

Лекция 6

Файловая система

Блокировки файлов (file locking)

- Advisory (рекомендательная) — для процессов, которые добровольно соглашаются соблюдать блокировки
 - Процесс может игнорировать блокировки других процессов
- Mandatory (обязательная) — для любых процессов
 - Не везде поддерживаются
 - Требуют специального монтирования файловой системы

Типы блокировки

- Read (shared) — блокировка на чтение. Несколько процессов могут заблокировать ресурс на чтение, при условии, что нет блокировок на запись
- Write (exclusive) — единственный процесс блокирует на запись, нет блокировок на чтение
- Если требуемый тип блокировки не может быть немедленно удовлетворен, процесс переводится в состояние ожидания или блокировка завершается ошибкой

СИСТЕМНЫЙ ВЫЗОВ fcntl

```
struct flock {  
    short l_type;      /* F_RDLCK, F_WRLCK, F_UNLCK */  
    short l_whence;    /* SEEK_SET, SEEK_CUR, SEEK_END */  
    off_t l_start;     /* Starting offset for lock */  
    off_t l_len;       /* Number of bytes to lock */  
    pid_t l_pid;       /* PID of process blocking our lock  
(F_GETLK only) */  
};
```

```
struct flock flk;
```

```
int res = fcntl(fd, OPER, &flk);
```

```
// OPER – один из F_SETLK, F_SETLKW, F_GETLK
```

Операции fcntl

- F_SETLK
 - F_RDLCK — заблокировать на чтение
 - F_WRLCK — заблокировать на запись
 - F_UNLCK — разблокировать
 - Если операция невозможна, возвращается ошибка EACCESS или EAGAIN
- F_SETLKW
 - Те же операции блокировки
 - Если операция невозможна, процесс блокируется
- F_GETLK
 - Проверить возможность блокировки
 - Если блокировка невозможна, получить информацию о процессе

Особенности fcntl

- Per-process, т. е. каждый процесс может иметь только один тип блокировки на каждый конкретный байт файла
- Advisory, т. е. системные вызовы read и write не проверяют наличие и тип блокировки
- При закрытии любого файлового дескриптора, связанного с файлом, в этом процессе блокировки процесса сбрасываются

Mandatory locking

- В Linux необходимо выполнить следующее:
 - Разрешить обязательные блокировки при монтировании
`mount DEVICE PATH -o mand`
 - Файл не должен иметь разрешение исполнения на группу (бит 010 прав)
 - Файл должен иметь sgid бит (02000)
- Тогда read write будут проверять блокировку файлов и переводить процесс в состояние ожидания в случае конфликта

Квотирование

- Можно квотировать (ограничить число):
 - Индексных дескрипторов (т. е. файлов)
 - Блоков данных (т. е. суммарный размер файлов)
- Типы квоты:
 - Hard — ограничение не может быть превышено
 - Soft — ограничение может быть превышено на ограниченное время (grace period)
- Квоты могут применяться к пользователю и группе

Управление квотированием

- Квотирование включается при монтировании файловой системы
`mount DEV PATH -o usrquota,grpquota`
- Базы данных квотирования создаются и управляются с помощью
`quotacheck OPTIONS`
- Квота для пользователя (группы) редактируется с помощью
`edquota`

Типы файловых систем

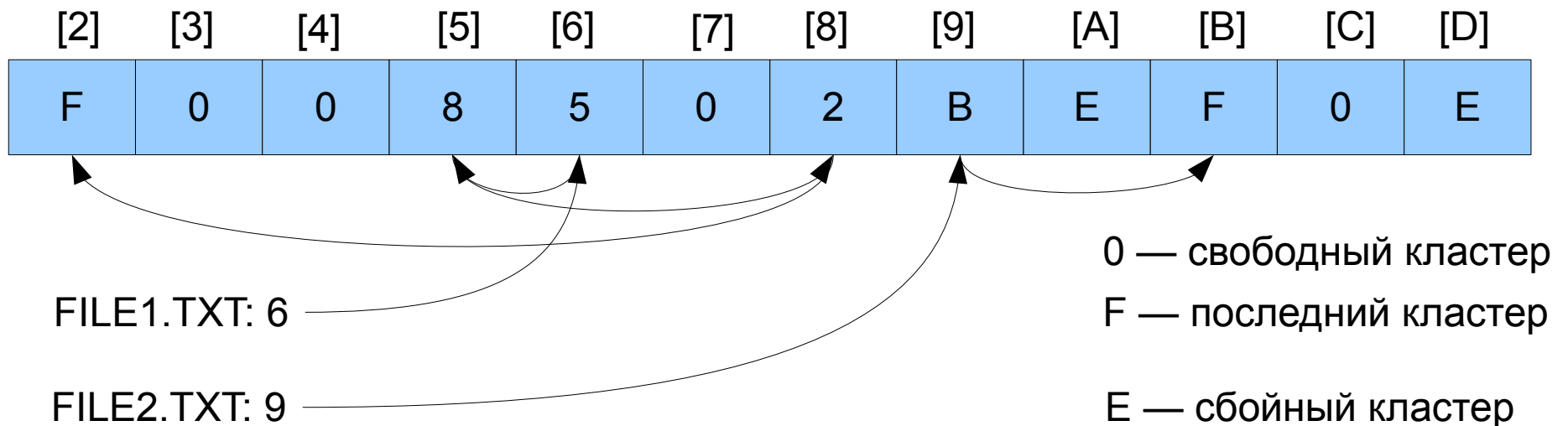
- Поддержка иерархии каталогов (иерархические/не иерархические)
- Поддержка журналирования
- Поддержка особенностей хранения данных (dvd, flash)

Файловая система RT-11

- Одноуровневая иерархия файлов
- Файлы хранятся в непрерывных областях области данных диска
- Имена файлов — 6 + 3 заглавные латинские буквы и цифры (кодируется в 6 байтах)
- Записи о файлах в каталоге диска располагаются в порядке размещения файлов в области данных диска, специальные записи для «дыр»

FAT

- Иерархическая файловая система
- Имена файлов: 8 + 3 (символы занимают один байт)
- Таблица размещения файлов:



FAT

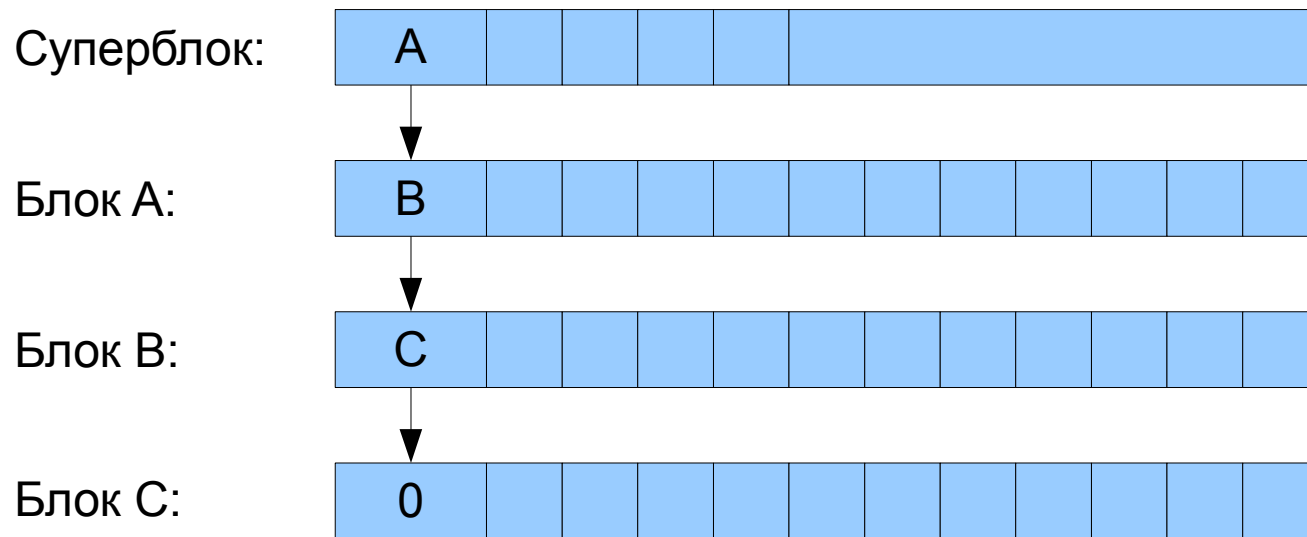
- Метаинформация о файле хранится в записях каталога
- FAT16: Максимальный размер кластера — 32 KiB (при размере блока 512 байт — 64 блока на кластер)
- Для надежности на диске хранится две копии FAT
- Для эффективной работы в памяти приходится держать FAT целиком
- Фрагментация файлов

UNIX System V FS (s5fs)

Загрузчик	Суперблок	Область инд. дескр.	Область данных
-----------	-----------	---------------------	----------------

- Суперблок хранит информацию о файловой системе:
 - Размер файловой системы в блоках
 - Размер области индексных дескрипторов (inode) в блоках
 - Число свободных блоков и инд. дескр.
 - Номер первого свободного инд. дескр.
 - Список свободных блоков данных (частично)
- Загружается в память при монтировании

Список свободных блоков



- При удалении блока он добавляется в начало списка
- При выделении блока он берется из начала списка

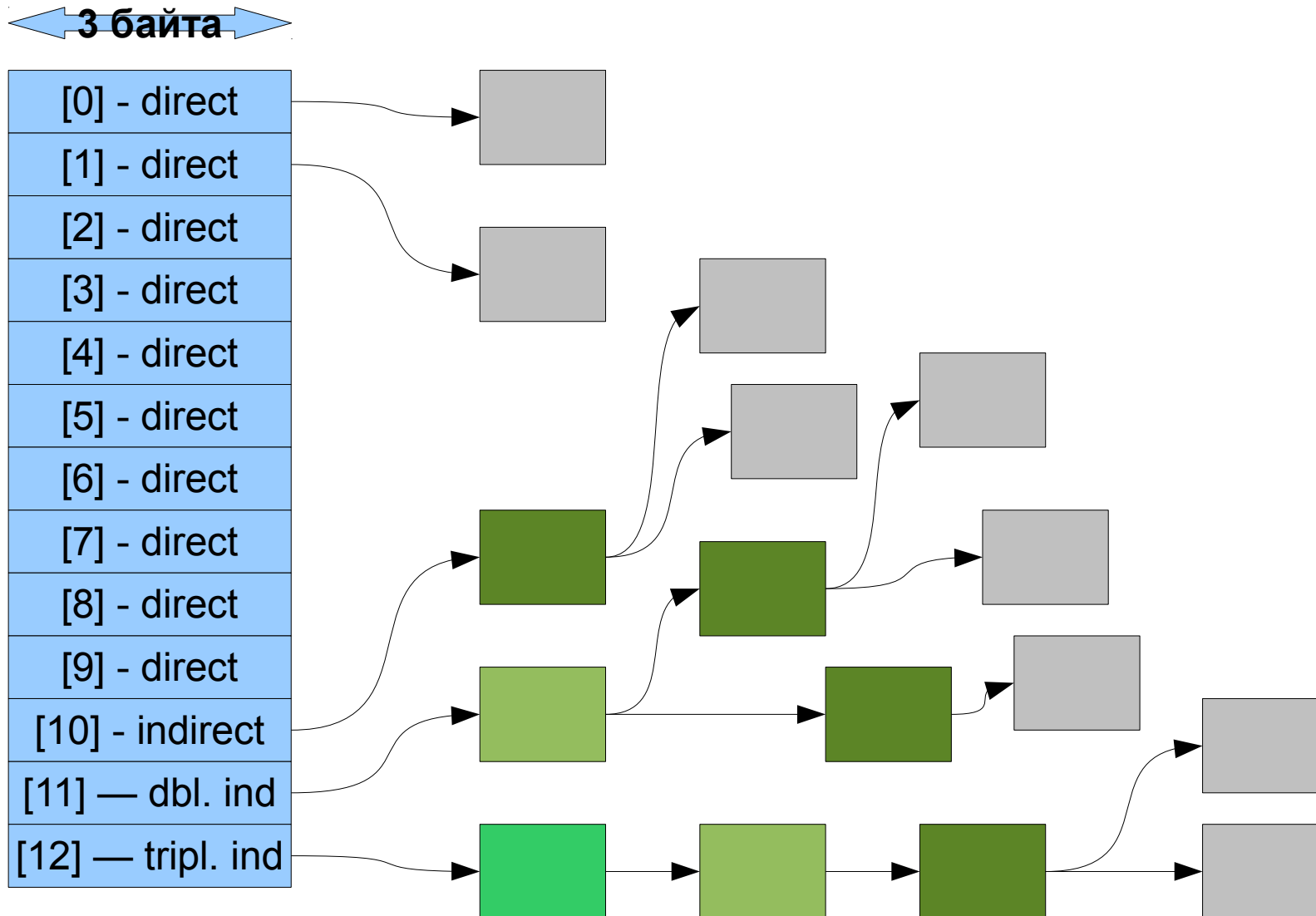
Индексный дескриптор (inode)

Поле	Размер	Описание
di_mode	2	Права доступа и тип файла
di_nlinks	2	Число ссылок на этот и. д.
di_uid	2	Идентификатор пользователя
di_gid	2	Идентификатор группы
di_size	4	Размер файла
di_addr	39	Массив адресов блоков данных
di_gen	1	Поколение
di_atime	4	Время посл. доступа к файлу
di_mtime	4	Время модификации файла
di_ctime	4	Время создания файла

Размер — 64 байта

Индексный дескриптор в памяти содержит дополнительные поля!

Массив адресов блоков



Размеры файлов

- Размер блока — 512 байт
- 10 непоср. номеров блоков — 5 KiB
- Номер косв. блока - 128 номеров блоков
 - Итого: $10 + 128 = 138$ блоков
- Номер двойного косв. блока - 128^2 блоков
 - Итого: $10 + 128 + 128^2 = \sim 8$ MiB
- Номер тройного косв. блока - 128^3 блоков
 - Итого: $10 + 128 + 128^2 + 128^3 = 1$ GiB

Структура каталога

- Каталог — файл, содержащий список файлов и каталогов
- Каждая запись в каталоге — 16 байт
 - Имя файла — 14 байтов
 - Номер индексного дескриптора — 2 байта

Недостатки

- Суперблок может быть поврежден
- Размер блока недостаточный (низкая скорость передачи)
- Блоки файлов и каталогов разбросаны по диску
- Индексные дескрипторы находятся далеко от блоков данных

Файловая система Ext2 (Linux)

Загрузочный Сектор	Группа блоков 1	Группа блоков 2	Группа блоков 3	Группа блоков N
--------------------	-----------------	-----------------	-----------------	-----------------

Группа блоков:

Суперблок	Дескриптор ФС	Карта своб. блоков	Карта своб. и.д.	Массив и.д.	Блоки данных
-----------	---------------	--------------------	------------------	-------------	--------------

- Размер блока данных: 1024, 2048, 4096 байт
- Номера индексных дескрипторов и блоков — 32 битные беззнаковые
- Размер индексного дескриптора — 128 байт
- Запись в каталоге имеет переменный размер (до 256 с)

Ext2 (1993)

- В inode хранится 12 прямых ссылок на блоки, indirect, double indirect, triple indirect (каждая — 32 бита)
- Если длина symlink < 60 байт, то он хранится в inode
- Свободные блоки хранятся в битовом множестве

Журналирование

- Обеспечение целостности файловой системы в случае краха ОС или сбоя питания
- Журнал — специальная область на диске
- Каждая операция, модифицирующая данные, выполняется в три стадии:
 - В журнал записывается операция (с флагом невыполненной)
 - Выполняется операция
 - Операция в журнале помечается как выполненная

Ext3 (2001)

- Совместима снизу вверх с ext2
- Обеспечивает журналирование
- Индексирование больших каталогов (htree)
- Максимальный размер файла увеличен до $2 \text{ TiB} = 2 * 1024 \text{ GiB}$

Ext4 (2008)

- Макс. размер файловой системы до 2^{60}
- Макс. размер файла — 16TiB
- Extends вместо отображения блоков
 - До 4 на каждый inode, 128 MiB каждый
 - Оптимизация размещения на диске больших файлов
- Отметки времени с точностью до наносекунд
- 34 бита на секундную часть времени

Работа со временем

```
#include <time.h>
```

```
time_t time(time_t *tptr);
```

```
// получить время в секундах
```

```
struct tm *localtime(time_t *tptr);
```

```
time_t mktime(struct tm *ptm);
```

Struct tm

```
struct tm {  
    int tm_sec;           /* seconds */  
    int tm_min;           /* minutes */  
    int tm_hour;          /* hours */  
    int tm_mday;          /* day of the month */  
    int tm_mon;           /* month */  
    int tm_year;          /* year */  
    int tm_wday;          /* day of the week */  
    int tm_yday;          /* day in the year */  
    int tm_isdst;         /* daylight saving time */  
};
```

mktime

- Работает в локальной временной зоне
- Для корректной обработки летнего времени при вызове mktime поле `tm_isdst` должно быть -1
- Возвращает -1, если время не представимо в `time_t`
- Нормализует значения полей `tm_year`, `tm_mon`, `tm_mday`, `tm_hour`, `tm_min`, `tm_sec`
- Заполняет `tm_wday`, `tm_yday`