СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 5](#_Toc89639123)

[1 ОБЗОР ИСТОЧНИКОВ 6](#_Toc89639124)

[2 РАЗРАБОТКА СТРУКТУРНОЙ СХЕМЫ 7](#_Toc89639125)

[3 РАЗРАБОТКА ФУНКЦИОНАЛЬНОЙ СХЕМЫ 8](#_Toc89639126)

[4 РАЗРАБОТКА ПРИНЦИПИАЛЬНОЙ СХЕМЫ 9](#_Toc89639127)

[5 ТЕХНИКО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ 10](#_Toc89639128)

[6 РАЗРАБОТКА ПЕЧАТНОЙ ПЛАТЫ 11](#_Toc89639129)

[7 ОПИСАНИЕ РАБОТЫ УСТРОЙСТВА 12](#_Toc89639130)

[ПРИЛОЖЕНИЕ A 15](#_Toc89639131)

[ПРИЛОЖЕНИЕ Б 16](#_Toc89639132)

[ПРИЛОЖЕНИЕ В 17](#_Toc89639133)

[ПРИЛОЖЕНИЕ Г 18](#_Toc89639134)

[ПРИЛОЖЕНИЕ Д 19](#_Toc89639135)

[ПРИЛОЖЕНИЕ Е 20](#_Toc89639136)

# ВВЕДЕНИЕ

Файловая система является неотъемлемой частью любого современного устройства хранения данных: начиная от обычных жестких дисков для бытовых компьютеров или flash-карт для смартфонов и заканчивая стримерами для создания резервных копий промышленных серверов. Определяя формат содержимого и способ физического хранения информации, файловая система организует данные в файлы и каталоги. При этом она связывает носитель информации с одной стороны и API для доступа к файлам – с другой.

Без файловой системы было бы невозможно использовать современный

# 1 ОБЗОР ИСТОЧНИКОВ

Современные операционные системы традиционно предоставляют механизм API для уведомлений от файловых систем о событиях, происходящих в файловой системе. Примером могут служить уведомления об открытии файла или каталога, его изменении, удалении, перемещении. Со временем в ядре операционной системы Linux появилось три различных API уведомления файловой системы: dnotify, inotify и fanotify.

Эти уведомления могут находить применение в самых разных областях работы с ОС, при этом как включая запросы конечного пользователя, так и так и обслуживая запросы, происходящие “под капотом” у прикладного ПО.

Отслеживание таких событий намного эффективнее, чем ручной вызов системных методов по типу readdir()*,* stat() и тд. К тому же, API уведомлений позволяет обнаружить такие события, которые невозможно или очень затратно отслеживать с помощью системных вызовов выше.

# 2 РАЗРАБОТКА СТРУКТУРНОЙ СХЕМЫ

Данную программу можно разбить на следующие главные блоки, реализующие ее полный функционал:

# 3 РАЗРАБОТКА ФУНКЦИОНАЛЬНОЙ СХЕМЫ

Согласно выделенным в разделе структурного проектирования блоками, дерево проекта разбито на следующие файлы:

# 4 РАЗРАБОТКА ПРИНЦИПИАЛЬНОЙ СХЕМЫ

Разработку программных модулей стоит начать со структур данных, которые будут использоваться в процессе работы приложения.

# 5 ТЕХНИКО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ

На этапе тестирования была проверена работоспособность программы. При запуске пользователь видит следующее окно (рисунок 5.1):

# 6 РАЗРАБОТКА ПЕЧАТНОЙ ПЛАТЫ

На этапе тестирования была проверена работоспособность dddfhdfhdfhdfhdfhdfhdfhdfhdfhdfhdfhdfh

# 7 ОПИСАНИЕ РАБОТЫ УСТРОЙСТВА

На этапе тестирования была проверена работоспособность dddfhdfhdfhdfhdfhdfhdfhdfhdfhdfhdfhdfh

**ЗАКЛЮЧЕНИЕ**

В ходе выполнения курсового проекта было создано приложение под Свободное место

**СПИСОК ЛИТЕРАТУРЫ**

[1] Прата, С. Язык программирования С++. Лекции и упражнения /П. Стивен; 6-е издание. Пер. с англ. ­­­­– М. ООО “И.Д.Вильямс”, 2012. –1248 c.

[2] Реализация epoll [Электронный ресурс]. – Режим доступа:  [https://habr.com/ru/company/ruvds/blog/523946/](%20https://habr.com/ru/company/ruvds/blog/523946/).

[3] Википедия – Свободная энциклопедия [Электронный ресурс]. – Режим доступа: https://en.wikipedia.org/wiki/Dnotify .

[4] Ddnotify(1) - Linux man page [Электронный ресурс]. – Режим доступа: [https://linux.die.net/man/1/dnotify](https://linux.die.net/man/1/dnotify%20)  .

[5] Filesystem notification, part 1: An overview of dnotify and inotify [Электронный ресурс]. – Режим доступа: [https://lwn.net/Articles/604686/](https://lwn.net/Articles/604686/%20)  .

[6] inotify(7) — Linux manual page [Электронный ресурс]. – Режим доступа: [https://man7.org/linux/man-pages/man7/inotify.7.html](https://man7.org/linux/man-pages/man7/inotify.7.html%20)  .

[7] fanotify(7) — Linux manual page [Электронный ресурс]. – Режим доступа: [https://man7.org/linux/man-pages/man7/fanotify.7.html](https://man7.org/linux/man-pages/man7/fanotify.7.html%20)  .

[8] Articles by Michael Kerrisk [Электронный ресурс]. – Режим доступа: [https://man7.org/articles/index.html](https://man7.org/articles/index.html%20)

[9] ftw(3): file tree walk - Linux man page [Электронный ресурс]. – Режим доступа: [https://linux.die.net/man/3/ftw](https://linux.die.net/man/3/ftw%20)

[10] Epoll\_ctl: интерфейс управления файловым дескриптором epoll [Электронный ресурс]. – Режим доступа: [http://ru.manpages.org/epoll\_ctl/2](http://ru.manpages.org/epoll_ctl/2%20)

[11] epoll - I/O event notification facility [Электронный ресурс]. – Режим доступа: [https://www.opennet.ru/man.shtml?topic=epoll&category=7](https://www.opennet.ru/man.shtml?topic=epoll&category=7&russian=2)

[12] Знакомство с межпроцессным взаимодействием на Linux [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/post/122108/>

[13] Средства межпроцессного взаимодействия на OS Linux [Электронный ресурс]. – Режим доступа: <https://stepik.org/lesson/46413/step/1?unit=24448/>

# ПРИЛОЖЕНИЕ A

(обязательное)

**Схема структурная**

# ПРИЛОЖЕНИЕ Б

(обязательное)

**Схема функциональная**

# ПРИЛОЖЕНИЕ В

(обязательное)

**Схема принципиальная**

# ПРИЛОЖЕНИЕ Г

(обязательное)

**Схема печатной платы**

# ПРИЛОЖЕНИЕ Д

**Листинг кода**

#include <sstream>

#include <unistd.h>

#include <stdio.h>

#include <stdlib.h>

#include"inotifiermainwindow.h"

#include "interProcessingCommunications.h"

#include "Inotifier/InotifyServiceInterface.h"

#include <QApplication>

int main(int argc, char \*argv[]){

QApplication app (argc, argv);

# ПРИЛОЖЕНИЕ Е

(обязательное)

**Ведомость документов**