

Файловая система Ext2



Bce структуры, описывающие части файловой системы, взяты из Linux 2.4.0, i386, linux/include/linux/ext2_fs.h (C) 1991, 1992 Linus **Torvalds**

ехt2 пришла на замену более древней ФС ехt . Благодаря своей сокорости она может служить эталоном в тестах производительности файловых систем, но скорость была достигнута благодаря отказу от технологии журналирования, что делает ее неплохой системой для SSD накопителей, т.к. отсуствие журналирования является в данном случае преимуществом.

Содержание [развернуть]

Физическая организация файловой системы 🤌 Править

Файловую систему можно представить в виде следующей структурной схемы:

- Суперблок (Superblock)
- Описание группы блоков (Block Group Descriptors)
- Битовая карта блоков (Blocks Bitmap)
- Битовая карта индексных дескрипторов (Inode Bitmap)
- Таблица индексных дескрипторов (Inode Table)
- Данные

Все пространство раздела диска разбивается на так называемые блоки, имеющие порядковые номера. Блоки имеют фиксированный размер и кратны размеру сектора (1024,2048,4096...). Для уменьшения количества перемещений головки жесткого диска блоки объединяют в группы блоков.

Одним из базовых понятий ext2 является понятие индексного дескриптора (inode, инод, айнод). Это специальная структура, содержащая в себе информацию об физическом расположении файла/директории/ссылки и т.п. и его атрибутах.

Суперблок Править

Суперблок - одно из базовых понятий в ФС ext2. Суперблок содержит в себе общую информацию о файловой системе и находится в 1 килобайте от начала раздела. От цельности суперблока сильно зависит работоспособность ФС, и многие ОС создают его резервные копии на случай повреждения. Блок в виде структуры на языке программирования С представлен ниже:

```
struct ext2 super block {
        s_inodes_count;
                                           Количество индексных дескрипторов в ФС */
         s_blocks_count;
s_r_blocks_count;
                                           Количество блоков в ФС */
  u32
                                           Количество зарезервированых блоков */
  u32
       s_free_blocks_count; /*
  u32
                                           Количество свободных блоков */
                                           Количество свободных индексных дескрипторов */
  u32
         s_free_inodes_count;
                                           Номер данного блока */
log2(размер блока)-10 */
         s_first_data_block;
s_log_block_size;
  u32
 u32
                                     /* log2(размер олока)-10 */
/* log2(размер фрагмента)-10 */
/* # Кол-во блоков в группе */
/* # Кол-во фрагментов в группе */
         s_log_frag_size;
  s32
         s_blocks_per_group;
 u32
         s_frags_per_group;
s_inodes_per_group;
  u32
                                      /* # Koл-во индексных дескрипторов в группе */
 u32
                                          Время последнего монтирования в POSIX времени */
  u32
         s mtime;
 u32
         s_wtime;
                                           Время последней записи в POSIX времени */
         s_mnt_count;
s_max_mnt_count;
  u16
  s16
         s errors;
                                           Код ошибки(см.ниже) */
  u16
 u16
         s_pad;
         s_lastcheck;
  u32
                                           POSIX время последней проверки */
                                           POSIX время между принудительными проверками */
 u32
         s_checkinterval;
                                           ID OC (см. ниже)
  u32
         s creator os;
         s_rev_level;
s_def_resuid;
s_def_resgid;
 u32
                                           Версия */
                                           UserID, могущий использовать зар. блоки */ GroupID, могущий использовать зар. блоки */
 u16
 _u16
                                           Зарезервировано */
         s_reserved[235];
 u32
```

ID OC

0	Linux
1	GNU HURD
2	MASIX
3	FreeBSD
4	Other/Unknown

BGD \nearrow Править

BGD таблица содержит индексные дескрипторы для каждой группы блоков файловой системы и распологается сразу после суперблока. Таким образом, если указано, что размер блока составляет 1 кб, то блок с таблицей начинается в втором блоке. Блоки нумеруются с нуля, и номера блоков не соответствуют их физическому нахождению. Формат дескриптора группы блоков в виде структуры представлен ниже:

```
struct ext2_group_desc
  _u32 bg_block_bitmap;
                                             Номер блока в битовой карте*/
  _u32 bg_inode_bitmap;
                                             Дескриптор блока в битовой карте */
  _u32 bg_inode_stable; /*
_u16 bg_free_blocks_count; /*
_u16 bg_free_inodes_count; /*
_u16 bg_used_dirs_count; /*
                                             Номер блока в таблице дескрипторов */
                                             Свободно блоков */
                                            Свободно индексных дескрипторов */
                                             Количество директорий */
  _u16 bg_pad;
  _u32 bg_reserved[3];
<del>};</del>
```

Индексные дескрипторы Править

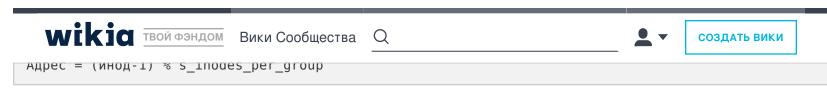
В первой версии файловой системы индексные дескрипторы имели фиксированный размер в 128Кб, начиная с версии 1.0 размер инода указывается в соответствующем поле суперблока. Как и блоки, каждый дескриптор имеет свой номер, отличие лишь в том, что нумерация инодов начинается с единицы. До версии Ext2 1.0 дескрипторы с 1 по 10 были зарезервированы, но начиная с версии 1.0 первый незарезервированный инод отражается с помощью поля в суперблоке. Второй инод имеет самое большое значение, так как он используется для корневого каталога.

Поиск местонахождения дескриптора 🥜 Править

Для определения того, какую группу блоков "представляет" данный индексный дескриптор, можно воспользоваться формулой:

```
Группа блоков = (инод-1) / s_inodes_per_group
```

где s_inodes_per_group соответствующее поле в суперблоке (см. Суперблок).



где % - деление по модулю.

Следующим шагом будет определение адреса блока, содержащего данный индексный дескриптор:

```
Блок = (Адрес * inode_size) / block_size
```

Чтение информации из дескриптора

Править

см. Система адресации данных Структура дескриптора

```
struct ext2 inode {
                  i mode;
                                    /* Тип файла и права доступа (см.ниже) */
           u16
                                    /* UserID */
           u16
                  i uid;
                                    /* Размер в байтах */
           u32
                 i_size;
                                     /* POSIX время последнего обращения к файлу */
           u32
                  i_atime;
                                     /* POSIX время создания */
           u32
                  i_ctime;
                                     /* POSIX время последней модификации */
           u32
                  i mtime;
                                     /* POSIX время удаления */
           u32
                  i_dtime;
                  i_gid;
i_links_count;
           u16
                                     /* GroupID */
                                   /* Кол-во ссылок на дескриптор */
/* Кол-во секторов (не блоки!) */
           u16
                  i blocks;
           u32
                                     /* Флаг (cм.ниже) */
           u32
                  i_flags;
         union {
                  struct {
                              u32 l i reserved1; /* Зарезервировано */
                  } linux1;
                  struct {
                              } hurd1;
                  struct {
                              u32 m_i_reserved1; /* Зарезервировано */
                  } masix1;
          } osd1;
                  i_block[EXT2_N_BLOCKS];/* Указатели на блок */
           u32
                  i_generation; /* Версия файла (для NFS) */i_file_acl; /* Доп. аттрибуты файла */
           u32
                                    /* Доп. аттрибуты файла */
/* Доп. аттрибуты директории */
           _u32
           _u32
                  i dir acl;
                                     /* Адрес фрагмента */
           u32
                  i_faddr;
         union {
                  struct {
                                                       /* Номер фрагмента */
                              u8
                                     l_i_frag;
                                    l_i_fsize;
                                                       /* Размер фрагмента */
                              u8
                                                       /* Зарезервировано */
                              u16
                                     i_pad1;
                                                      /* 16 старших битов UserID */
/* 16 старших битов GroupID */
                              u16
                                     l_i_uid_high;
                                     l_i_gid_high;
                              u16
                  __u32 l_i_reserved2; /* Зарезервировано */
} linux2; /* LINUX */
                  struct {
                                                       /* Номер фрагмента */
                                     h_i_frag;
                                                       /* Размер фрагмента */
                              u8
                                     h_i_fsize;
                                     h_i_mode_high; /* 16 старших битов T&P */
                              u16
                                                      /* 16 старших битов UserID */
/* 16 старших битов GroupID */
                                    h_i_uid_high;
                              u16
                                    h_i_gid_high;
                              u16
                                                       /* UserID или автор */
                             u32
                                     h_i_author;
                  } hurd2; \(\bar{7}\)* GNU H\(\overline{U}\)R\(\overline{D}\) */
                  struct {
                                                       /* Номер фрагмента */
                                     m_i_frag;
                                                       /* Размер фрагмента */
                             น8
                                    m_i_fsize;
                                                        /* Зарезервировано */
                              u16
                                    m pad1;
                                    m_i_reserved2[2]; /* Зарезервировано */
                             u32
                  \} masix2; /* MASI\overline{X} */
         } osd2;
                                              /* Специфичные значения */
```

Тип файла и права доступа

};

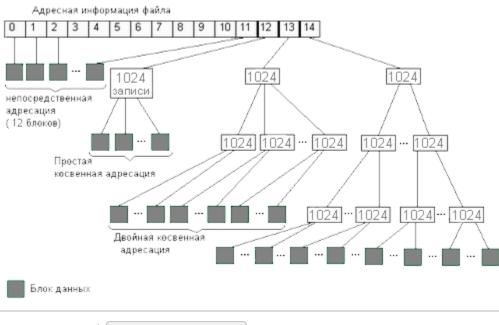
В верхних четырех битах (15 по 12) поля і_том находятся описания типа файла, а в остальных - права доступа.

Тип файла **FIFO** 0x1 Символьное 0x2 устройство 0x4 Директория Блочное устройство 0x6 0x8 Регулярный файл Ссылка 0xA 0xC UNIX-сокет

Для представления прав доступа в UNIX используется 10 битное поле, маску которого можно представить как DRWERWERWE (в данном случае нужно просто "отсечь" последние два бита), где первый бит указывает, является ли файл специальным. Первая RWE тройка указывает права пользователя на чтение, запись, и исполнение файла, вторая - на права группы пользователя, третья - права для всех остальных. К примеру, если получена последовательность битов 0111101101, то данный файл не является директорией, его могут исполнить и читать все "пользователи" данного компьютера, но редактировать позволено только пользователю-создателю.

Система адресации данных 🤌 Править

Для хранения адреса файла выделено 15 полей по 4 байта. Если файл умещается в пределах 12 блоков, то в первых 12 полях адреса перечислены номера соответствующих кластеров, иначе следующее поле используется для косвенной адресации. Возможна ситуация, когда размер файла превышает 1024+12 блоков. Тогда 14 поле используется для двойной косвенной адресации, но если вдруг файл включает в себя более 1049612 блоков, в дело вступает тройная косвенная адресация и 15 блок. Такая организация позволяет при максимальном размере блока в 4Кб оперировать файлами, размер которых превышает 2Тб. Наглядно данная схема показана на рисунке ниже:



Категории: Диски и файловые системы Добавить категорию

