

OC(UNIX)

(ФС ext2)

ФС ext2

ext2 - Наиболее богатая функциональными возможностями файловая система из семейства совместимых с Linux.

На данный момент считается самой популярной системой.

- ext - Предыдущая версия системы ext2, не совместима с последующими

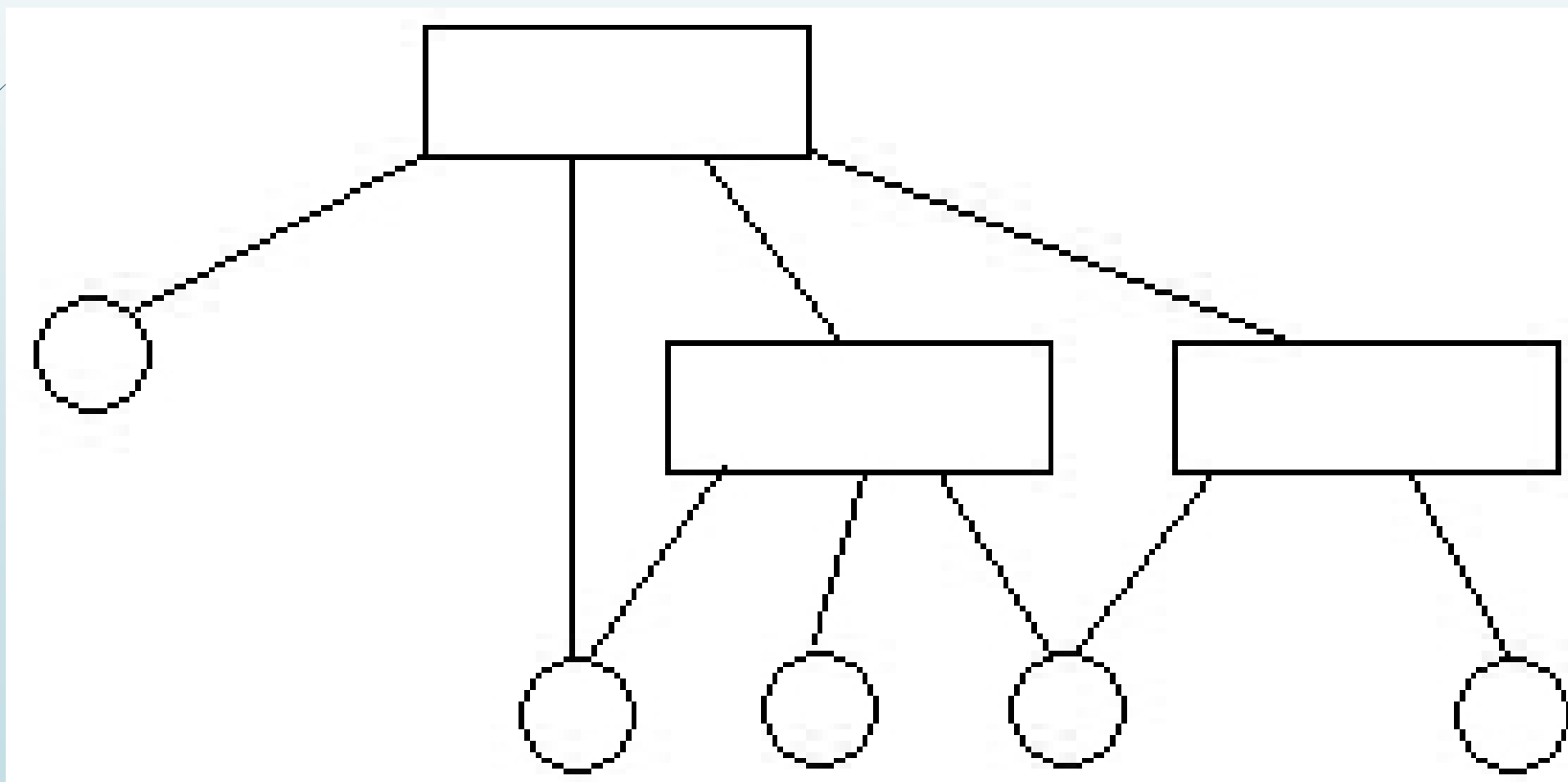
Логическая организация файловой системы ext2

- иерархию каталогов ФС ext2 представляет собой сеть,
- это достигается тем, что один файл может входить сразу в несколько каталогов, т.е. иметь несколько полных имен;
- здесь справедливо соответствие **«один файл — много полных имен»**.
- полное имя однозначно определяет файл.

ФС ext2

Логическая организация файловой системы ext2

► иерархия каталогов ФС ext2



ФС ext2

Атрибуты ФС ext2 :

- - тип и права доступа к файлу;
- - владелец, группа;
- - информация о разрешённых операциях доступа к файлу;
- - время создания, последнего доступа, последнего изменения и время последнего удаления;
- - текущий размер файла;

ФС ext2

Атрибуты ФС ext2 :

- - тип файла:
 - обычный файл;
 - каталог;
 - файл байт-ориентированного устройства;
 - файл блочно-ориентированного устройства;
 - сокет;
 - именованный канал;
 - символическая ссылка;
- - число блоков, занимаемых файлом;
- - другие.

ФС ext2

Атрибуты файлов хранятся не в каталогах, а в специальных таблицах.

каталог имеет простую структуру, состоящую из двух частей: **номера индексного дескриптора и имени файла.**

Физическая организация файловой системы ext2

Структура дискового раздела:

- - блоки и группы блоков;
- - массив индексных дескрипторов;
- - суперблок.

Физическая организация файловой системы ext2

Всё пространство раздела диска разбивается на блоки фиксированного размера, кратные размеру сектора — 1024, 2048 и 4096 байт.

Размер блока указывается при создании ФС на разделе диска.

Физическая организация файловой системы ext2

С целью уменьшения фрагментации и количества перемещений головок жёсткого диска при чтении больших массивов данных блоки объединяются в группы блоков

ФС ext2

Обобщенная структурная схема ФС ext2



ФС ext2

индексный дескриптор
(информационный узел),
information node, или **inode**.

содержит информацию об атрибутах и
физическом расположении файла.

ФС ext2

Суперблок — основной элемент файловой системы ext2, содержит общую информацию о ФС:

- - общее число блоков и индексных дескрипторов в файловой системе;
- - число свободных блоков и индексных дескрипторов в файловой системе;
- - размер блока файловой системы;

ФС ext2

Суперблок — основной элемент файловой системы ext2, содержит общую информацию о ФС:

- -- количество блоков и индексных дескрипторов в группе;
- - размер индексного дескриптора;
- - идентификатор файловой системы.

ФС ext2

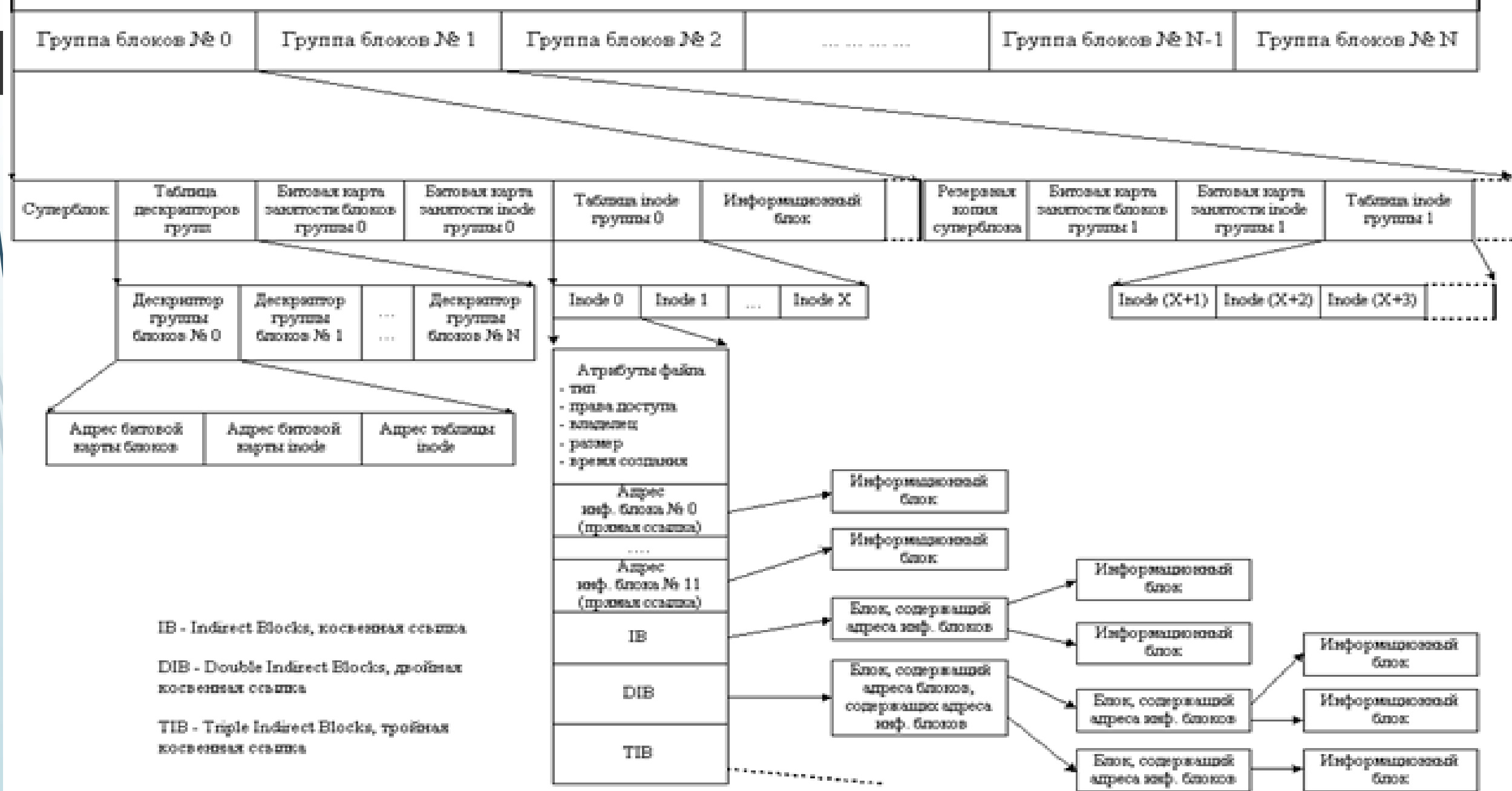
- ОС создаёт несколько резервных копий суперблока для возможности его восстановления в случае повреждения.
- Описание группы блоков, представляет собой массив, содержащий общую информацию обо всех блоках раздела.

ФС ext2

- **Битовая карта блоков** (соответствующий ему блок соответствует какому-либо файлу. Если бит равен 1, то блок занят).
- **Битовая карта индексных дескрипторов**, показывает какие индексные дескрипторы заняты, а какие нет.

ФС ext2

- Все оставшееся место, обозначенное в таблице, как данные, отводится для хранения файлов



ФС ext2

Структурные типы, описывающие основные компоненты файловой системы ext2 - суперблок, дескриптор группы блоков, информационный узел, запись каталога - определены в заголовочном файле **<linux/ext2_fs.h>**.

ФС ext2

Структурные типы, описывающие основные компоненты файловой системы ext2 - суперблок, дескриптор группы блоков, информационный узел, запись каталога - определены в заголовочном файле **<linux/ext2_fs.h>**.

ФС ext2: Структура суперблока

struct ext2_super_block:

_u32 s_inodes_count	- общее число inode-ов в файловой системе;
_u32 s_blocks_count	- общее число блоков в файловой системе;
_u32 s_free_blocks_count	- количество свободных блоков;
_u32 s_free_inodes_count	- количество свободных inode-ов;
_u32 s_first_data_block	- номер первого блока данных (номер блока, в котором находится суперблок);

ФС ext2: Структура суперблока

struct ext2_super_block:

_u32 s_log_block_size

- это значение используется для вычисления размера блока. Размер блока определяется по формуле: $\text{block size} = 1024 \ll \text{s_log_block_size}$;

_u32 s_blocks_per_group

- количество блоков в группе;

_u32 s_inodes_per_group

- количество inode-ов в группе;

_u16 s_magic

- идентификатор файловой системы ext2;

_u16 s_inode_size

- размер информационного узла (inode);

ФС ext2: Структура суперблока

struct ext2_super_block:

_u32 s_first_ino - номер первого не зарезервированного inode;

ФС ext2: Структура дескриптора группы блоков

struct ext2_group_desc

- _u32 bg_block_bitmap** - битовая карта занятости блоков группы;
- _u32 bg_inode_bitmap** - битовая карта занятости inode-ов группы;
- _u32 bg_inode_table** - адрес таблицы inode-ов группы;

ФС ext2: Структура информационного узла

struct ext2_inode:

_u16 i_mode

- тип файла и права доступа к нему. Тип файла определяют биты 12-15 этого поля:
- *0xA000 - символическая ссылка;*
- *0x8000 - обычный файл;*
- *0x6000 - файл блочного устройства;*
- *0x4000 – каталог;*
- *0x2000 - файл символьного устройства;*
- *0x1000 - канал FIFO;*

_u32 i_size

- размер в байтах;

_u32 i_atime

- время последнего доступа к файлу;

ФС ext2: Структура информационного узла

struct ext2_inode:

_u32 i_ctime

- время создания файла;

_u32 i_mtime

- время последней модификации;

_u32 i_blocks

- количество блоков, занимаемых файлом;

**_u32
i_block[EXT2_N_BLOCKS]**

**- адреса информационных блоков
(включая все косвенные ссылки);**

ФС ext2: Структура информационного

узла

Значение EXT2_N_BLOCKS определено в файле <linux/ext2_fs.h>:

```
/*
```

```
* Constants relative to the data blocks
```

```
*/
```

```
#define EXT2_NDIR_BLOCKS 12
```

```
#define EXT2_IND_BLOCK EXT2_NDIR_BLOCKS
```

```
#define EXT2_DIND_BLOCK (EXT2_IND_BLOCK + 1)
```

```
#define EXT2_TIND_BLOCK (EXT2_DIND_BLOCK + 1)
```

```
#define EXT2_N_BLOCKS (EXT2_TIND_BLOCK + 1)
```

ФС ext2: Структура записи каталога

struct ext2_dir_entry_2:

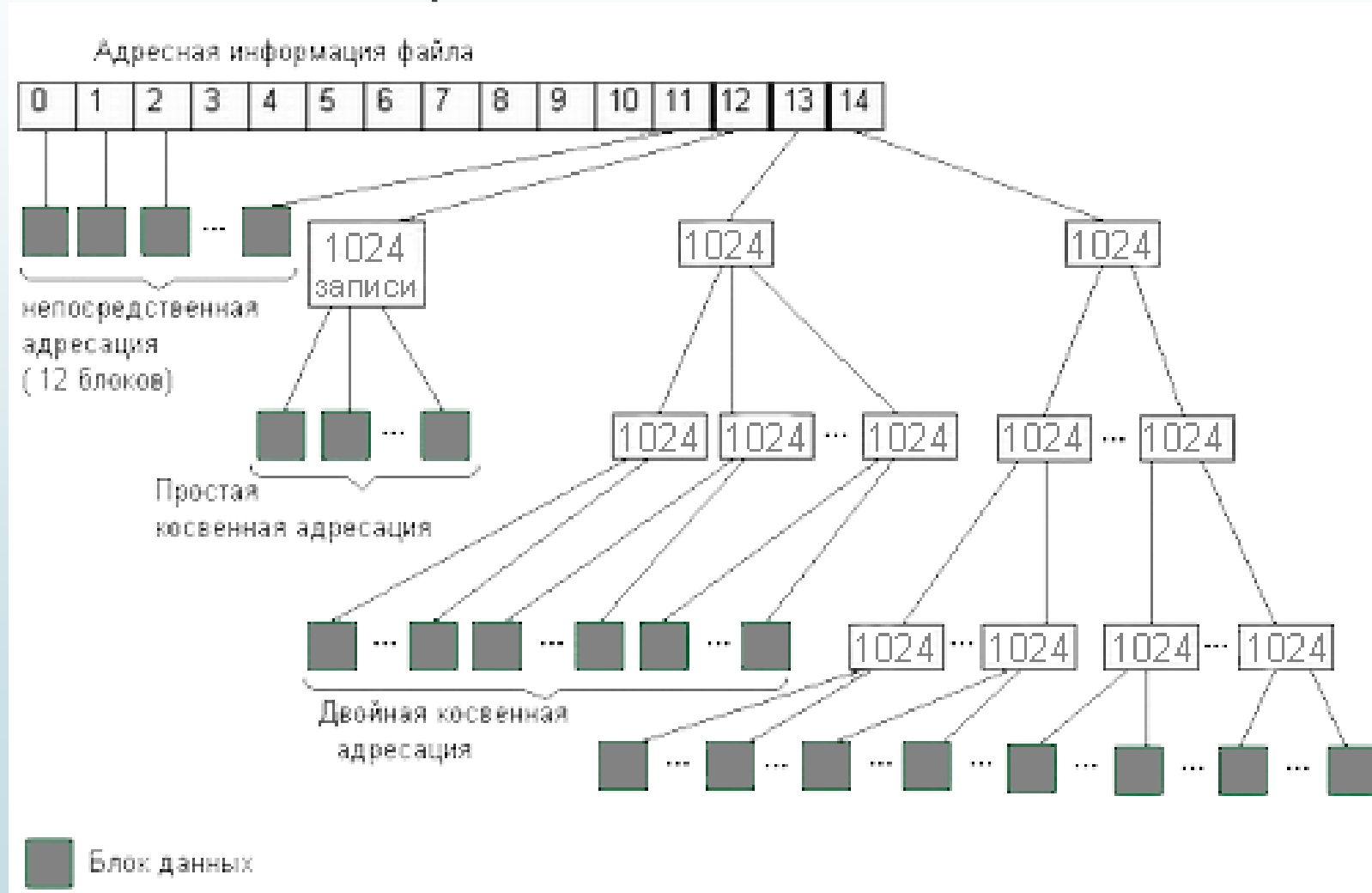
```
#define EXT2_NAME_LEN 255
```

_u32 inode	- номер inode-а файла;
_u16 rec_len	- длина записи каталога;
_u8 name_len	- длина имени файла;
char name[EXT2_NAME_LEN]	- имя файла;

ФС ext2: Система адресации данных

Для хранения адреса файла выделено 15 полей, каждое из которых состоит из 4 байт.

Система
адресации
ФС ext2



ФС ext2: Система адресации данных

■ система адресации, позволяет при максимальном размере блока 4 Кб иметь файлы размера до 2 терабайт или больше.

ФС ext2:

формат Записи каталога:

- - порядковый номер inode файла;
- - длина записи в байтах;
- - имя файла;
- - длина имени файла.

ФС ext2:Несколько первых номеров inode

зарезервированы ФС и содержится в <linux/ext2_fs.h>:

```
/* Special inode numbers*/
```

```
#define EXT2_BAD_INO 1 /* Bad blocks inode */
```

```
#define EXT2_ROOT_IN 2 /* Root inode */
```

```
#define EXT2_ACL_IDX_IN 3 /* ACL inode */
```

```
#define EXT2_ACL_DATA_INO 4 /* ACL inode */
```

```
#define EXT2_BOOT_LOADER_INO 5 //Boot  
loader inode
```

```
#define EXT2_UNDEL_DIR_INO 6 //Undelete  
directory inode
```



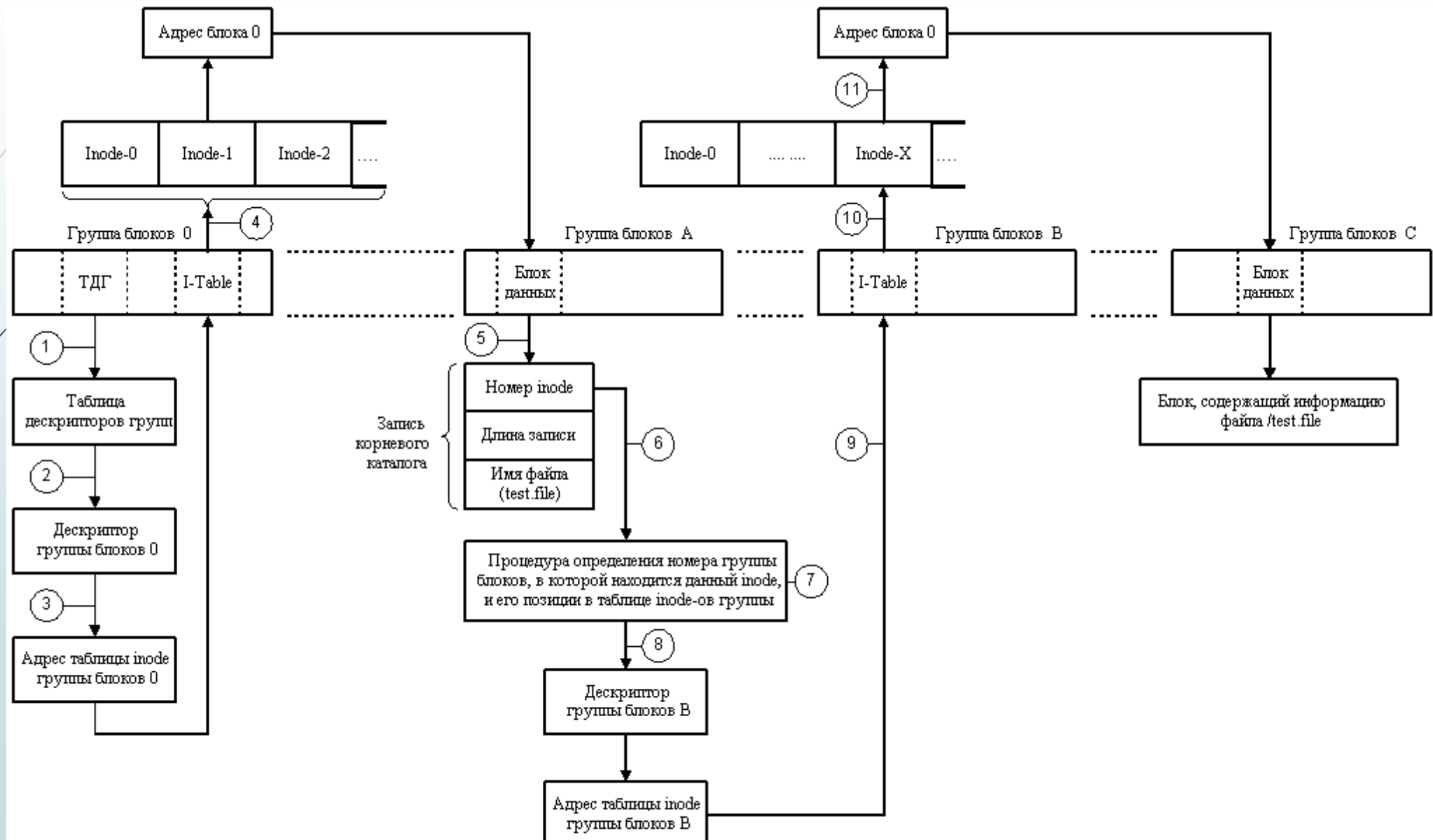

```
#define EXT2_ROOT_IN 2 /* Root inode */
```

Для записи корневого каталога
зарезервирован inode под
номером 2 (root inode). Этот
inode находится в группе блоков
0. Номер первого
незарезервированного inode
хранится в суперблоке.

ФС ext2:

- Определив порядковый номер inode файла, ядро вычисляет номер группы, в которой этот inode расположен, и его позицию в таблице inode группы. Считав из этой позиции inode, ОС получает полную информацию о файле, включая адреса блоков, в которых хранится содержимое файла.
- Номер группы блоков, в которой расположен inode, вычисляется по формуле:
- $group = (inode_num - 1) / inodes_per_group$
- где:
- - group – искомый номер группы блоков;
- - inode_num – порядковый номер inode, определяющего файл;
- - inodes_per_group – число inode в группе (эта информация находится в суперблоке).
- Позиция inode в таблице inode группы определяется по формуле:
- $index = (inode_num - 1) \% inodes_per_group$
- где index – позиция inode в таблице.
- Рассмотрим пример получения содержимого файла test.file, находящегося в корневом каталоге. Для чтения файла /test.file необходимо:
- - в массиве записей корневого каталога найти запись об этом файле;
- - извлечь порядковый номер inode файла, вычислить номер группы, в которой этот inode расположен;
- - из дескриптора данной группы извлечь адрес таблицы inode группы;
- - вычислить позицию inode в этой таблице;
- - считать inode файла;
- - из inode извлечь адреса информационных блоков и осуществить чтение информации, находящейся в этих блоках.

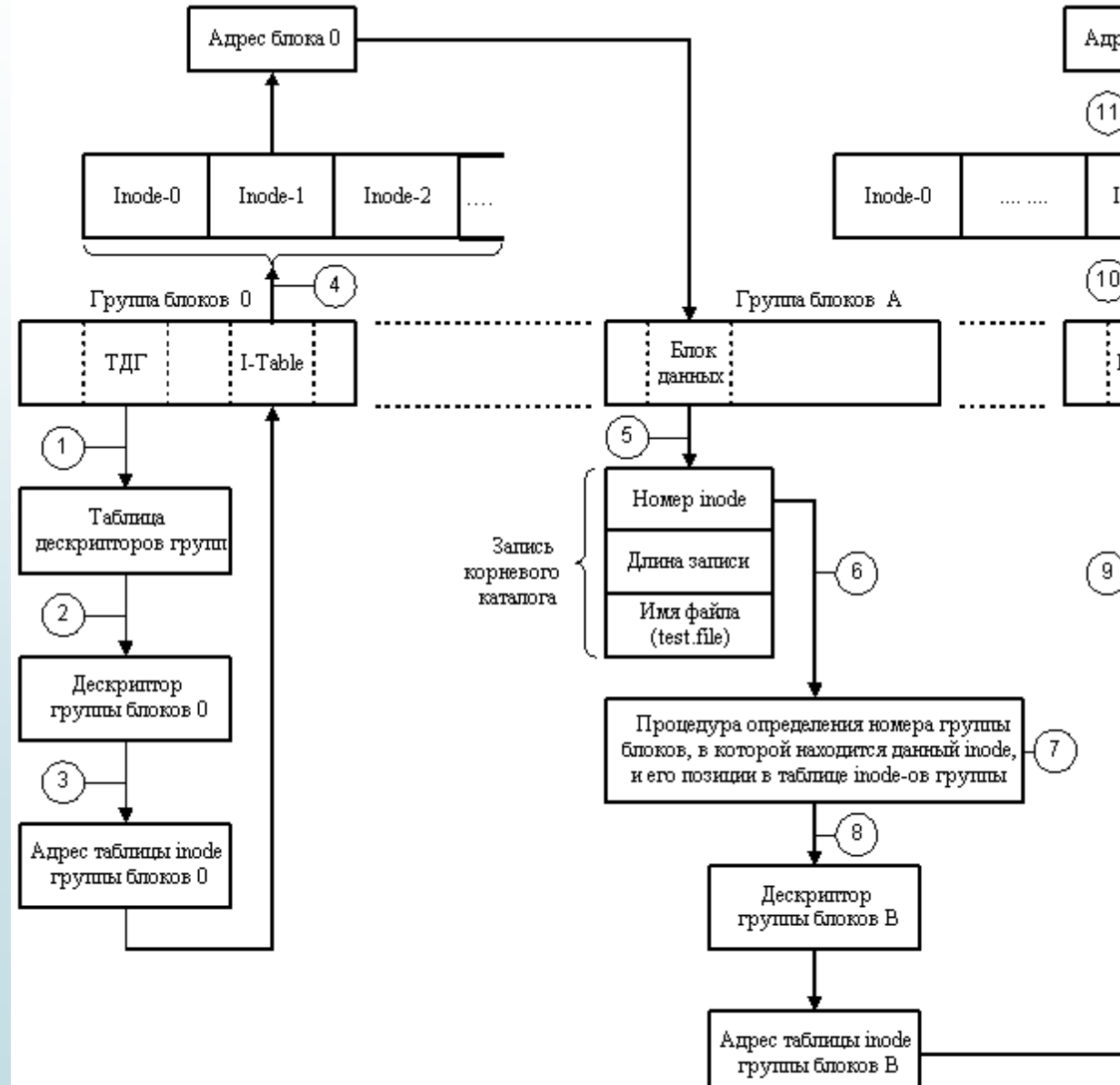
ФС ext2: Порядок выполнения процедуры чтения файла в файловой системе ext2 (на примере файла /test.file)



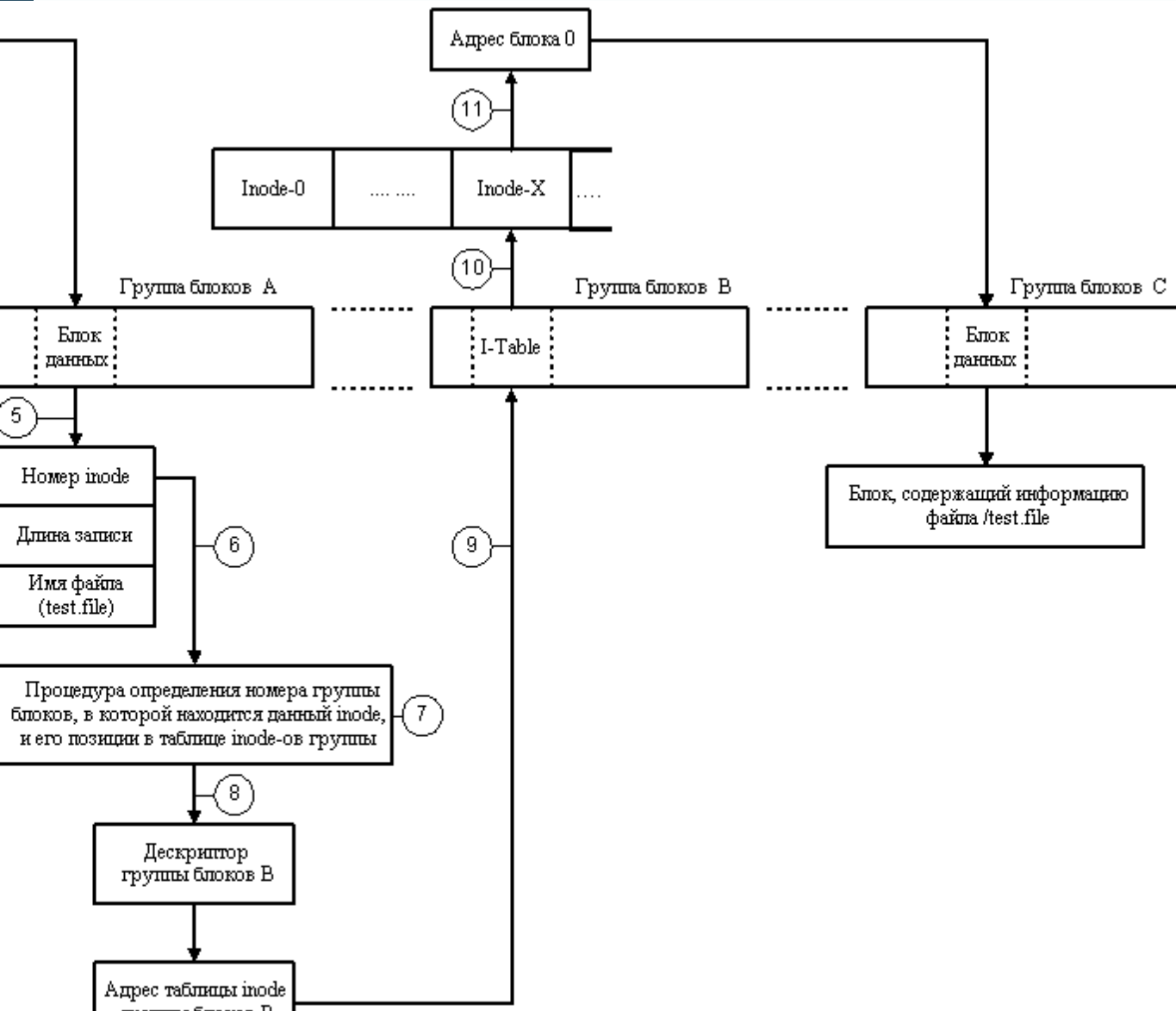
ФС ext2: Порядок выполнения процедуры чтения файла в файловой системе ext2 (на примере файла /test.file)

Этапы 1-6 – чтение корневого каталога:

1. Из группы блоков 0 считывается таблица дескрипторов групп.
2. Из таблицы дескрипторов групп извлекается дескриптор группы блоков 0 и из него считывается адрес таблицы inode группы 0.
3. Из группы блоков 0 считывается таблица inode.
4. Порядковый номер inode корневого каталога фиксирован и равен 2, поэтому из таблицы inode группы 0 считывается второй элемент, который содержит адрес блока с содержимым корневого каталога. Предположим, что этот блок расположен в группе блоков A.
5. Из группы блоков A считывается блок, содержащий записи корневого каталога.
6. Выполняется поиск записи с именем «test.file». Если такая запись найдена, из нее извлекается порядковый номер inode файла «test.file».



ФС ext2: Порядок выполнения процедуры чтения файла в файловой системе ext2 (на примере файла /test.file)



Определив номер inode, можно получить доступ к информационным блокам файла (этапы 7-11):

7. Вычисляется номер группы, в которой находится данный inode, и его позицию в таблице inode группы (предположим, что номер группы равен B, а позиция в таблице – X).

8. Из таблицы дескрипторов групп извлекаем дескриптор группы блоков B, и из него считывается адрес таблицы inode этой группы блоков.

9. Из группы блоков B считывается таблица inode.

10. Из таблицы inode группы блоков B считывается inode, находящийся в позиции X.

11. Из считанного inode извлекаются адреса блока с содержимым файла /test.file и выполняется чтение информации из блока с указанным адресом.