

UNIX

Лекция 2

Базовая файловая система UNIX System V

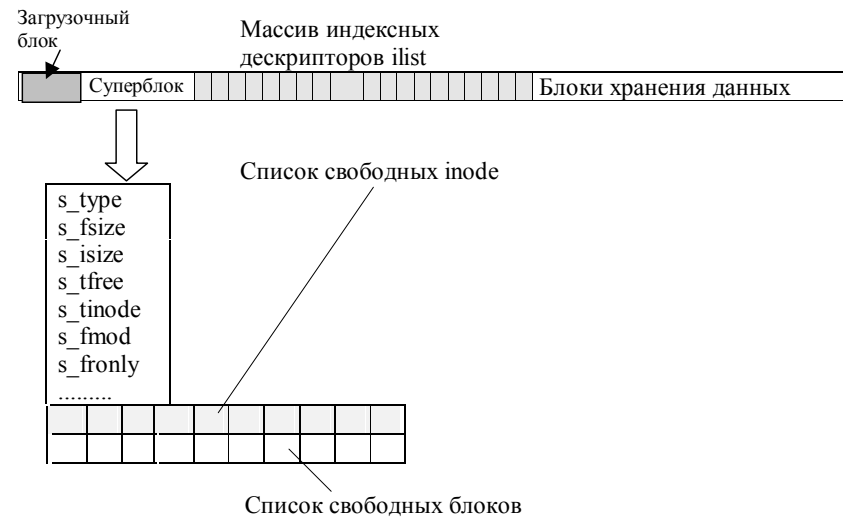
Загрузочный блок. Располагается в самом начале дискового пространства и содержит программу начальной загрузки. Каждая файловая система имеет свой блок загрузки.

Суперблок. Информация, необходимая для монтирования и управления ФС (тип файловой системы, флаг монтирования и т.п.):

- s_type - тип файловой системы
- s_fsize - размер файловой системы в блоках
- s_ isize - размер массива индексных дескрипторов
- s_tfree - число свободных блоков, доступных для размещения данных
- s_tinode - число свободных элементов inode
- s_fmod - флаг модификации
- s_fronly - флаг монтирования

- Поле, определяющее размер каждого блока хранения данных
- номера свободных элементов списка ilist.
- номера свободных блоков файловой системы
- **ilist** – список элементов inode, описывающих файл (метаданные файла).

Основная единица обработки – блок.

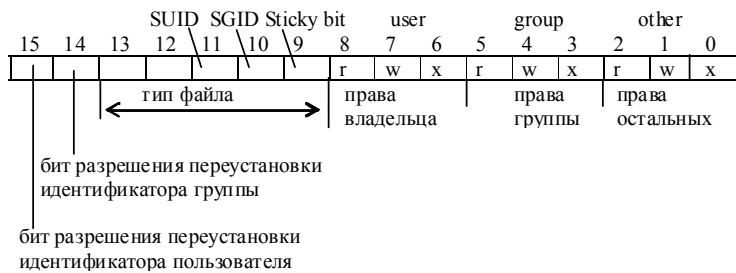


Индексные дескрипторы

Описатель файла или дескриптор inode:

- di_mode атрибуты файла – режимы доступа и тип файла;
- di_nlink счетчик числа ссылок на файл;
- di_uid идентификатор его владельца-пользователя;
- di_gid идентификатор группы;
- di_size размер файла в байтах;
- di_atime дата последнего доступа;
- di_mtime дата последней модификации БХД;
- di_ctime дата последнего изменения метаданных (все, кроме времен);
- di_addr[13] указатели на блоки диска, в которых хранится файл (адреса блоков, в которых хранится информация).

Поле di_mode состоит из 16 разрядов:

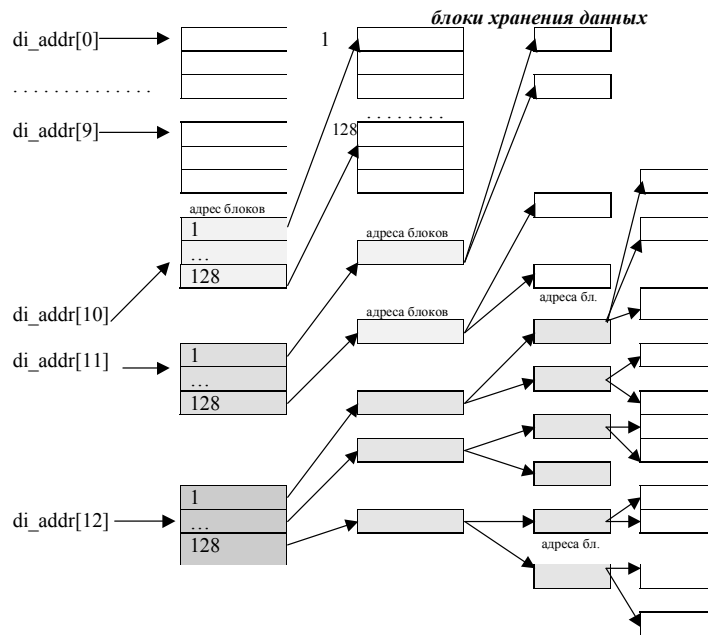


```

struct dinode
{ unsigned
short di_mode; /* режим доступа и тип файла */
short di_nlink; /* счетчик числа ссылок на файл */
short di_uid; /* идентификатор его владельца */
short di_gid; /* идентификатор группы */
off_t di_size; /* счетчик числа байт в файле */
char di_addr[40]; /* указатели на блоки диска,
в которых хранится сам файл */
time_t di_atime; /* дата последнего доступа */
time_t di_mtime; /* дата посл. модификации БХД */
time_t di_ctime; /* дата посл. измен. метаданных */
}
    
```

Поле di_addr

- di_addr[0]
 - di_addr[1]
 -
 - di_addr[8]
 - di_addr[9]
- } прямая (непосредственная) адресация
- di_addr[10] - косвенная адресация - указатель на блок, содержащий 128 адресов (указателей) на блоки хранения данных (БХД).
 - di_addr[11] - двойная косвенная адресация - содержит адрес блока, хранящего 128 указателей на блоки косвенной адресации.
 - di_addr[12] - тройная косвенная адресация - содержит адрес блока, хранящего 128 указателей на блоки двойной косвенной адресации.

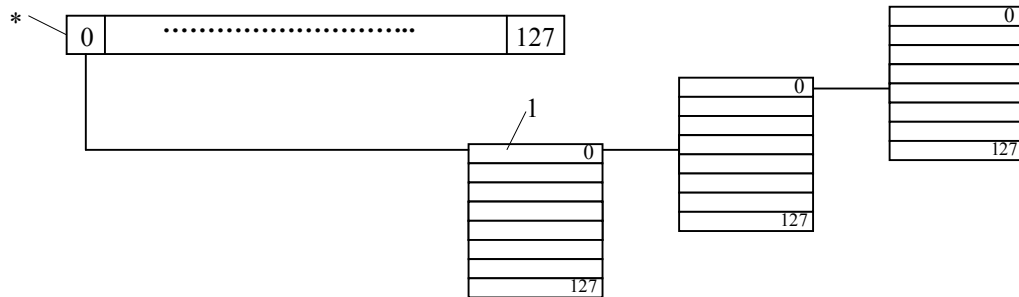


Оптимизация файловой системы

- Расположения БХД на диске
- *Упреждающее считывание*



Список свободных блоков

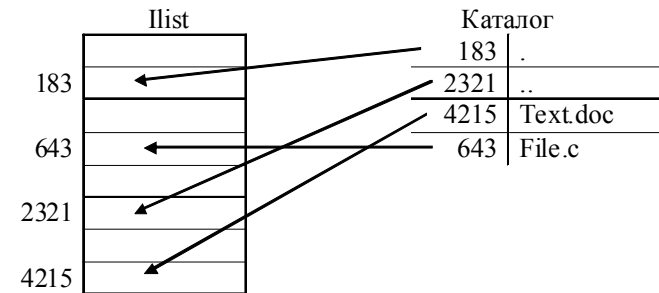


Каталоги s5fs

- Всего не более 65535 файлов (2 байта – 65535 шт.).
- Имя файла – последовательность, содержащая не более 14 символов (цифры и буквы латинского алфавита и символы '_' и '.').
- Имя файла – это способ определения номера *inode*.
- Если файл имеет номер дескриптора равный нулю, то это означает, что запись свободна.
- В каталоге всегда есть имена '.' И '..'.
- '.' Адресует текущий каталог;
- '..' адресует каталог на 1 уровень выше текущего.

Номер inode	Имя
2 байта	14 байт

16 байт



ls -ial

Номер inode	
6104	drwxr-xr-x 5 root root 1024 Jul 22 16:59 .
2	drwxr-xr-x 17 root root 1024 Jun 15 17:17 ..
6528	-rw-r--r-- 1 root root 408 May 5 13:07 .bash_profile
6530	-rwxrwxr-x 1 root root 7168 May 9 12:27 a.out
45073	drwxrwxr-x 7 root root 1024 May 9 12:30 gamma
45069	drwxr-xr-x 5 root root 1024 Jun 15 17:27 lg
6539	-rw-rw-r-- 1 root root 10 Jul 22 17:00 mail.1

Атрибуты каталога

Права	Вход	Прочитать содержимое	Получить метаданные	Добавить/удалить запись
-r	Да	Нет	Нет	Да
-w	Да	Да	Да	Нет
-x	Нет	Да	Нет	Нет

Достоинства и недостатки s5fs

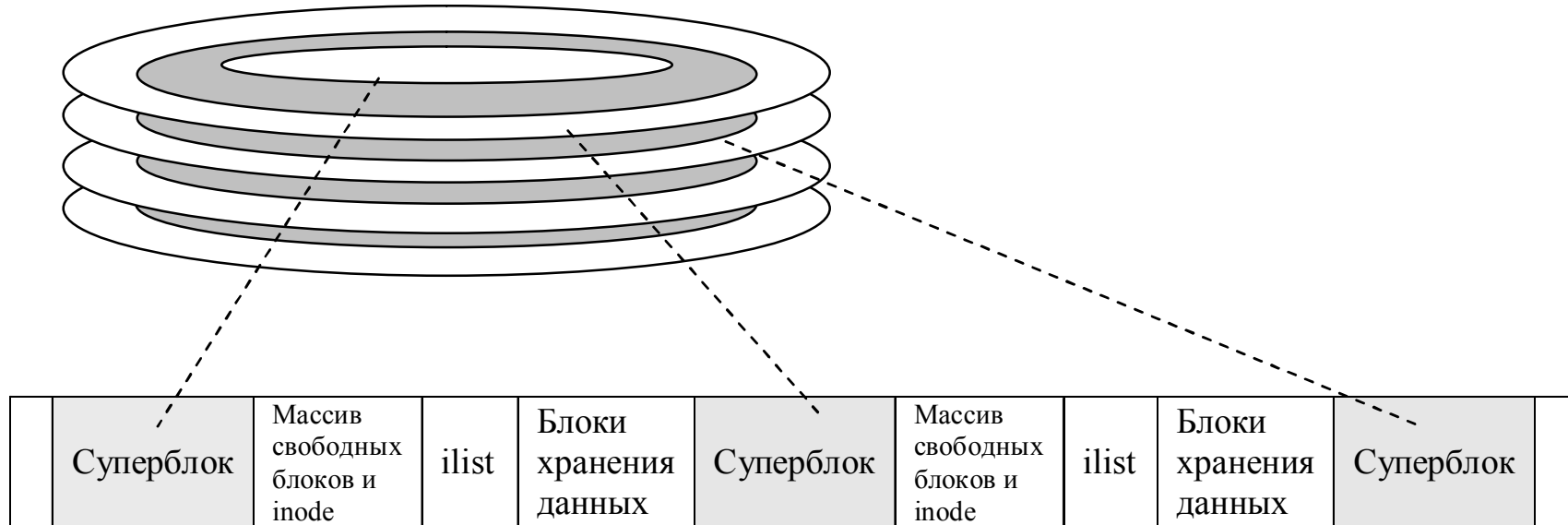
Достоинство - простота (чем проще и логичнее механизм, тем меньше вероятность возникновения ошибки).

Недостатки:

- Низкая производительность. Связана с размещением компонентов ФС системы на диске (обращение как к метаданным, так и к дисковым блокам, разнесенным в пространстве). Фрагментация ФС.
- Низкая надежность. С точки зрения надежности слабым местом этой файловой системы является суперблок..
- Длина имени файлов до 14 символов и общее максимальное число inode (65535).

Файловая система BSD UNIX FFS

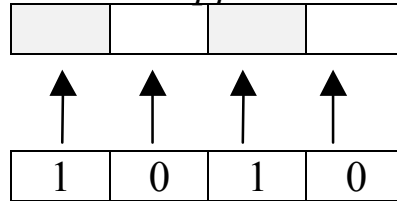
- FFS (Fast File System). Основное отличие FFS от s5fs - в расположении ФС на диске.
- Принцип организации FFS основан на том, что раздел, в котором расположен UNIX, разбивается на логические области – цилиндры (группы цилиндров).
- Каждая группа цилиндров хранит резервную копию суперблока.



FFS

- Список свободных блоков формируется для данной группы цилиндров. Это несколько снижает остроту проблемы фрагментации блоков.
- Файловая система поддерживает большие блоки хранения данных – до 64 кБайт.
- Для большей эффективности использования дискового пространства блоки могут быть разбиты на 2, 4 или 8 частей. Для указания на занятую часть блока используется *битовая карта*. Битовая карта показывает, занят или свободен данный фрагмент. Таким образом, информация о свободном пространстве хранится не в виде списка свободных блоков, как в *s5fs*, а в виде битовой карты блоков.

Разбитый на фрагменты блок



Битовая карта

Каталог FFS

Каталог более **громоздкий**, чем в s5fs.
 Каждая запись каталога состоит из 4 полей. Все поля выравниваются до 4-х байтной границы. Имена файлов – до **255** символов. При этом имена файлов также дополняются нулями до 4-х байтной границы.

Запись содержит следующие поля:

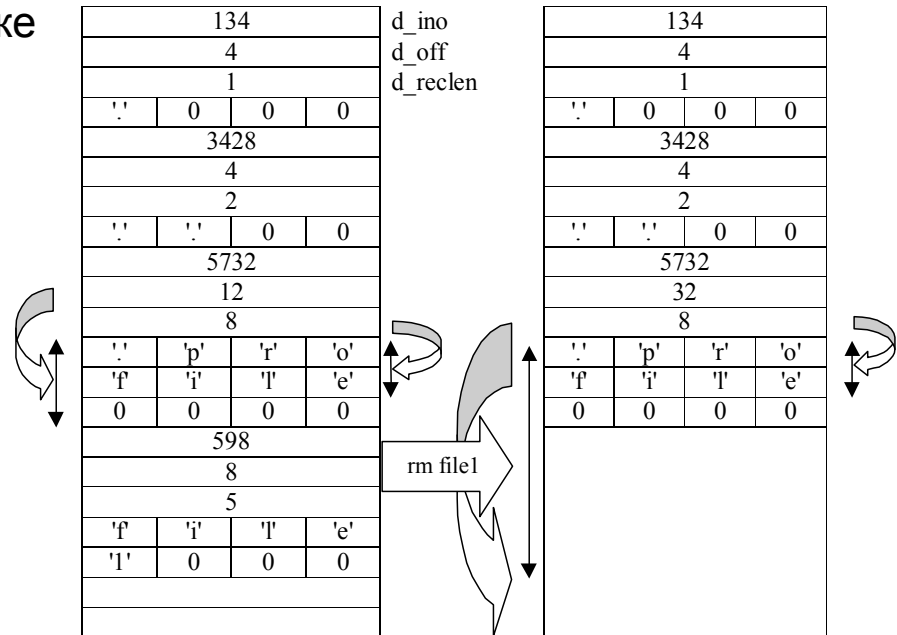
- `d_ino` - номер индексного дескриптора (long);
- `d_off` - длина записи;
- `d_reclen` - длина имени файла;
- `d_name[]` - имя файла.

Структура `dirent`:

```

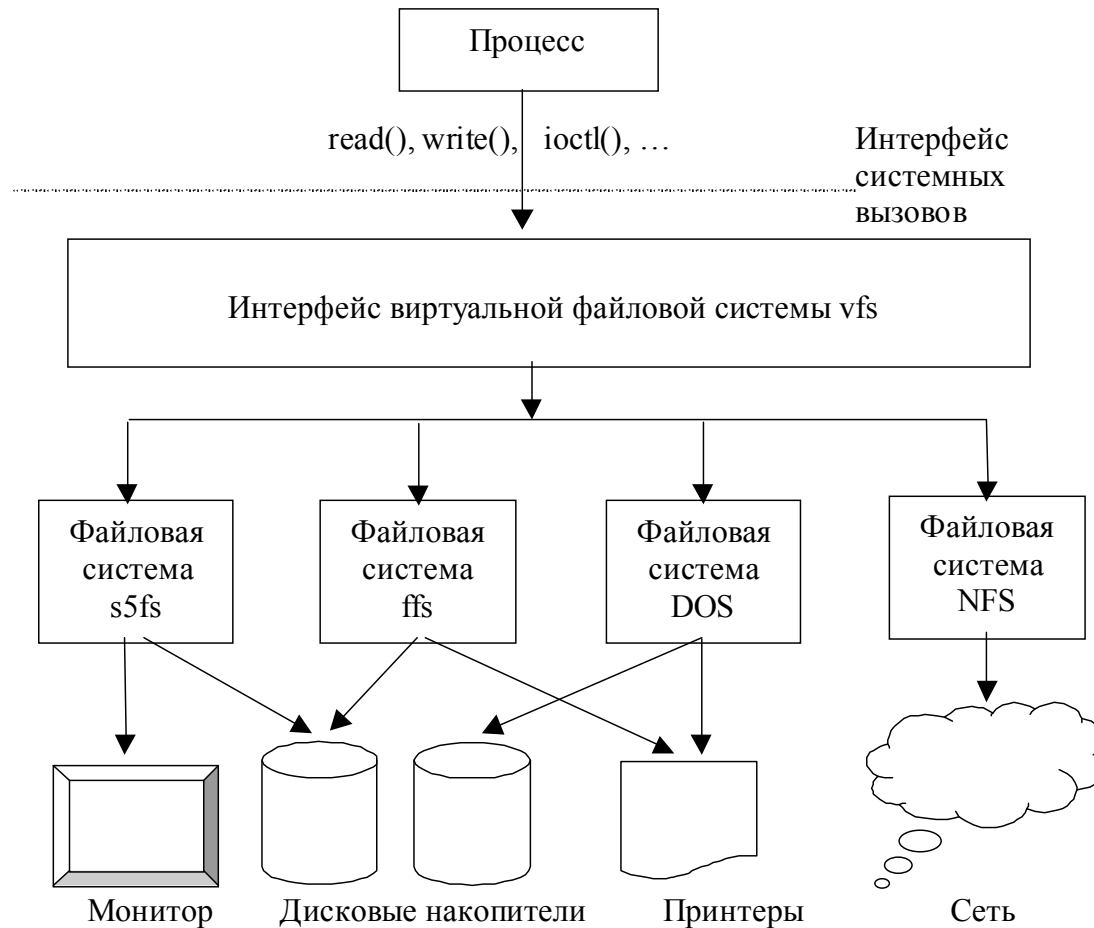
struct dirent
{
    long d_ino;
    off_t d_off;
    unsigned short d_reclen;
    char d_name [NAME_MAX+1];
}
    
```

При удалении имени файла принадлежащая ему запись присоединяется к предыдущей.



Виртуальная файловая система

- UNIX может подсоединять к себе файловые системы других ОС. У каждой системы есть свой список метаданных.
- Общий метаданных список находится в таблице Vnode.



Монтирование внешней ФС

- Определение точки подключения внешней ФС – точки монтирования.
- Обычно используется специальный каталог *mnt*.
- Необходимо определить специальный файл устройства, обслуживающий монтируемый накопитель.
- **Утилита mount**
- Подключение floppy-диска:
 - *mount* /dev/fd0 /mnt/floppy
 - *mount* -t msdos /dev/fd0 /mnt/floppy
- Подключение компакт-диска:
 - *mount* /dev/hdc /etc/mnt (hdc | cd0)
 - *mount* -t msdos /dev/hda1/dos
- Подключение жесткого диска:
 - *mount* /dev/hda5/mnt

```
ls -ial /mnt/document
```

```
total 1344
```

```
4896 drwxrwxr-x 2 root root 16384 Sep 24 01:11 .
```

```
1 drwxrwxr-x 43 root root 16384 Jan 1 1970 ..
```

```
3264244 -rwxrwxr-x 1 root root 56653 Sep 27 10:07 list.arj
```

```
3264243 -rwxrwxr-x 1 root root 276910 Sep 27 10:07 oop.txt
```

```
3264242 -rwxrwxr-x 1 root root 987527 Oct 1 00:39 sysprog.doc
```

Файл `/etc/fstab`

Описание подключаемых и распознаваемых внешних ФС.

Там же можно определить и автоматическое монтирование ФС.

#

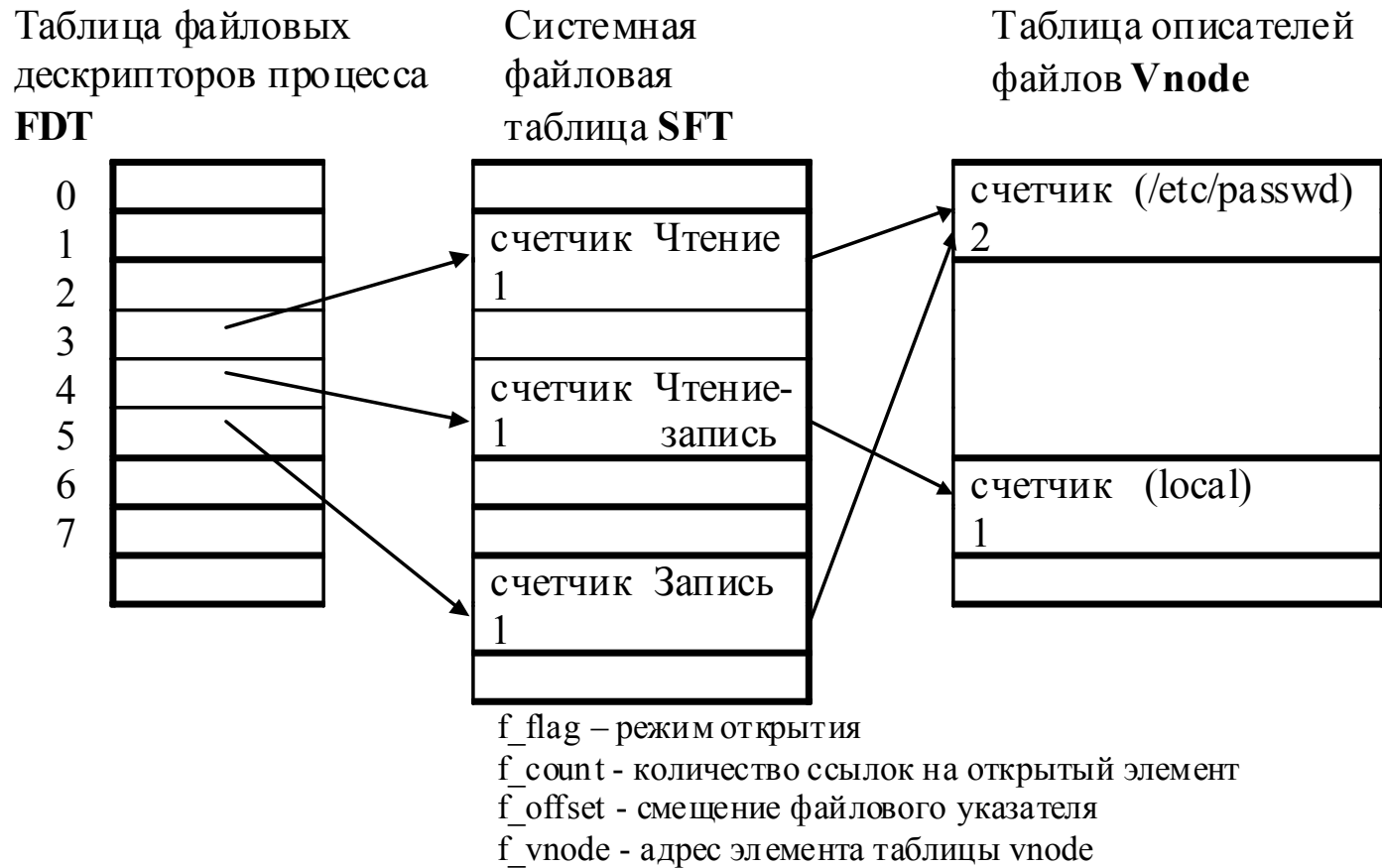
```
# <device> <mountpoint>      <filesystemtype>      <options> <dump> <fsckorder>
/dev/hda4 /                          ext2                    defaults  0       1
/dev/hdb  /mnt/cdrom                    iso9660                ro,noauto 0       0
/dev/fd0  /mnt/floppy                   ext2                    defaults,noauto 0       0
/dev/hda1 none                          msdos                   defaults  0       0
/dev/hda2 none                          ignore                  0          0       0
/dev/hda5 none                          msdos                   defaults  0       0
/proc    /proc                          proc                    defaults
/dev/hda3 none                          swap                    sw
```

- `/etc/fstab` позволяет упростить использование утилиты `mount`.
- Например, монтирование может производиться в более простом варианте, не использующим параметры, уже определенные в `/etc/fstab`:

```
mount          /mnt/floppy
mount          /mnt/cdrom
```

Управление файлами

Доступ к файловой системе



Алгоритм open

ВХОД:

имя файла

режим открытия

права доступа (при создании файла)

ВЫХОД: дескриптор файла

```
{  
    превратить имя файла в идентификатор индекса (алгоритм namei);  
    if (файл не существует или к нему не разрешен доступ) return (код  
        ошибки);  
    выделить для индекса запись в таблице файлов, инициализировать  
        счетчик, смещение;  
    выделить запись в таблице пользовательских дескрипторов файла,  
        установить указатель на запись в таблице файлов;  
    if (режим открытия подразумевает усечение файла) освободить все блоки  
        файла (алгоритм free);  
    снять блокировку (с индекса); /* индекс заблокирован выше, в namei */  
    return (пользовательский дескриптор файла);  
}
```


Пример

1. `fd1 = open("/etc/passwd", O_RDONLY);`
2. `fd2 = open("local", O_RDWR);`
3. `fd3 = open("/etc/passwd", O_WRONLY);`

Таблица открытых
файлов процесса
FDT

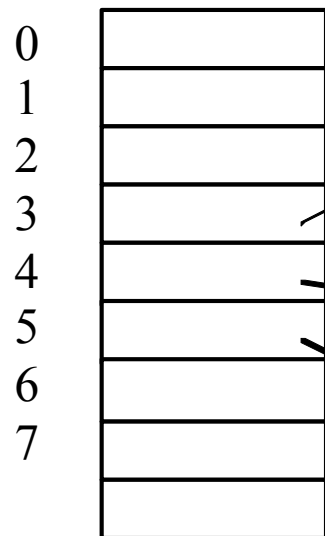


Таблица
файлов
SFT



Таблица описателей
файлов **Vnode**

