2.2.2/2.4.1340-03, ФЗ – 197 [9] каждый работник имеет право на:

- рабочее место, соответствующее требованиям охраны труда;
- обязательное социальное страхование от несчастных случаев на производстве и профессиональных заболеваний в соответствии с федеральным законом;
- получение достоверной информации от работодателя, соответствующих государственных органов и общественных организаций об условиях и охране труда на рабочем месте, о существующем риске повреждения здоровья, а также о мерах по защите от воздействия вредных и (или) опасных производственных факторов;
- отказ от выполнения работ в случае возникновения опасности для его жизни и здоровья вследствие нарушения требований охраны труда, за исключением случаев, предусмотренных федеральными законами, до устранения такой опасности;
- обеспечение средствами индивидуальной и коллективной защиты в соответствии с требованиями охраны труда за счет средств работодателя;
- обучение безопасным методам и приемам труда за счет средств работодателя;
- личное участие или участие через своих представителей в рассмотрении вопросов, связанных с обеспечением безопасных условий труда на его рабочем месте, и в расследовании происшедшего с ним несчастного случая на производстве или профессионального заболевания;
- внеочередной медицинский осмотр в соответствии с медицинскими рекомендациями с сохранением за ним места работы (должности) и среднего заработка во время прохождения указанного медицинского осмотра;
- гарантии и компенсации, установленные в соответствии с настоящим Кодексом, коллективным договором, соглашением, локальным

нормативным актом, трудовым договором, если он занят на работах с вредными и (или) опасными условиями труда.

– повышенные или дополнительные гарантии и компенсации за работу на работах с вредными и (или) опасными условиями труда могут устанавливаться коллективным договором, локальным нормативным актом с учетом финансово-экономического положения работодателя.

## **4.2 Производственная безопасность**

### **4.2.1 Анализ условий труда на рабочем месте**

Основным объектом в производственных условиях является рабочее место, представляющее собой в общем случае пространство, в котором может находиться человек при выполнении производственного процесса. Рабочее место является основной подсистемой производственного процесса.

В соответствии с ГОСТ 12.2.032-78 ССБТ [10], сделаны следующие выводы:

Модульными размерами рабочей поверхности стола для ПЭВМ, на основании которых должны рассчитываться конструктивные размеры, следует считать: ширину 800 и 1000 мм, глубину 800 и 1000 мм при нерегулируемой его высоте, равной 725 мм. Рабочий стол должен иметь пространство для ног высотой не менее 600 мм, шириной – не менее 500 мм, глубиной на уровне колен – не менее 450 мм и на уровне вытянутых ног – не менее 650 мм.

Конструкция рабочего стула должна обеспечивать:

- ширину и глубину поверхности сиденья не менее 400 мм;
- поверхность сиденья с закругленным передним краем;
- регулировку высоты поверхности сиденья в пределах 400-550 мм и углам наклона вперед до 15 град, и назад до 5 град.;
- высоту опорной поверхности спинки  $300 \pm 20$  мм, ширину - не менее 380 мм и радиус кривизны горизонтальной плоскости - 400 мм;

- угол наклона спинки в вертикальной плоскости в пределах  $\pm 30$  градусов;
- регулировку расстояния спинки от переднего края сиденья в пределах 260-400 мм;
- стационарные или съемные подлокотники длиной не менее 250 мм и шириной - 50-70 мм;
- регулировку подлокотников по высоте над сиденьем в пределах  $230 \pm 30$  мм и внутреннего расстояния между подлокотниками в пределах 350-500 мм.

Рабочее место пользователя ПЭВМ следует оборудовать подставкой для ног, имеющей ширину не менее 300 мм, глубину не менее 400 мм, регулировку по высоте в пределах до 150 мм и по углу наклона опорной поверхности подставки до  $20^\circ$ . Поверхность подставки должна быть рифленой и иметь по переднему краю бортик высотой 10 мм. Клавиатуру следует располагать на поверхности стола на расстоянии 100-300 мм от края, обращенного к пользователю или на специальной, регулируемой по высоте рабочей поверхности, отделенной от основной столешницы. Окна в помещениях, где эксплуатируется вычислительная техника, преимущественно должны быть ориентированы на север и северо - восток.

Мое рабочее место – спальня. На рабочем месте есть холодильник. Окно выходит на дорогу. Фактор, который может создать объект исследования, это электромагнитное излучение(ЭИ) от компьютера. Перечень всех опасных и вредных факторов, которые могут возникнуть на рабочем месте при проведении исследований, приведены в таблице 22.

Таблица 22 — Вредные и опасные факторы

Факторы (ГОСТ 12.0.003-2015) [11]	Этапы работ			Нормативные документы
	<i>Разработка</i>	<i>Изготовление</i>	<i>Эксплуатация</i>	
1. электромагнитное излучение	+	+	+	СанПиН 2.2.2/2.4.1340-03 [12]
2. Отклонение показателей микроклимата	+	+	+	СанПиН 2.2.4.548-96 [13]
3. Превышение уровня шума		+	+	СН 2.2.4/2.1.8.562-96 [14]
4. Недостаточная освещенность рабочей зоны	+	+	+	СанПиН 2.2.1/2.1.1.1278-03 [15]

#### 4.2.2 электромагнитное излучение

По результатам исследований, проведенных в 2005 году Центром электромагнитной безопасности, в России лишь 15 % компьютеров полностью удовлетворяют международным нормам, 31 % – частично, 54 % – никак не соответствуют международным стандартам и требуют защиты как пользователя, так и окружающих людей.

Исследования показали, что продолжительное влияние ЭИ, даже относительно слабого уровня, может вызвать: раковые заболевания; потерю памяти, болезни Паркинсона и Альцгеймера; изменение гормонального статуса организма; повысить склонность к самоубийству.

Для того чтобы избежать негативного воздействия от электромагнитного излучения необходимо следовать основным нормам, описанным в СанПиН 2.2.2/2.4.1340-03 [12]. Временные допустимые уровни ЭМП, создаваемых ПЭВМ указаны в таблице 23.

Таблица 23 — Временные допустимые уровни ЭМП, создаваемых ПЭВМ

Наименование параметров		ВДУ ЭМП
Напряженность электрического поля	В диапазоне частот от 5 Гц до 2 кГц	25 В/м
	В диапазоне частот от 2 кГц до 400 кГц	2,5 В/м
Плотность магнитного поля	В диапазоне частот от 5 Гц до 2 кГц	250 нТл
	В диапазоне частот от 2 кГц до 400 кГц	25 нТл
Напряженность электростатического поля		25 кВ/м

#### 4.2.3 Отклонение показателей микроклимата

В таблице 24 приведены оптимальные величины показателей микроклимата на рабочих местах производственных помещений в соответствии с временем года и категории тяжести работ согласно требованиям СанПиН 2.2.4.548-96 [13].

Таблица 24 — Оптимальные показатели микроклимата

Период года	Температура воздуха, °С	Температура поверхностей, °С	Относительная влажность воздуха, %	Скорость движения воздуха, м/с
Холодный	(22 - 24)	(21 - 25)	(60 - 40)	0,1
Тёплый	(23 - 25)	(22 - 26)	(60 - 40)	0,1

В соответствии с характеристикой помещения определен расход свежего воздуха согласно СанПиН 2.2.4.548-96 и приведен в таблице 25.

Таблица 25 — Расход свежего воздуха по СанПиН 2.2.4.548-96 [13]

Характеристика помещения	Объемный расход подаваемого в помещение свежение свежего воздуха, м <sup>3</sup> /на одного человека в час
Объем до 20 м <sup>3</sup> на человека	Не менее 30
(20 - 40) м <sup>3</sup> на человека	Не менее 20
Более 40 м <sup>3</sup> на человека	Естественная вентиляция

#### 4.2.4 Превышение уровня шума

Гудящий звук холодильника на работе и звук транспортных средств, едущих по дороге за окном, оказывают на меня большое влияние. В соответствии с ГОСТ 12.1.003-2015 [11], допустимые уровни звукового

давления в октавных полосах частот, уровни звука и эквивалентные уровни звука на рабочем месте приведены в таблице 26.

Таблица 26 — Допустимые уровни звукового давления

Помещения и рабочие места	Уровень звукового давления, дБ, в октавных полосах среднегеометрическими частотами, Гц					Уровень звука, дБА
	63	125	250	500	1000	
Помещения управления, рабочие комнаты	79	70	68	55	50	60

#### 4.2.5 Недостаточная освещенность рабочей зоны

Согласно СНиП 23-05-95 [18] в лаборатории, где происходит периодическое наблюдение за ходом производственного процесса при постоянном нахождении людей в помещении освещенность при системе общего освещения не должна быть ниже 300 Лк.

Правильно спроектированное и выполненное освещение обеспечивает высокий уровень работоспособности, оказывает положительное психологическое действие на человека и способствует повышению производительности труда.

На рабочей поверхности должны отсутствовать резкие тени, которые создают неравномерное распределение поверхностей с различной яркостью в поле зрения, искажает размеры и формы объектов различия, в результате повышается утомляемость и снижается производительность труда.

Для защиты от слепящей яркости видимого излучения (факел плазмы в камере с катализатором) применяют защитные очки, щитки, шлемы. Очки на должны ограничивать поле зрения, должны быть легкими, не раздражать кожу, хорошо прилегать к лицу и не покрываться влагой.

Расчёт общего равномерного искусственного освещения горизонтальной рабочей поверхности выполняется методом коэффициента светового потока, учитывающим световой поток, отражённый от потолка и стен. Длина помещения  $A = 9$  м, ширина  $B = 6$  м, высота = 3,5 м. Высота

рабочей поверхности над полом  $h_p = 1,0$  м. Согласно СНиП 23-05-95 необходимо создать освещенность не ниже 150 лк, в соответствии с разрядом зрительной работы.

Площадь помещения:

$$S = A \times B,$$

где  $A$  – длина, м;

$B$  – ширина, м.

$$S = 9 \times 5 = 45 \text{ м}^2$$

Коэффициент отражения свежепобеленных стен с окнами, без штор  $\rho_c = 50\%$ , свежепобеленного потолка  $\rho_{II} = 70\%$ . Коэффициент запаса, учитывающий загрязнение светильника, для помещений с малым выделением пыли равен  $K_z = 1,5$ . Коэффициент неравномерности для люминесцентных ламп  $Z = 1,1$ .

Выбираем лампу дневного света ЛД-40, световой поток которой равен  $\Phi_{ЛД} = 2600$  Лм.

Выбираем светильники с люминесцентными лампами типа ОДОР-2-40. Этот светильник имеет две лампы мощностью 40 Вт каждая, длина светильника равна 1227 мм, ширина – 265 мм.

Интегральным критерием оптимальности расположения светильников является величина  $\lambda$ , которая для люминесцентных светильников с защитной решёткой лежит в диапазоне 1,1–1,3. Принимаем  $\lambda = 1,1$ , расстояние светильников от перекрытия (свес)  $h_c = 0,3$  м.

Высота светильника над рабочей поверхностью определяется по формуле:

$$h = h_n - h_p,$$

где  $h_n$  – высота светильника над полом, высота подвеса,

$h_p$  – высота рабочей поверхности над полом.

Наименьшая допустимая высота подвеса над полом для двухламповых светильников ОДОР:  $h_n = 3,5$  м.

Высота светильника над рабочей поверхностью определяется по формуле:

$$h = H - h_p - h_c = 3,5 - 1 - 0,5 = 2,0 \text{ м}$$

Расстояние между соседними светильниками или рядами определяется по формуле:

$$L = \lambda * h = 1,1 * 2 = 2,2 \text{ м}$$

Число рядов светильников в помещении:

$$N_b = \frac{B}{L} = \frac{5}{2,2} = 2,27 \approx 2$$

Число светильников в ряду:

$$N_a = \frac{A}{L} = \frac{9}{2,2} = 4,09 \approx 4$$

Общее число светильников:

$$N = N_a * N_b = 4 * 2 = 8$$

Расстояние от крайних светильников или рядов до стены определяется по формуле:

$$l = \frac{L}{3} = \frac{2,2}{3} = 0,7 \text{ м}$$

Размещаем светильники в четыре ряда. На рисунке 24 изображен план помещения и размещения светильников с люминесцентными лампами.

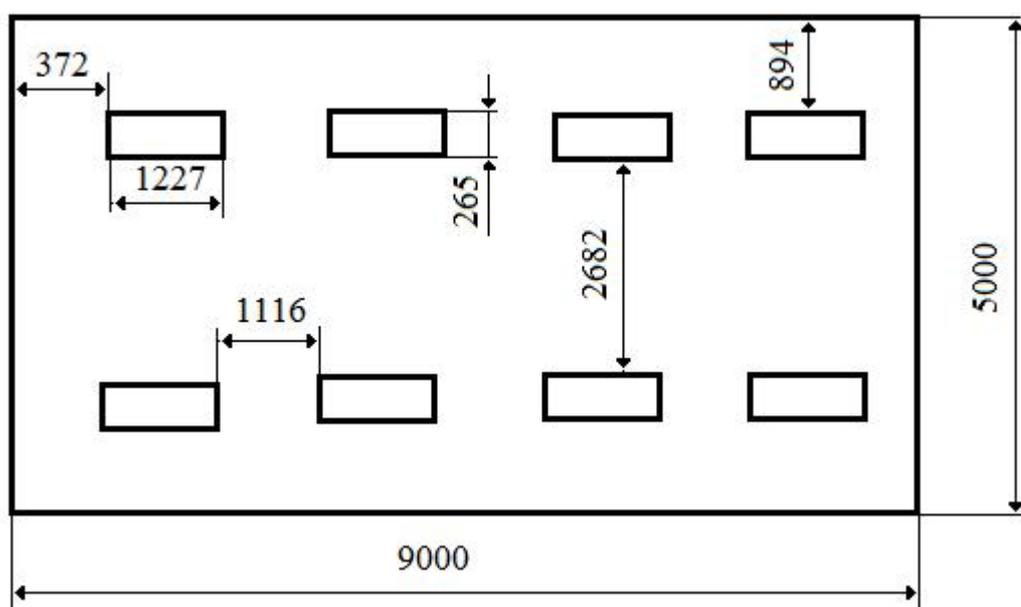


Рисунок 24 – План помещения и размещения светильников с люминесцентными лампами.

Индекс помещения определяется по формуле:

$$i = \frac{A \cdot B}{h \cdot (A + B)} = \frac{9 \cdot 5}{2 \cdot (9 + 6)} = 1,6$$

Коэффициент использования светового потока, показывающий какая часть светового потока ламп попадает на рабочую поверхность, для светильников типа ОДОР с люминесцентными лампами при  $\rho_{\text{л}} = 70\%$ ,  $\rho_{\text{с}} = 50\%$  и индексе помещения  $i = 1,6$  равен  $\eta = 0,47$ .

Учитывая, что в каждом светильнике установлено две лампы, общее число ламп в помещении  $N = 16$ . Потребный световой поток группы люминесцентных ламп светильника определяется по формуле:

$$\Phi_{\text{л}} = (E \cdot S \cdot K_3 \cdot Z) / N \cdot \eta = (300 \cdot 45 \cdot 1,5 \cdot 1,1) / (16 \cdot 0,47) = 2962,1 \text{ лм}$$

выбираем ближайшую стандартную лампу – ЛТБ 40 Вт с потоком 2850 лм. Делаем проверку выполнения условия:

$$-10\% \leq \frac{\Phi_{\text{лд}} - \Phi_{\text{л}}}{\Phi_{\text{лд}}} \cdot 100\% \leq 20\%;$$

$$\frac{\Phi_{\text{лд}} - \Phi_{\text{л}}}{\Phi_{\text{лд}}} \cdot 100\% = \frac{2850 - 2962,1}{2850} = -3\%$$

Таким образом, мы получили, что необходимый световой поток не выходит за пределы требуемого диапазона. Теперь рассчитаем мощность осветительной установки:

$$P = 16 \cdot 40 = 640 \text{ Вт}$$

### 4.3 Экологическая безопасность

В компьютерах огромное количество компонентов, которые содержат токсичные вещества и представляют угрозу, как для человека, так и для окружающей среды.

К таким веществам относятся:

- свинец (накапливается в организме, поражая почки, нервную систему);
- ртуть (поражает мозг и нервную систему);

- никель и цинк (могут вызывать дерматит);
- щелочи (прожигают слизистые оболочки и кожу);

Поэтому компьютер требует специальных комплексных методов утилизации. В этот комплекс мероприятий входят:

- отделение металлических частей от неметаллических;
- металлические части переплавляются для последующего производства;
- неметаллические части компьютера подвергаются специально переработке[6];

Исходя из сказанного выше перед планированием покупки компьютера необходимо:

- Побеспокойтесь заранее о том, каким образом будет утилизирована имеющаяся техника, перед покупкой новой.
- Узнать насколько новая техника соответствует современным эко-стандартам и примут ее на утилизацию после окончания срока службы.

Утилизировать оргтехнику, а не просто выбрасывать на «свалку» необходимо по следующим причинам:

Во-первых, в любой компьютерной и организационной технике содержится некоторое количество драгоценных металлов. Российским законодательством предусмотрен пункт, согласно которому все организации обязаны вести учет и движение драгоценных металлов, в том числе тех, которые входят в состав основных средств. За несоблюдение правил учета, организация может быть оштрафована на сумму от 20000 до 30000 руб. (согласно ст. 19.14. КоАП РФ [22]);

Во-вторых, предприятие также может быть оштрафовано за несанкционированный вывоз техники или оборудования на «свалку»;

Стадия утилизации, утилизируя технику мы заботимся об экологии: количество не перерабатываемых отходов минимизируется, а такие отходы, как пластик, пластмассы, лом черных и цветных металлов, используются во

вторичном производстве. Электронные платы, в которых содержатся драгметаллы, после переработки отправляются на аффинажный завод, после чего чистые металлы сдаются в Госфонд, а не оседают на свалках.

Таким образом утилизацию компьютера можно провести следующим образом:

1.Использовать услуги профессиональной компании по рециклингу, которая может приехать и забрать все приборы, которые планируется сдать в переработку.

2.Можно обратиться в местный муниципалитет по вопросу переработки электроники.

#### **4.4 Безопасность в чрезвычайных ситуациях**

##### **4.4.1 Природная чрезвычайная ситуация**

Природная чрезвычайная ситуация – обстановка на определенной территории или акватории, сложившейся в результате возникновения источника природной чрезвычайной ситуации, который может повлечь или повлек за собой человеческие жертвы, ущерб здоровью людей и (или) окружающей природной среде, значительные материальные потери и нарушение условий жизнедеятельности людей.

Производство находится в городе Томске с континентально-циклоническим климатом. Природные явления (землетрясения, наводнения, засухи, ураганы и т. д.), в данном городе отсутствуют.

Возможными ЧС на объекте в данном случае, могут быть сильные морозы и диверсия.

Для Сибири в зимнее время года характерны морозы. Достижение критически низких температур приведет к авариям систем теплоснабжения и жизнеобеспечения, приостановке работы, обморожениям и даже жертвам среди населения. В случае переморозки труб должны быть предусмотрены запасные обогреватели. Их количества и мощности должно хватать для того, чтобы работа на производстве не прекратилась.

##### **4.4.2 Анализ пожарной безопасности**

По взрывопожарной и пожарной опасности помещения подразделяются на категории А, Б, В1-В4, Г и Д, а здания на категории А, Б, В, Г и Д.

Согласно НПБ 105-03 [19] лаборатория относится к категории В–горючие и трудно горючие жидкости, твердые горючие и трудно горючие вещества и материалы, вещества и материалы, способные при взаимодействии с водой, кислородом воздуха или друг с другом только гореть, при условии, что помещения, в которых находится, не относятся к категории наиболее опасных А или Б.

По степени огнестойкости данное помещение относится к 1-й степени огнестойкости по СНиП 2.01.02-85 [20] (выполнено из кирпича, которое относится к трудностгораемым материалам).

Возникновение пожара при работе с электронной аппаратурой может быть по причинам как электрического, так и неэлектрического характера.

Причины возникновения пожара неэлектрического характера:

а) халатное неосторожное обращение с огнем (курение, оставленные без присмотра нагревательные приборы, использование открытого огня);

Причины возникновения пожара электрического характера: короткое замыкание, перегрузки по току, искрение и электрические дуги, статическое электричество и т. п.

Для локализации или ликвидации загорания на начальной стадии используются первичные средства пожаротушения. Первичные средства пожаротушения обычно применяют до прибытия пожарной команды.

Огнетушители водо-пенные (ОХВП-10) используют для тушения очагов пожара без наличия электроэнергии. Углекислотные (ОУ-2) и порошковые огнетушители предназначены для тушения электроустановок, находящихся под напряжением до 1000В. Для тушения токоведущих частей и электроустановок применяется переносной порошковый огнетушитель, например ОП-5.

В общественных зданиях и сооружениях на каждом этаже должно размещаться не менее двух переносных огнетушителей. Огнетушители следует располагать на видных местах вблизи от выходов из помещений на высоте не более 1,35 м. Размещение первичных средств пожаротушения в коридорах, переходах не должно препятствовать безопасной эвакуации людей.

Для предупреждения пожара и взрыва необходимо предусмотреть:

1. специальные изолированные помещения для хранения и разлива легковоспламеняющихся жидкостей (ЛВЖ), оборудованные приточно-вытяжной вентиляцией во взрывобезопасном исполнении - соответствии с ГОСТ 12.4.021-75 [21];

2. специальные помещения (для хранения в таре пылеобразной канифоли), изолированные от нагревательных приборов и нагретых частей оборудования;

3. автоматические сигнализаторы для сигнализации о присутствии в воздухе помещений дозрывных концентраций горючих паров растворителей и их смесей.

Лаборатория полностью соответствует требованиям пожарной безопасности, а именно, наличие охранно-пожарной сигнализации, плана эвакуации, изображенного на рисунке 25, порошковых огнетушителей с поверенным клеймом, табличек с указанием направления к запасному (эвакуационному) выходу.

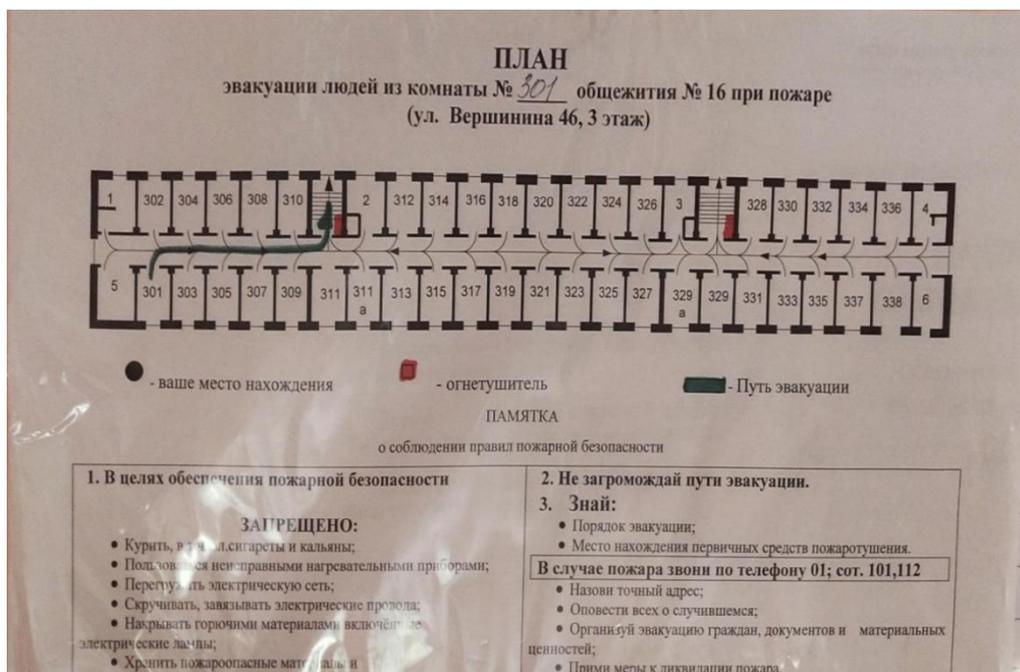


Рисунок 25 – План эвакуации

### Выводы по разделу

В ходе исследования рабочего места было выявлено соответствие следующий факторов: электромагнитное излучение, Отклонение показателей микроклимата, Превышение уровня шума, Недостаточная освещенность рабочей зоны, Пожарная опасность, безопасность в чрезвычайных ситуациях.

В работе рассмотрено воздействие вредных и опасных факторов на человека и окружающую среду в процессе обработки алгоритма. И как избежать и предотвратить эти эффекты.

## **Заключение**

Результатом выполнения выпускной квалификационной работы является, разработанная в соответствии с техническим заданием, Алгоритм автоматического определения типа файловой системы по сигнатуре. В процессе выполнения работы были выполнены ряд задач, таких как:

- узнавание структуру файловых систем FAT и NTFS;
- узнавание структуру главной загрузочной области(MBR) жесткого диска;
- В Delphi7 разработан алгоритм для автоматического определения типа файловой системы;
- На языке C разработан алгоритм автоматического определения типа файловой системы по сигнатуре.

Наконец, было выполнено обоснование проекта с экономической точки зрения, а также были рассмотрены вопросы экологической безопасности проекта и условий труда работающего персонала.

## Список используемых источников

1. Файловая система NTFS – URL: <https://www.ixbt.com/storage/ntfs.html> (дата обращения: 20.05.2020).
2. Файловая система FAT32 – URL: <https://ru.wikipedia.org/wiki/FAT32> (дата обращения: 20.05.2020).
3. Файловая система exFAT – URL: <https://ru.wikipedia.org/wiki/ExFAT> (дата обращения: 20.05.2020).
4. Архитектура файловой системы FAT – URL: <http://samag.ru/archive/article/245> (дата обращения: 20.05.2020).
5. Главная загрузочная запись (MBR) – URL: [https://ru.wikipedia.org/wiki/%D0%93%D0%BB%D0%B0%D0%B2%D0%BD%D0%B0%D1%8F\\_%D0%B7%D0%B0%D0%B3%D1%80%D1%83%D0%B7%D0%BE%D1%87%D0%BD%D0%B0%D1%8F\\_%D0%B7%D0%B0%D0%BF%D0%B8%D1%81%D1%8C](https://ru.wikipedia.org/wiki/%D0%93%D0%BB%D0%B0%D0%B2%D0%BD%D0%B0%D1%8F_%D0%B7%D0%B0%D0%B3%D1%80%D1%83%D0%B7%D0%BE%D1%87%D0%BD%D0%B0%D1%8F_%D0%B7%D0%B0%D0%BF%D0%B8%D1%81%D1%8C) (дата обращения: 20.05.2020).
6. Список сигнатур файлов – URL: [https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B8%D1%81%D0%BE%D0%BA\\_%D1%81%D0%B8%D0%B3%D0%BD%D0%B0%D1%82%D1%83%D1%80\\_%D1%84%D0%B0%D0%B9%D0%BB%D0%BE%D0%B2](https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B8%D1%81%D0%BE%D0%BA_%D1%81%D0%B8%D0%B3%D0%BD%D0%B0%D1%82%D1%83%D1%80_%D1%84%D0%B0%D0%B9%D0%BB%D0%BE%D0%B2) (дата обращения: 20.05.2020).
7. Технологии восстановления данных: поиск по сигнатурам – URL: <https://recovery-software.ru/blog/data-recovery-technology.html> (дата обращения: 20.05.2020).
8. Максименко, Георгий Тарасович. Техника безопасности при применении пожароопасных, взрывоопасных и токсичных материалов / Г. Т. Максименко, В. М. Покровский. — 3-е изд., перераб. и доп. — Киев: Будівельник, 1987. — 150 с.: ил.: 22 см. — Библиогр.: с. 148 (27 назв.). <http://catalog.lib.tpu.ru/catalogue/simple/document/RU%5CTPU%5Cbook%5C294981>
9. Трудовой кодекс Российской Федерации от 30.12.2001 N 197-ФЗ (ред. от 31.05.2020)

10. ГОСТ 12.2.032-78 ССБТ. Рабочее место при выполнении работ сидя. Общие эргономические требования.(ред. от 31.05.2020)
11. ГОСТ 12.0.003-2015. Система стандартов безопасности труда (ССБТ). Опасные и вредные производственные факторы. Классификация.(ред. от 31.05.2020)
12. СанПиН 2.2.2/2.4.1340-03. Гигиенические требования к персональным электронно-вычислительным машинам и организации работы.(ред. от 31.05.2020)
13. СанПиН 2.2.4.548-96. Гигиенические требования к микроклимату производственных помещений.(ред. от 31.05.2020)
14. СН 2.2.4/2.1.8.562-96. Шум на рабочих местах, в помещениях жилых, общественных зданий и на территории жилой застройки.(ред. от 31.05.2020)
15. СанПиН 2.2.1/2.1.1.1278-03. Гигиенические требования к естественному, искусственному и совмещенному освещению жилых и общественных зданий.(ред. от 31.05.2020)
16. СанПиН 2.1.7.1322-03. Гигиенические требования к размещению и обезвреживанию отходов производства и потребления.(ред. от 31.05.2020)
17. ГОСТ Р 22.3.03-94. Безопасность в чрезвычайных ситуациях. Защита населения. Основные положения (принят в качестве межгосударственного стандарта ГОСТ 22.3.03-97).(ред. от 31.05.2020)
18. СНиП 23-05-95\* Естественное и искусственное освещение (с Изменением N 1). (ред. от 31.05.2020)
19. НПБ 105-03 Определение категорий помещений, зданий и наружных установок по взрывопожарной и пожарной опасности. (ред. от 31.05.2020)
20. СНиП 2.01.02-85\* Противопожарные нормы. (ред. от 31.05.2020)
21. ГОСТ 12.4.021-75 Система стандартов безопасности труда (ССБТ). Системы вентиляционные. Общие требования (с Изменением N 1). (ред. от 31.05.2020)

22. Ст. 19.14 КоАП РФ Утратила силу. - Федеральный закон от 27.12.2019 N 501-ФЗ. (ред. от 31.05.2020)