

# Лекция 3

## Файловая система

# Определение

Файловая система (англ. file system) — порядок, определяющий способ организации, хранения и именования данных на носителях информации в компьютерах, а также в другом электронном оборудовании: цифровых фотоаппаратах, мобильных телефонах и т. п. Файловая система определяет формат содержимого и способ физического хранения информации, которую принято группировать в виде файлов. Конкретная файловая система определяет размер имен файлов и (каталогов), максимальный возможный размер файла и раздела, набор атрибутов файла. Некоторые файловые системы предоставляют сервисные возможности, например разграничение доступа или шифрование файлов.

# Типы файлов в Linux

- : обычный файл

d : директория(каталог)

c : файл символьного устройства

b : файл блочного устройства

s : файл локального сокета

p : именованный канал

l : символическая ссылка

**Обычный файл** — к этим файлам относятся текстовые файлы, бинарные данные, исполняемые программы.

**Каталог** — это файл, содержащий имена находящихся в нем файлов, а также указатели на дополнительную информацию — метаданные, позволяющие осуществлять операции над этими файлами. С помощью каталога формируется логическое дерево ф.с. Он определяет положение файла в дереве ф.с, поскольку сам файл не содержит информации о своем местонахождении.

В Linux различают символьные(character) и блочные(block) файлы устройств.

**Символьные файлы** — используются для небуферизированного обмена данными с устройством посимвольно.

**Блочные файлы** — позволяют производить обмен данными с устройствами в виде пакетов фиксированной длины - блоков

**Именованный канал**(named pipe) — это файл, используемый для связи между процессами.

В Linux один файл может иметь несколько имен это достигается тем что метаданные(индексный дескриптор) не связан с названием файла. И мы имеем возможность задавать жесткую ссылку — второе название файла.

```
$> ln name.file new_name.file
```

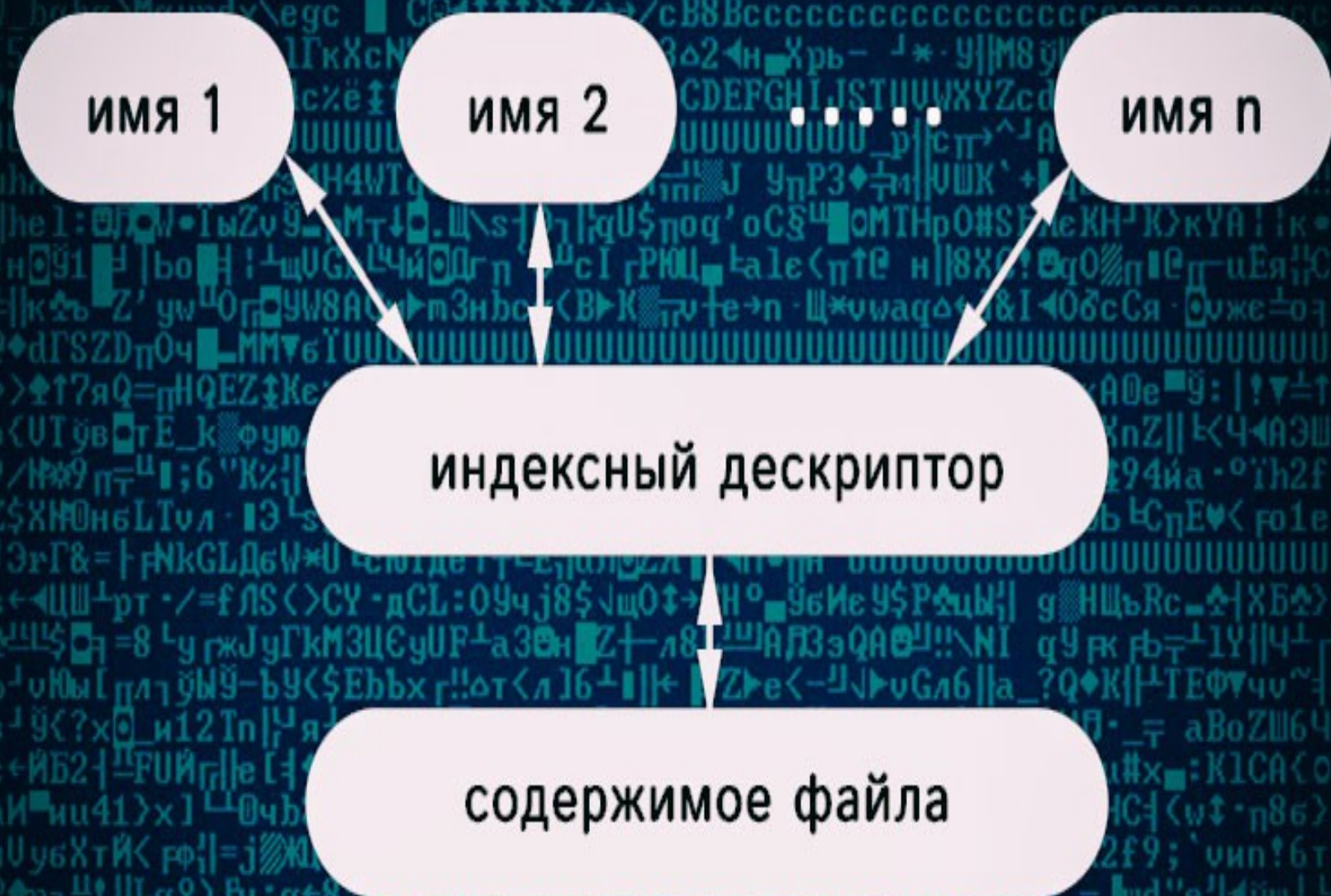
При таком подходе индексные дескрипторы будут одинаковые. При удалении и перемещении одного из файлов ничего не меняется. Жесткая связь не принадлежит к особому типу файла и является естественной формой связи.

**Символическая ссылка** — специальный файл который содержит ссылку на название другого файла.

```
$> ln -s name.file link_name.file
```

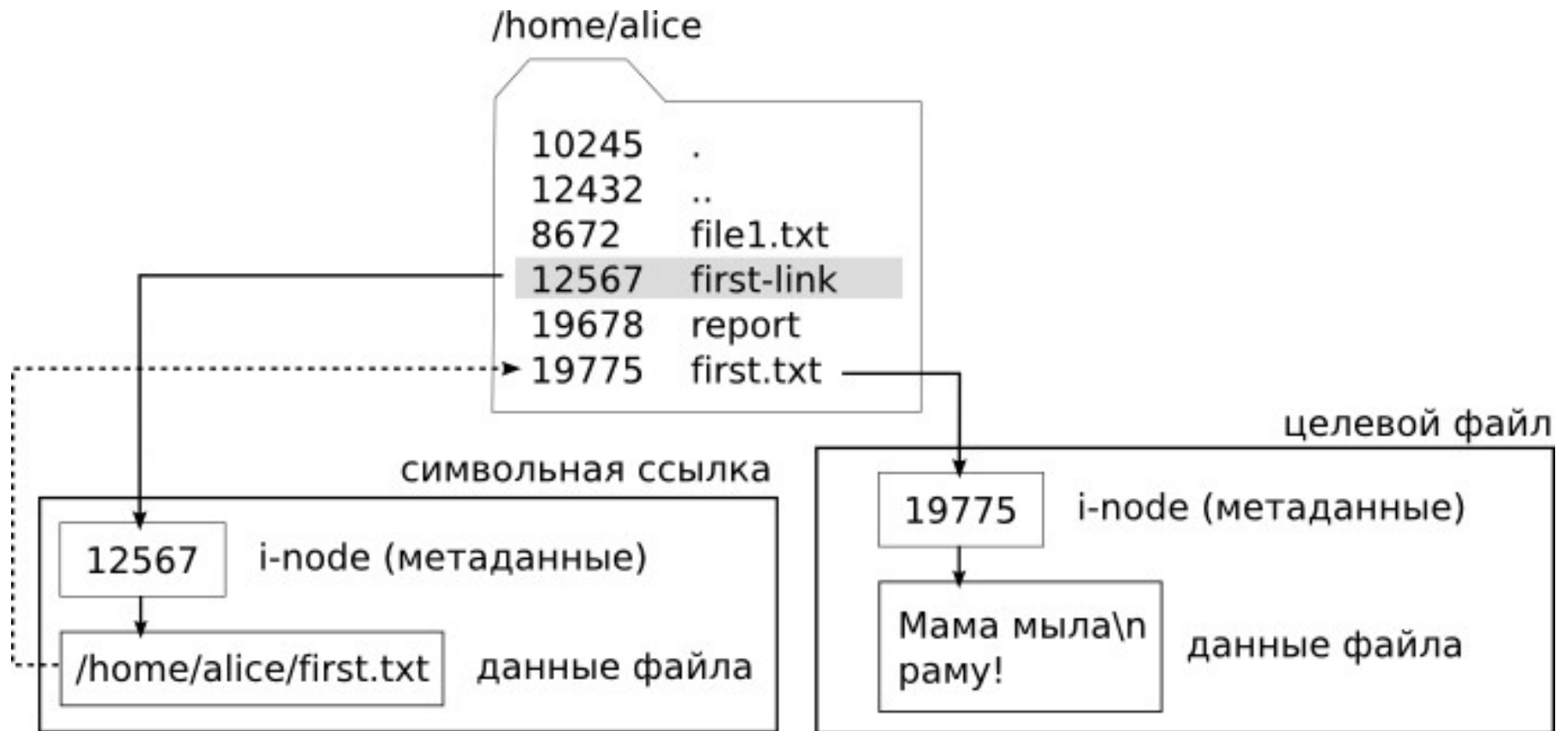
При удалении исходного файла, ссылка становится битой.



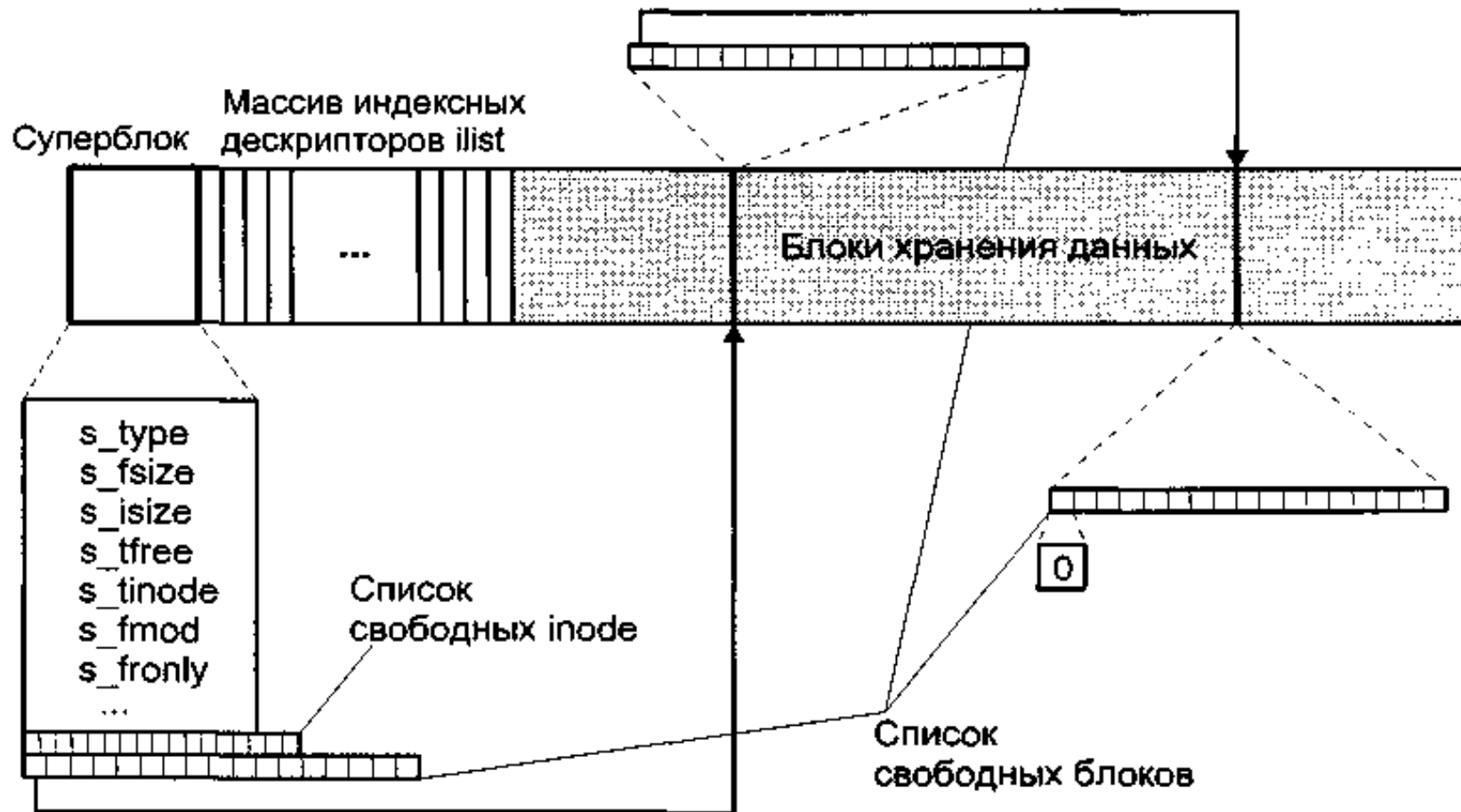


**Максимальное количество  
индексных дескрипторов**

# Символьная ссылка



# Базовая файловая система System V





# Базовая файловая система System V

Файловая система s5fs занимает раздел диска и состоит из 3-х основных компонентов

- Суперблок — содержит информацию необходимую для монтирования и управления работой ф.с. В каждой ф.с существует только один суперблок, который располагается в начале раздела. Суперблок считывается в память при монтировании файловой системы и находится там до ее отключения (размонтирования)
- Индексные дескрипторы — содержат информацию о файле, необходимую для обработки данных т. е. метаданные файла. Каждый файл ассоциирован с одним inode
- Блоки хранения данных — содержат в себе данные файлов и каталогов.

# Суперблок

содержит следующую информацию

s\_type — тип файловой системы

s\_fsize — размер в лог. блоках

s\_isize — размер массива индексных дискр.

s\_tfree - число свободных блоков для размещ.

s\_tinode — свободных inode для размещения.

Размер логических блоков (512, 1024, 2048)

Список адресов свободных блоков

# Как суперблок выглядит в коде

current->namespace->list->mnt\_sb

См. рисунок 3

```
struct super_block {  
    struct list_head  
    unsigned long  
    struct file_system_type *s_type;  
    struct super_operations *s_op;  
    struct semaphore  
    int  
    struct list_head  
    struct block_device  
    ...  
};
```

s\_list;

s\_blocksize;

s\_type;

s\_op;

s\_lock;

s\_need\_sync\_fs;

s\_dirty;

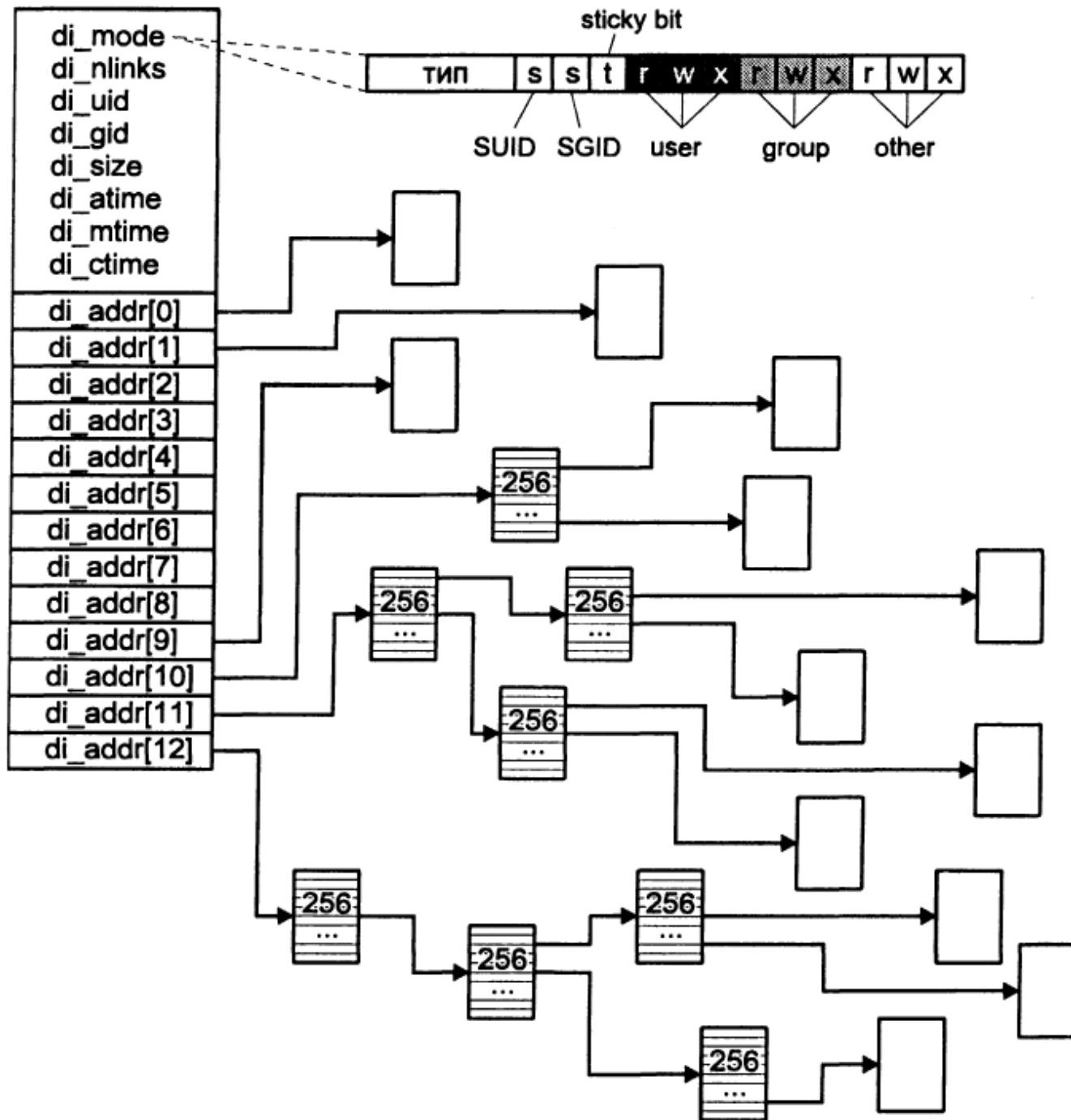
\*s\_bdev;

Двусвязный список всех  
смонтированных файловых систем

См. рисунок 2

```
struct super_operations {  
    struct inode *(*alloc_inode)(struct super_block *sb);  
    void (*destroy_inode)(struct inode *);  
    void (*read_inode)(struct inode *);  
    void (*write_inode)(struct inode *, int);  
    int (*sync_fs)(struct super_block *sb, int wait);  
    ...  
};
```

# Индексный дескриптор



# Основные поля inode

di\_mode — тип файла, доп. атрибуты выполнения и права доступа(IFDIR,IFBLK)

di-nlinks — число ссылок на файл

di\_uid, di\_gid — идентификаторы владельца-пользователя и владельца группы.

di\_size — размер файла в байтах

di\_atime — время последнего доступа к файлу

di\_mtime — время последней модификации

di\_addr[13] — Массив адресов дисковых блоков хранения данных(фиксированный 13 элементов)



# Недостатки файловой системы

- Слабым местом является суперблок так как он хранится в единственном варианте и при его повреждении файловая система не может использоваться.
- Фиксированный список inode что ограничивает нас в максимальном кол-ве файлов в системе.
- Блок данных (512, 1024 byte) закреплен за одним файлом что дает не экономичное использование ресурса

# Монтирование файловой системы

Копируем образ диска

```
$> dd if=/dev/cdrom of=cdrom.iso bs=1M
```

Монтируем содержимое файла в папку

```
$> mount -o loop cdrom.iso /mnt/cdrom
```

# Современные файловые системы

На текущий момент в Linux используются продвинутое файловые системы ext4, reiser4. Которые лишены многих перечисленных недостатков.