Права доступа к файлам

Базовые классы доступа

User (u)

Group (g)

Other (o)

Типы прав

Read (r)

Write (w)

Execute (x)

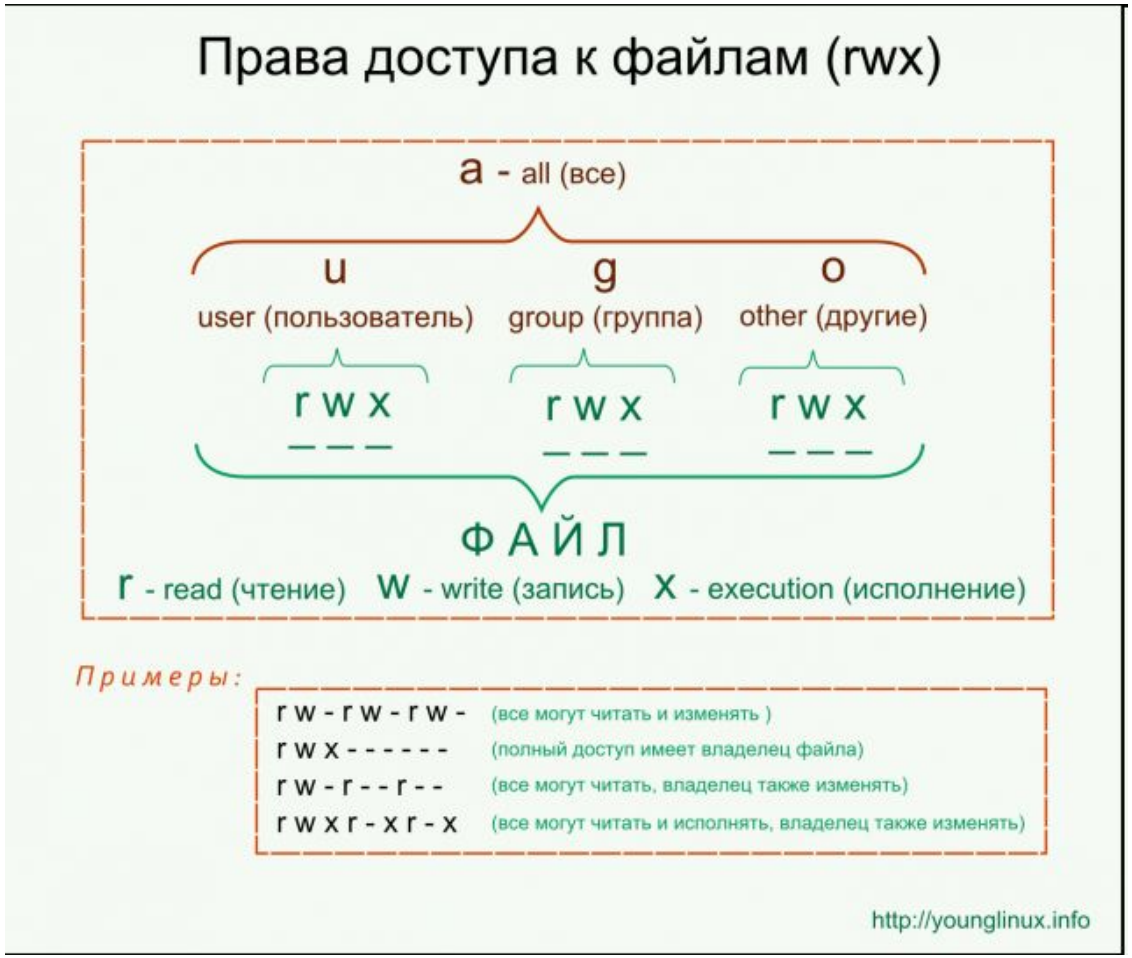
Дополнительные атрибуты

Sticky bit

SUID bit

SGID bit

Права доступа к файлам



Права доступа к файлам

Права доступа к файлам

Права доступа к файлам

Маска прав (mask)

- Это фильтр прав, который определяет права на создаваемые файлы и директории
- У каждого процесса своя маска, которая наследуется от родительского процесса
- Маска состоит из 4 чисел, просмотр и изменение команда **umask**

Права доступа к файлам

Маска прав (mask)

- Права на файл **--- rw- rw- rw- 0666**
- Значение маски **--- --- -w- -w- 0022**
- Результирующие права **--- rw- r-- r-- 0644**

Вычисление: perm AND (NOT (mask))

000 110 110 110 AND (NOT (000 000 010 010))=

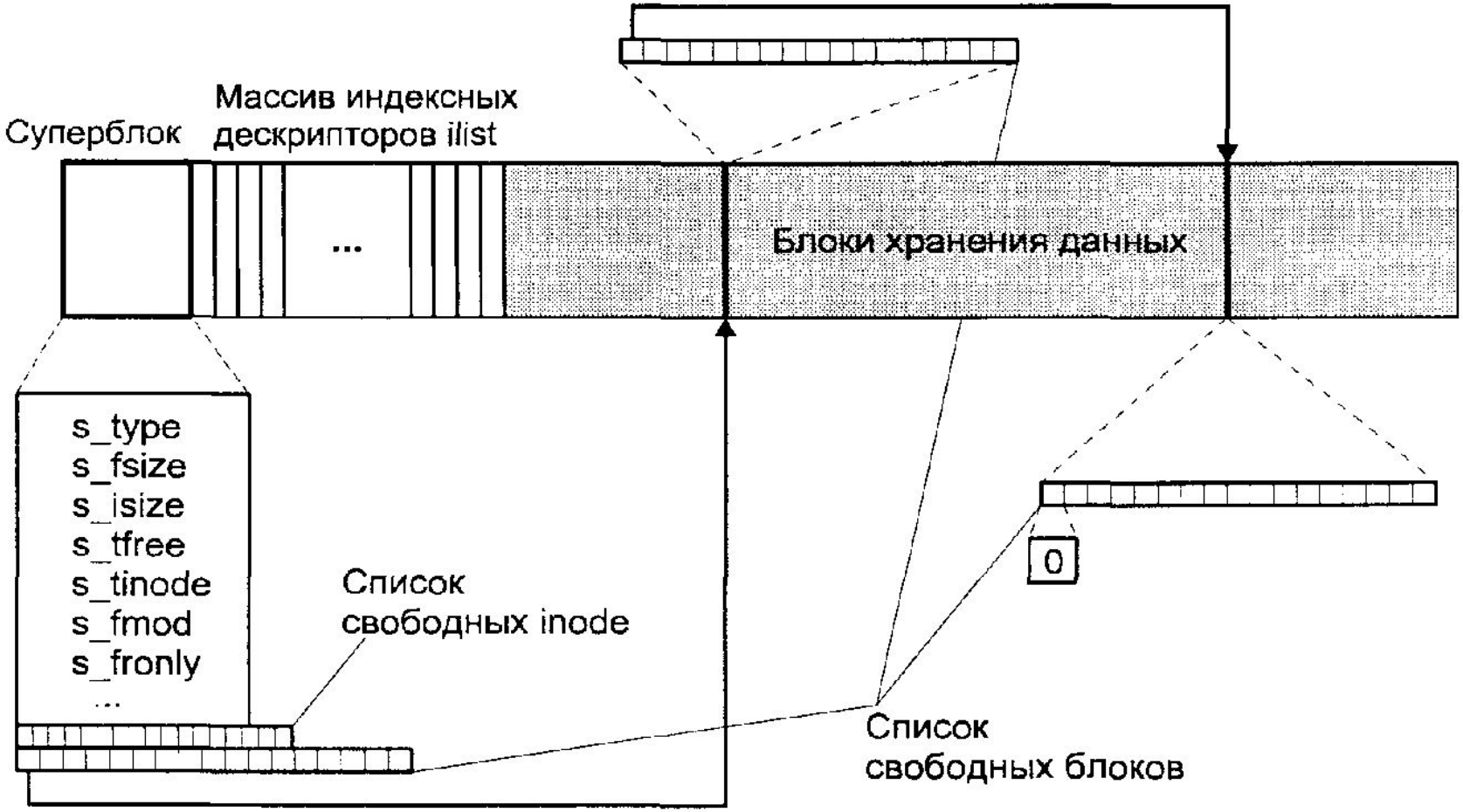
000 110 110 110

111 111 101 101

000 110 100 100 = 0644 --- rw- r-- r--

Физическая структура ФС

Структура фс s5fs

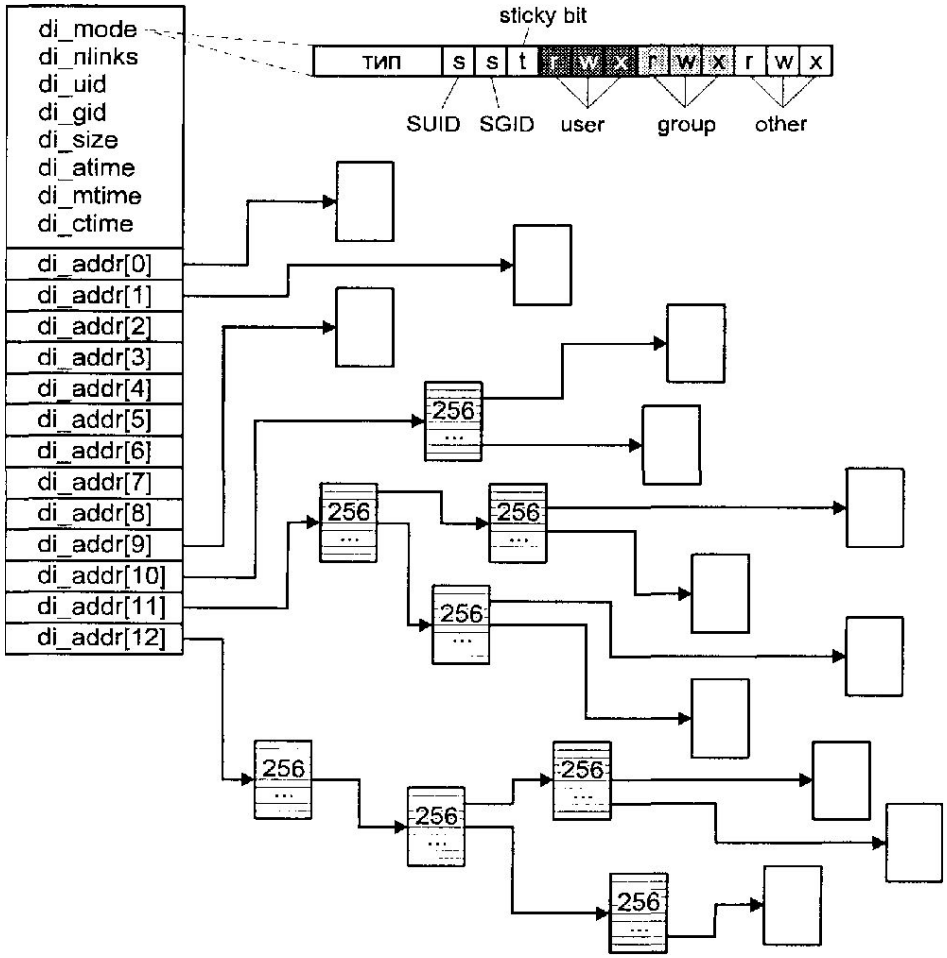


Физическая структура ФС

- Суперблок
- Тип системы
- Размер файловой системы в логических блоках, включая сам суперблок, ilist и блоки хранения данных
- Размер массива индексных дескрипторов
- Число свободных блоков, доступных для размещения
- Число свободных inode, доступных для размещения
- Флаги (флаг модификации флаг режима монтирования
- Размер логического блока (512, 1024, 2048 байт)
- Список номеров свободных inode
- Список адресов свободных блоков
- Индексный дескриптор
- Тип файла, дополнительные атрибуты выполнения и права доступа.
- Число ссылок на файл, т. е. количество которые имеет файл в файловой системе.
- Идентификаторы владельца,пользователя и владельца группы.
- Размер файла в байтах. Для специальных файлов это поле содержит старший и младший номера устройства.
- Время последнего доступа к файлу.
- Время последней модификации.
- Время последней модификации inode (кроме модификации полей di_atime, di_mtime).
- Массив адресов дисковых блоков хранения данных.

Физическая структура ФС

Структура дискового inode

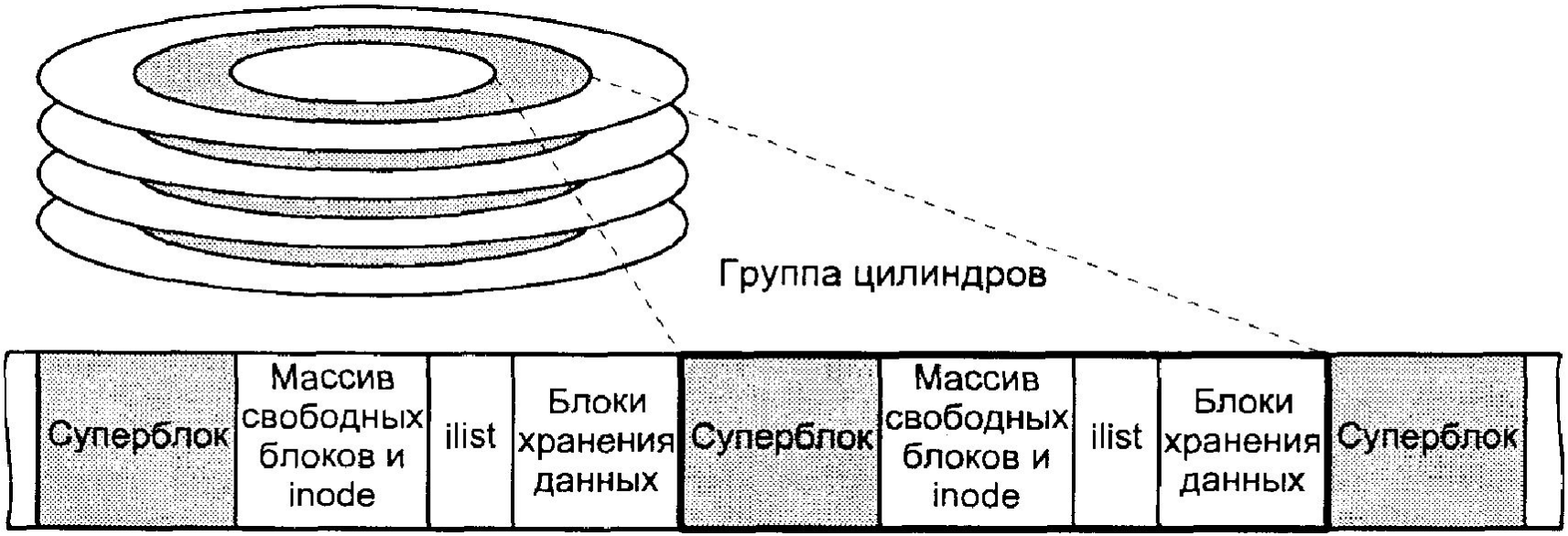


Недостатки ФС S5FS

- Низкая надежность из-за единственного экземпляра суперблока
- Низкая производительность из-за отдаленности блоков с метаданными и самими данными, распределения блоков данных одного файла по всему диску

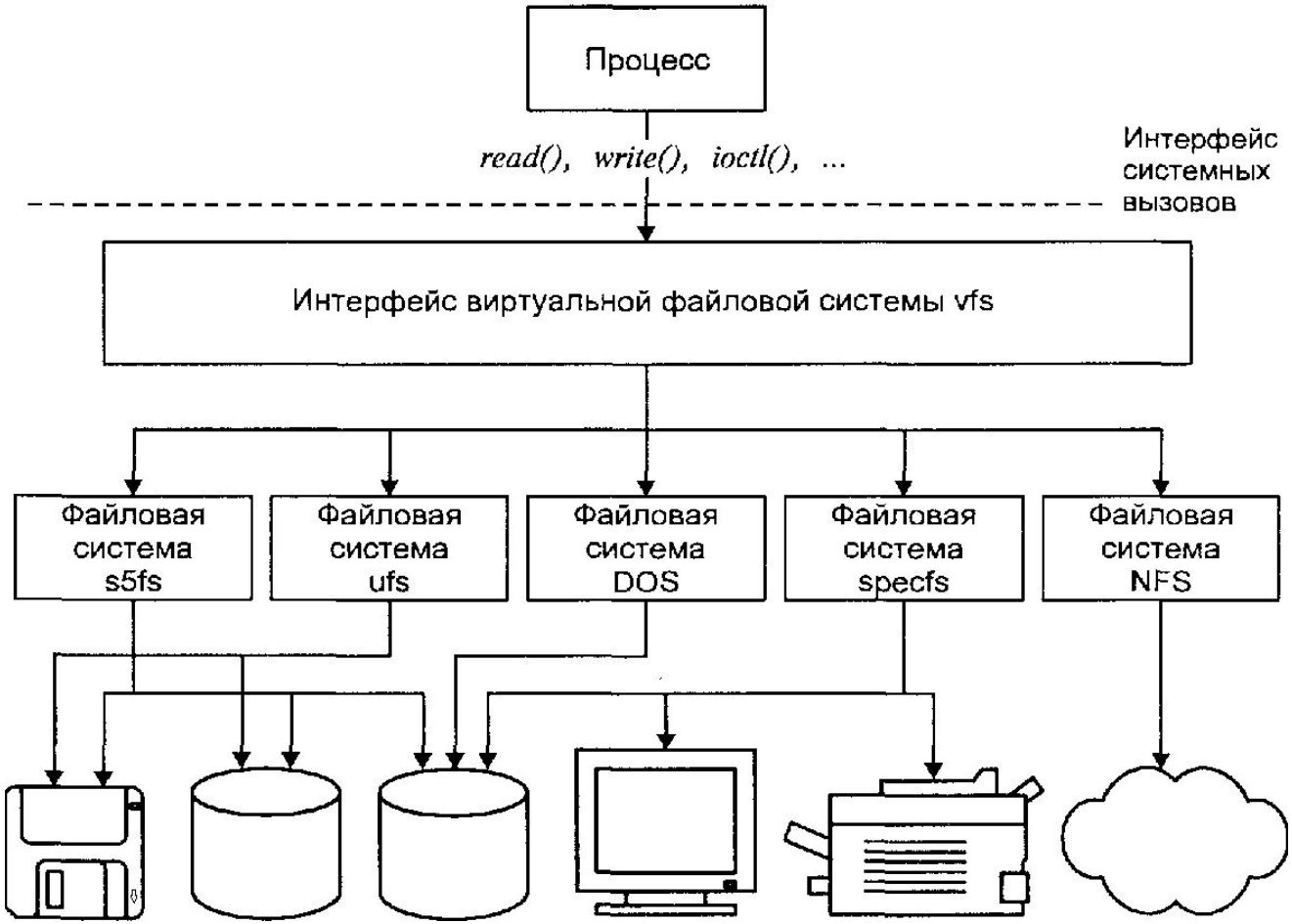
Физическая структура ФС

Структура фс ffs(ufs)



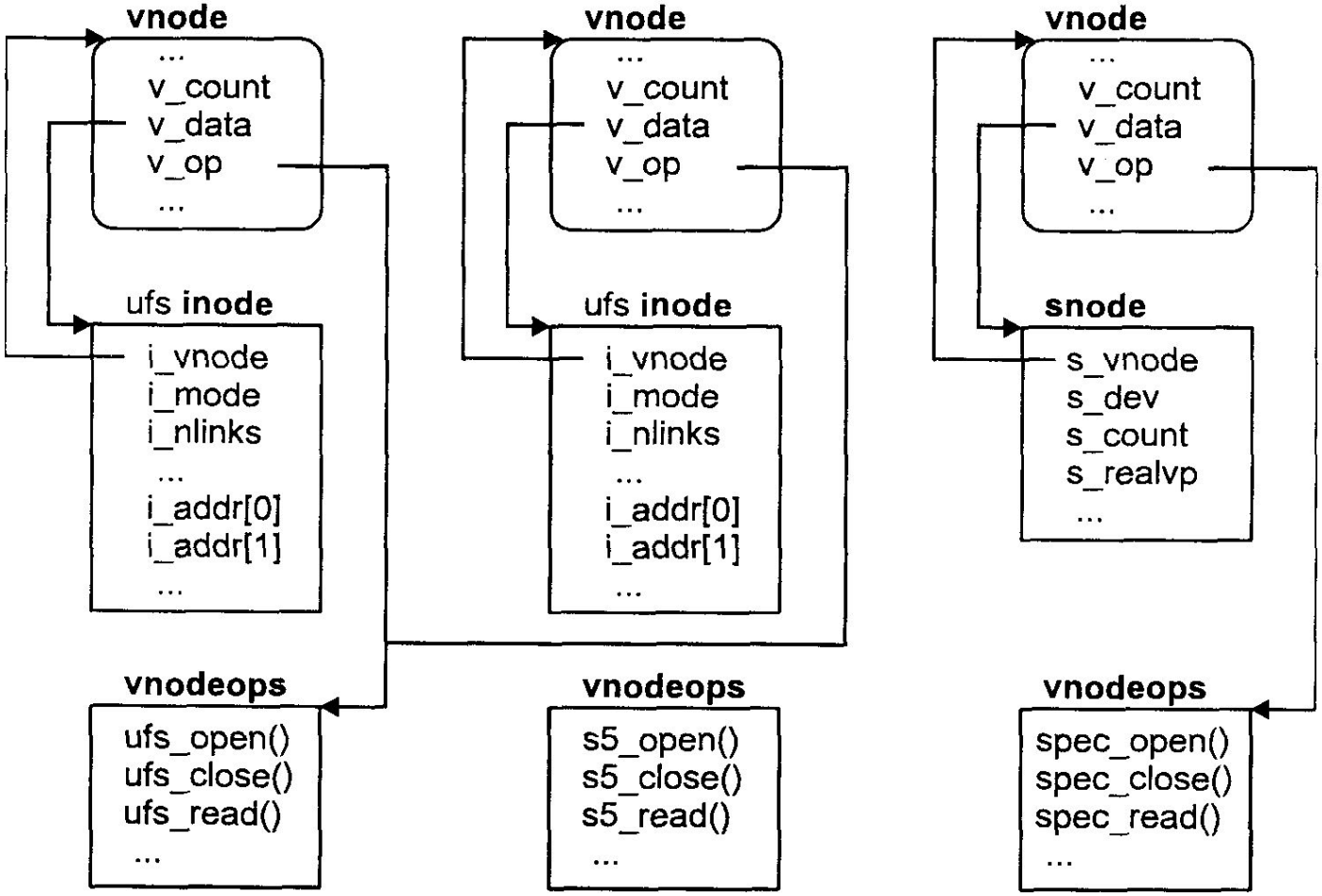
Логическое деление раздела на группы цилиндров
В суперблоке не хранится информация о свободных блоках и inode
Резервные копии суперблока
Распределенные кластеры inode
Увеличенный (до 64кбайт) размер блока
Возможность фрагментации блока
Новая стратегия размещения блоков данных

Архитектура виртуальной файловой системы VFS

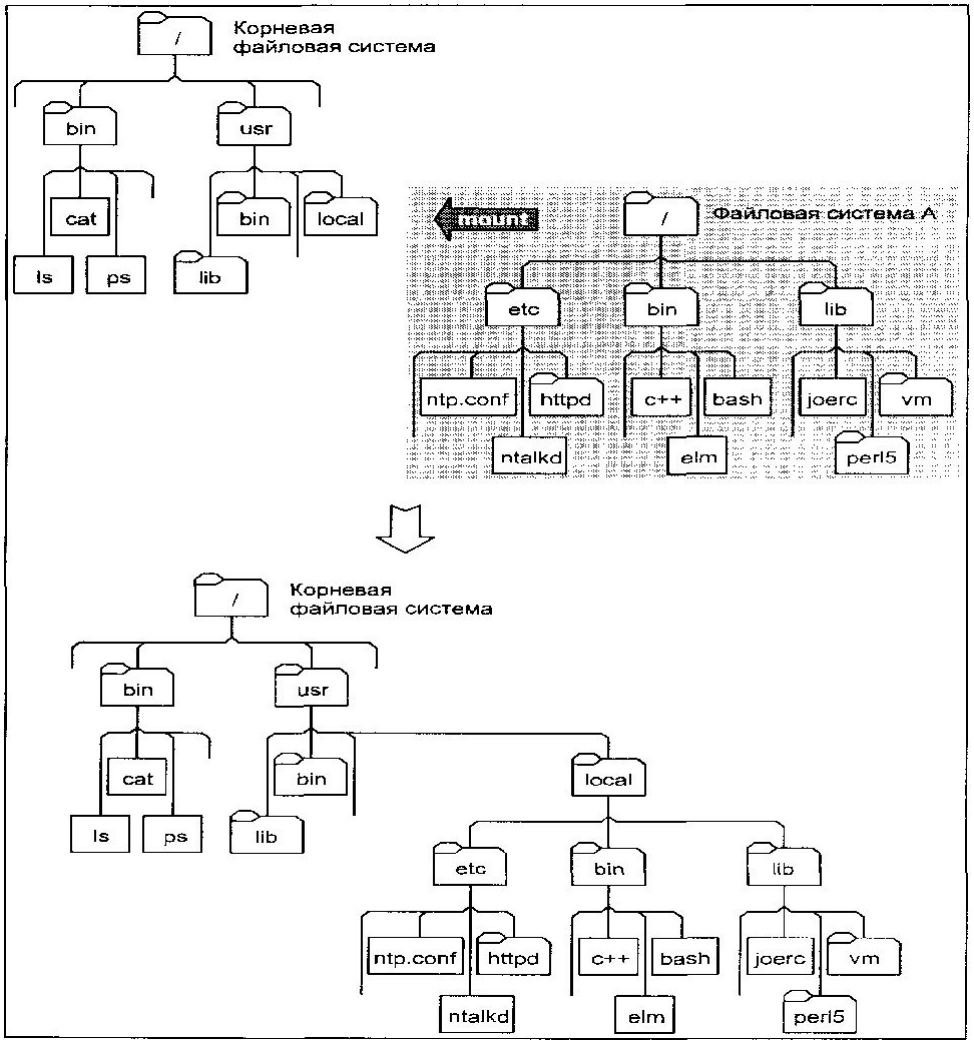


Архитектура виртуальной файловой системы VFS

Метаданные файла виртуальной файловой системы



Монтирование файловых систем



Формат файла /etc/fstab

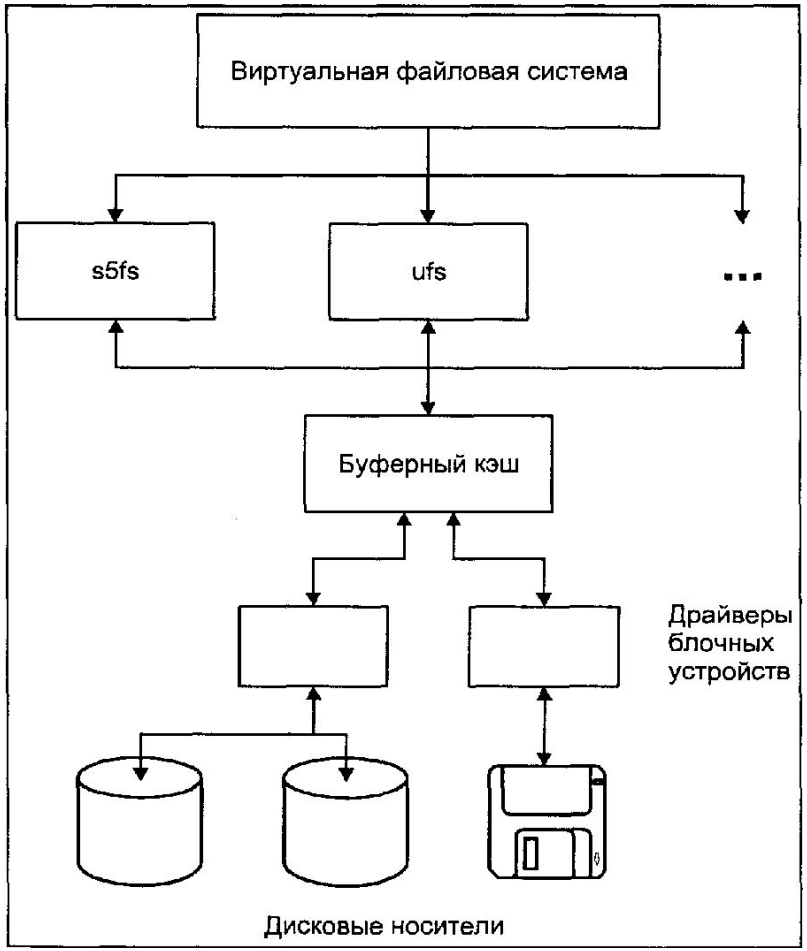
```
/dev/hda5    /              ext2     defaults                                  1     1
/dev/hda7    swap           swap     defaults                          0     0
/dev/fd0     /mnt/floppy    auto     sync, user, noauto, nosuid, nodev, unhide     0     0
/dev/cdrom   /mnt/cdrom     auto     user, noauto, nosuid, exec, nodev, ro         0     0
None         /proc  proc    defaults                          0     0
none         /dev/pts       devpts mode=0622                        0     0
```

Типы файловых систем
по назначению

- 1 Физические файловые системы
  - 1.1 Носители с произвольным доступом: ext2, ext3,ext4,ReiserFS,XFS,Btrfs, ZFS, UFS
  - 1.2 Носители с последовательным доступом:QIC, fifofs
  - 1.3 Оптические носители: ISO9660,HFS,UDF
- 2. Виртуальные файловые системы: /proc, /pipefs
- 3. Сетевые файловые системы: NFS, SMBFS,SSHFS

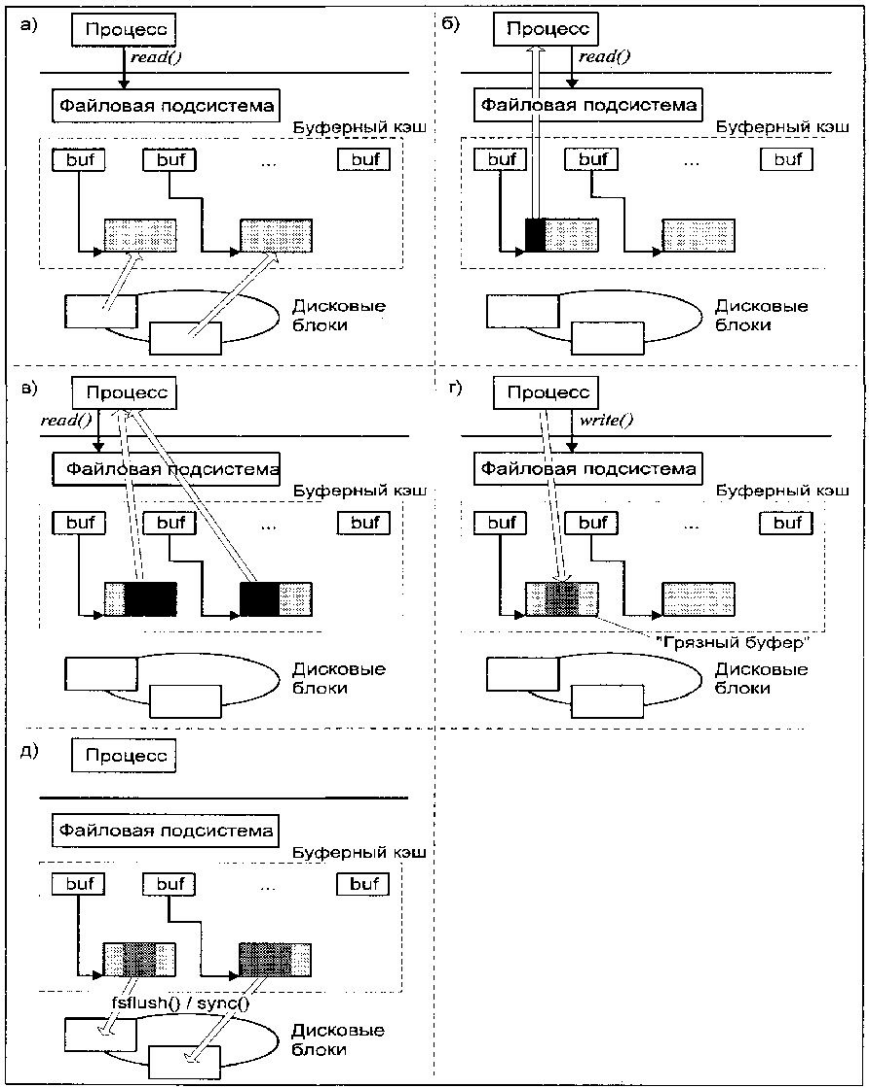
Буферный кэш

Роль буферного кэша



Буферный кэш

Схема работы буферного кэша (read-ahead)



Целостность ФС

Создания жесткой связи для файла.

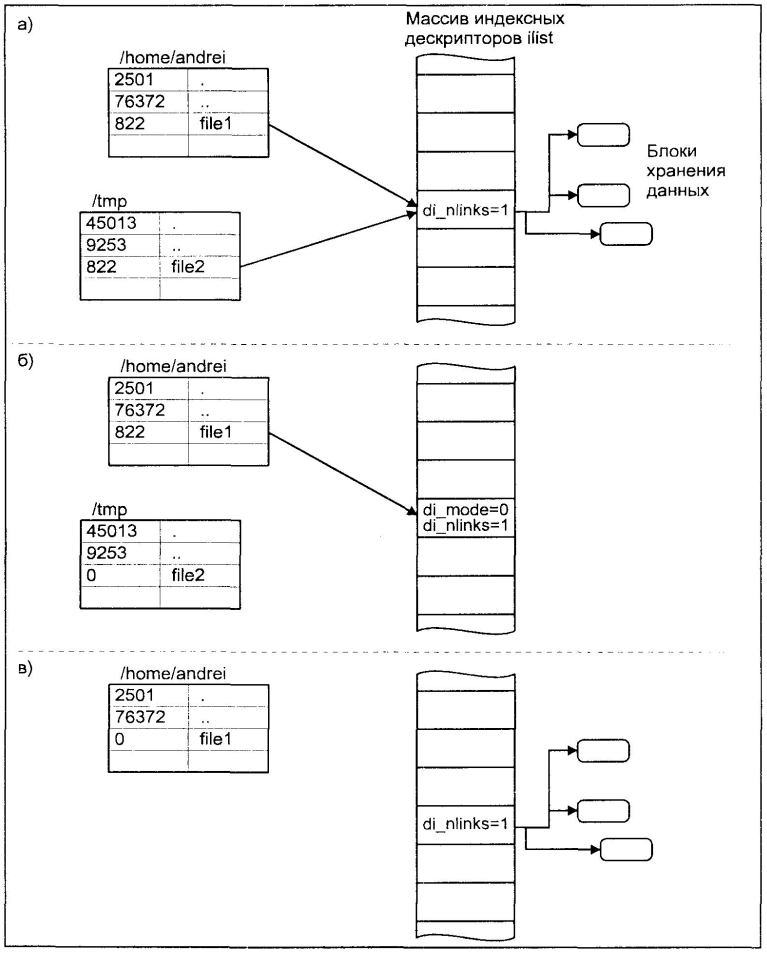
1. Создать новую запись в необходимом каталоге
2. Увеличить счетчик связей в inode

Сбой между п.1 и п.2: удаление одной ссылки приводит к указанию второй ссылки на неразмещенный inode (а,б)

Создания жесткой связи для файла.

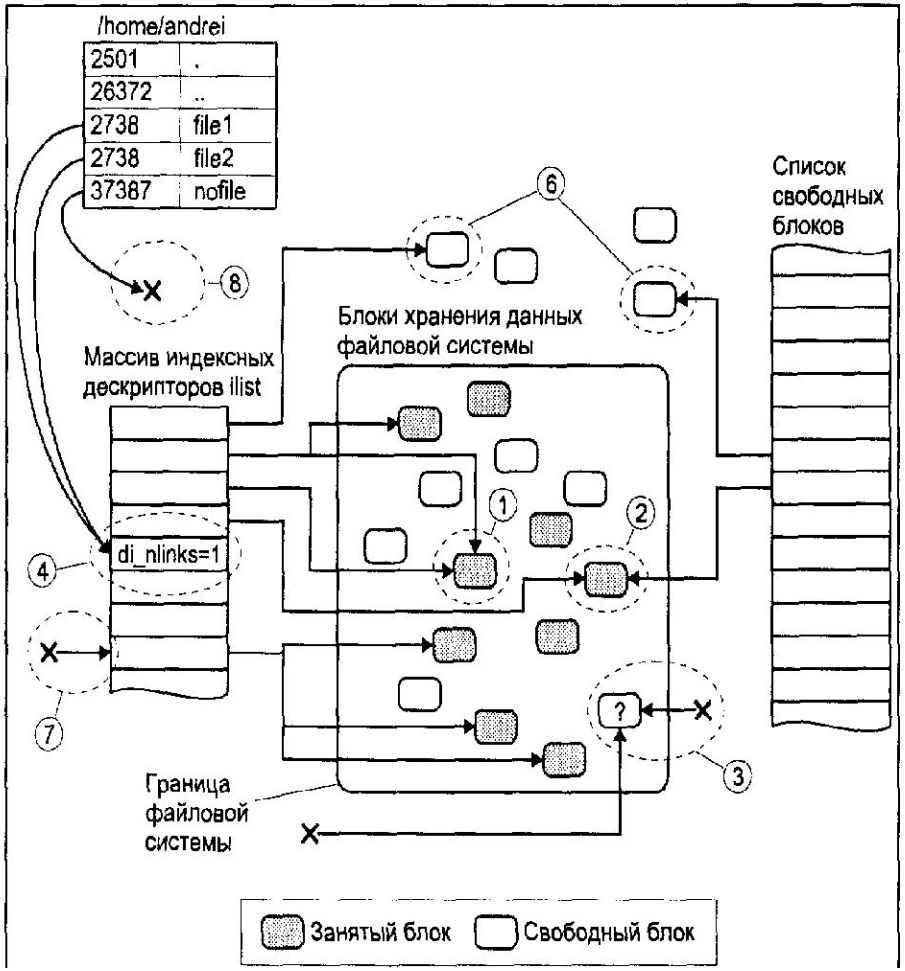
1. Увеличить счетчик связей в inode
2. Создать новую запись в необходимом каталоге

Сбой между п.1 и п.2: удаление ссылки не приводит к очистке диска.(в)



Целостность ФС

Отсутствие синхронизации между образом файловой системы в памяти и ее данными на диске в случае аварийного останова
1. Один блок адресуется несколькими mode (принадлежит нескольким файлам).
2. Блок помечен как свободный, но в то же время занят (на него ссылается inode)
3. Блок помечен как занятый, но в то же время свободен (ни один inode на него не ссылается).
4. Неправильное число ссылок в inode (недостаток или избыток ссылающихся записей в каталогах).
5. Несовпадение между размером файла и суммарным размером адресуемых inode блоков.
6. Недопустимые адресуемые блоки (например, расположенные за пределами файловой системы).
7. "Потерянные" файлы (правильные inode, на которые не ссылаются записи каталогов).
8. Недопустимые или неразмещенные номера inode в записях каталогов.



Для исправления ошибок на ФС существует утилита fsck. Проверка и исправления должны производится на размонтированной ФС.

Журнализация ФС

- Для операционной системы рассогласование между буферным кэшем и блоками хранения данных отдельных файлов, не приведет к катастрофическим последствиям даже в случае внезапного останова системы, хотя с точки зрения пользователя все может выглядеть иначе. Содержимое отдельных файлов не вносит существенных нарушений в целостность файловой системы.
- Другое дело, когда подобные несоответствия затрагивают метаданные файла или другую управляющую информацию файловой системы, например,суперблок.

Журнализация ФС

- Важное свойство файловых систем — поддержка журнализации. Журналируемая файловая система ведёт постоянный учёт всех операций записи на диск. Благодаря этому после сбоя электропитания файловая система всегда автоматически возвращается в рабочее состояние.
- **Файловые системы с журнализацией**:
- 1) системы, производящие журнализацию всех изменений,
- 2) системы, журнализующие только изменения метаданных.
- **Типы журналов**:
- 1) журнал, ориентированный только на повторное выполнение операций (redo-only)
- 2) журнал, способный поддерживать как повторное выполнение операций, так и их обратное выполнение (undo-redo).