

ГУАП


Классификация № 14

Отчет

Защиты с оценкой

Преподаватель

Старший преподаватель
должность, уч. степень, звание


подпись, дата

Т. А. Прохорова
инициалы, фамилия

Отчет о лабораторной работе № 2
Разработка файловой системы
по курсу: Операционные системы

Работу выполнил

Студент гр. № 1842

Иванов 29.05.21
подпись, дата

Иванов А.В.
инициалы, фамилия

Санкт - Петербург 2021

Оглавление

1.	Постановка задачи.....	3
2.	Схема файловой системы.....	4
3.	Структура дискового раздела	5
4.	Суперблок	6
5.	Группы блоков.....	7
6.	Каталоги	8
7.	Система адресации данных	9
8.	Список возможных команд	10

1. Постановка задачи

Используя имеющиеся знания, спроектировать собственную файловую систему для дальнейшей её реализации на языке программирования С.

2. Схема файловой системы

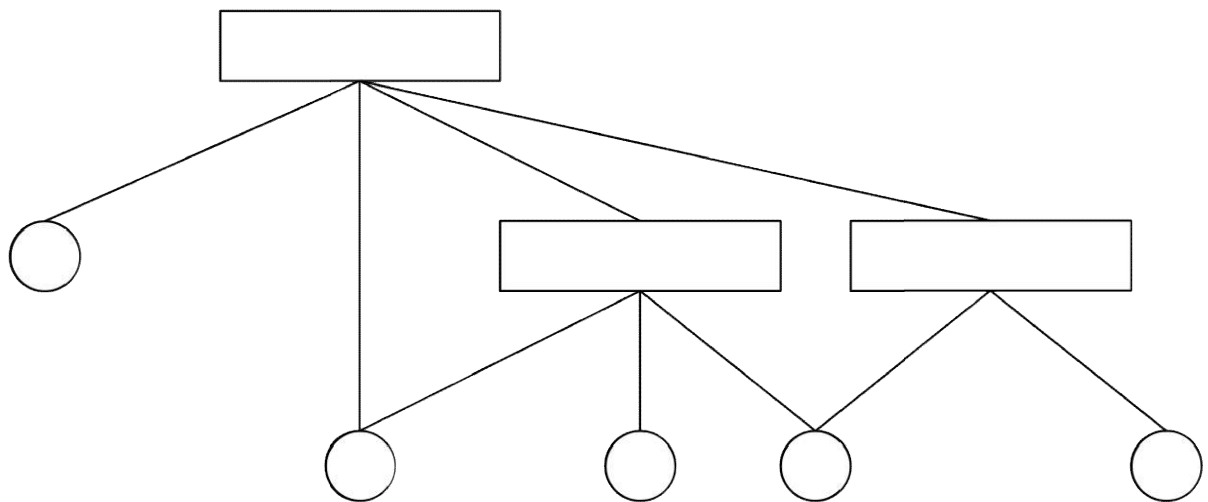


Рис. 1 – файловоедерево

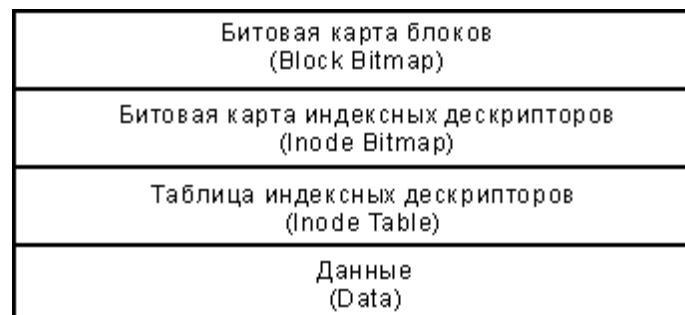


Рис.2 – обобщеннаяструктурнаясхема ФС

3. Структура дискового раздела

Система приближена к ext2.

Выделим следующие составляющие для нашей файловой системы:

- Блоки и группы блоков;
- inode
- суперблок

Всё пространство раздела диска разбивается на блоки фиксированного размера, кратные размеру сектора: 2048 байт. Размер блока указывается при создании файловой системы на разделе диска. Меньший размер блока позволяет сэкономить место на жёстком диске, но также ограничивает максимальный размер файловой системы. Все блоки имеют порядковые номера. С целью уменьшения фрагментации и количества перемещений головок жёсткого диска при чтении больших массивов данных блоки объединяются в группы блоков. (в нашем случае группа блоков – одна)

Базовым понятием файловой системы является индексный дескриптор, или inode (англ. informationnode). Это специальная структура, которая содержит информацию об атрибутах и физическом расположении файла. Индексные дескрипторы объединены в таблицу, которая содержится в начале каждой группы блоков.

4. Суперблок

Суперблок находится в 1024 байтах от начала раздела. В следующем блоке после суперблока располагается глобальная дескрипторная таблица — описание групп блоков, представляющее собой массив, содержащий общую информацию обо всех группах блоков раздела.

От целостности суперблока напрямую зависит работоспособность файловой системы. Операционная система создаёт несколько резервных копий суперблока на случай повреждения раздела. С помощью флага состояния операционная система определяет текущее состояние файловой системы. Если файловая система монтируется на чтение, то флаг состояния будет указывать, что файловая система целостна (состояние «clean»). Если файловая система монтируется на чтение и запись, то в флаг состояния заносится информация о том, что файловая система используется (состояние «notclean»), а после размонтирования файловой системы флаг состояния снова должен указывать на целостность файловой системы. Флаг состояния помогает определять возможные повреждения файловой системы. Например, если питание компьютера было неожиданно отключено, то флаг состояния будет указывать на некорректное завершение работы с файловой системой. При следующей загрузке компьютера операционная система должна будет проверить файловую систему на ошибки, если флаг состояния не указывает на целостность файловой системы.

В нашем случае суперблок отсутствует, но его функциональность будет реализована непосредственно в момент компиляции. Вся информация будет храниться в отдельном файле заголовке.

5. Группы блоков

Все блоки раздела ext2 объединяются в группы блоков. Для каждой группы создаётся отдельная запись в глобальной дескрипторной таблице, в которой хранятся основные параметры:

- номер блока в битовой карте блоков,
- номер блока в битовой карте inode,
- номер блока в таблице inode,
- число свободных блоков в группе,
- число индексных дескрипторов, содержащих каталоги.

Битовая карта блоков — это структура, каждый бит которой показывает, отведён ли соответствующий ему блок какому-либо файлу. Если бит равен 1, то блок занят. Аналогичную функцию выполняет битовая карта индексных дескрипторов, которая показывает, какие именно индексные дескрипторы заняты, а какие нет. Ядро Linux, используя число индексных дескрипторов, содержащих каталоги, пытается равномерно распределить inode каталогов по группам, а inode файлов старается по возможности переместить в группу с родительским каталогом. Все оставшееся место, обозначенное в таблице как данные, отводится для хранения файлов.

В нашем примере используется только 1 группа блоков.

6. Каталоги

Каталоги могут содержать внутри себя другие каталоги или файлы. Физически каталог представляет из себя специальный файл, содержащий записи произвольной длины. Каждая запись хранит в себе следующие данные:

- номер индексного дескриптора файла,
- размер записи,
- длину имени файла,
- имя файла.

Подобная организация каталога позволяет хранить в нём длинные имена файлов без потери места на диске.

Когда операционная система пытается найти расположение файла (или каталога) на диске, она по очереди загружает в память содержимое каждого каталога, указанного в пути к файлу (или каталогу), чтобы по имени найти индексный дескриптор следующего каталога, указанного в пути. Обход каталогов продолжается, пока необходимый файл или каталог не будет найден.

В нашем случае запись будет состоять из:

- номера индексного дескриптора файла
- имени файла

7. Система адресации данных

Система адресации данных — это одна из самых важных составляющих файловой системы. Именно она позволяет находить нужный файл среди множества как пустых, так и занятых блоков на диске.

Файловая система ext2 использует следующую схему адресации блоков файла. Для хранения адреса файла выделено 15 полей, каждое из которых состоит из 4 байт. Если файл умещается в 12 блоков, то номера соответствующих кластеров непосредственно перечисляются в первых двенадцати полях адреса. Если размер файла превышает 12 блоков, то следующее поле содержит адрес кластера, в котором могут быть расположены номера следующих блоков файла. Таким образом, 13-е поле используется для косвенной адресации.

При максимальном размере блока в 4096 байт кластер, соответствующий 13-му полю, может содержать до 1024 номеров следующих блоков файла. Если размер файла превышает 12+1024 блоков, то используется 14-е поле, в котором находится адрес кластера, содержащего 1024 номеров кластеров, каждый из которых ссылается на 1024 блока файла. Здесь применяется уже двойная косвенная адресация. И наконец, если файл включает более 12+1024+1048576 блоков, то используется последнее 15-е поле для тройной косвенной адресации.

Данная система адресации позволяет при максимальном размере блока в 4096 байт иметь файлы, размер которых превышает 2 ТБ.

