

Файловые системы

- Необходимо постоянное хранение данных
- Используется внешняя память
- Хранение данных организовано в виде файлов; часть операционной системы, управляющая файлами, называется файловой системой
- Файл — абстрактный механизм, именованный набор данных. Представляет собой способ сохранения информации на диске и её считывание. От пользователя скрываются детали: способ и место хранения информации, детали работы дисков и т.п.

Файлы

- Именованние файлов
 - при создании файла ему дается имя, правила именования (длина имени, используемые символы, зависимость от регистра и т.п.) зависят от системы
 - часто имя состоит из двух частей, разделенных точкой: собственно имя и расширения, которое позволяет пользователю быстро определить категорию содержимого файла (тип файла: исполняемый, изображение, музыка, программа на определенном языке и т.п.)
- Доступ к файлам:
 - последовательный (чтение и запись только по порядку от начала к концу, поддерживалась «перемотка»)
 - произвольный (байты файла могут быть прочитаны в произвольном порядке)
- Атрибуты файлов (метаданные): дополнительная информация (кроме имени), описывающая свойства файлов
 - защита (кто и какие операции может выполнять)
 - флаги: только чтение, скрытый, системный, ...
 - дата и время создания
 - время последнего изменения
 - текущий размер
 - максимальный размер
 - ...

Операции с файлами

- Создание/удаление
- Открытие/закрытие (считывание в атрибутов файла, создание связи с процессом – файловый дескриптор/сохранение данных из кэша, удаление связи с процессом)
- Чтение/запись (осуществляется с текущей позиции, которая хранится в файловом дескрипторе)
- Позиционирование в файле (изменение текущей позиции)
- Получение/установка атрибутов файла
- Переименование
- Блокирование
- Операции с файлами реализуются в виде системных вызовов

Каталоги

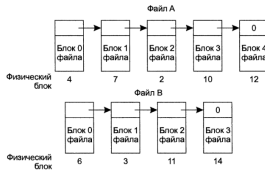
- Каталог (директория, папка) – способ организации и упорядочения файлов. Зачастую сами являются файлами
- Простейшая форма – один каталог, в котором расположены все файлы – корневой каталог
- Двухуровневая система – корневой каталог и каталоги для каждого пользователя
- Иерархическая – каталоги организованы в виде дерева
- Указание расположения файла в каталоге/каталогах:
 - абсолютное (от корневого каталога): `/usr/src/app/test.c`
`c:\Users\PC\Desktop\test.c`
 - относительное (от текущего каталога): `..` – родительский каталог,
`.` – текущий каталог

Операции с каталогами

- Создание/удаление
- Открытие/закрытие (для выполнения последующей операции чтения каталога)
- Чтение следующей записи в каталоге
- Переименование
- Связывание файлов
- Удаление связи файлов
- ...

Реализация файлов

- Главная задача файловой системы — обеспечить соответствие между файлом (который идентифицируется своим именем) и набором секторов на диске, в которых хранятся данные
- Простейшая схема — файл занимает набор смежных секторов
 - легко реализуется, быстро работает
 - диск дефрагментируется, нет возможности расширения файла
 - CD-ROM, DVD
- Связанные списки — первое слово каждого блока является указателем на следующий блок
 - достаточно хранить адрес начального блока
 - медленный произвольный доступ (необходимо прочитать данные с начала файла)



Реализация файлов - FAT

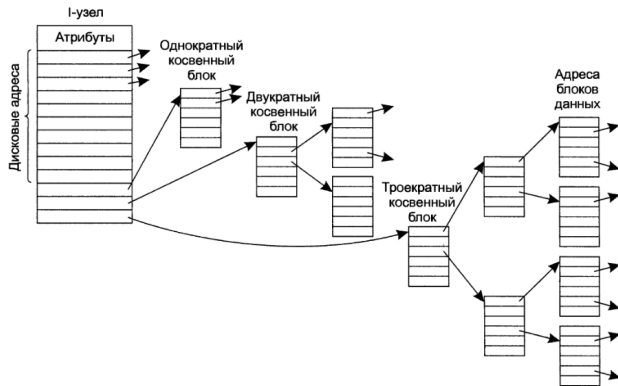
Связные списки с индексацией (указатели на следующие блоки хранятся в отдельной таблице, эта таблица загружается в память) — таблица размещения файлов (File Allocation Table, FAT)

- FAT содержит столько элементов, сколько есть физических секторов на диске (блоков или кластеров в случае объединения нескольких последовательных секторов)
- С каждым файлом связан начальный блок хранения данных на диске, номер которого одновременно служит индексом в таблице FAT для определения следующего блока
- В примере: файл А хранится в блоках 4,7,2,10,12; файл В хранится в блоках 6,3,11,14
- Специальные значения в ячейках FAT могут указывать на окончание файла (последний блок в цепочке) или определять, свободен ли блок
- Необходимо хранить в памяти всю таблицу, размер ее может быть большой

Физический блок		
0		
1		
2	10	
3	11	
4	7	← Файл А начинается здесь
5		
6	3	← Файл В начинается здесь
7	2	
8		
9		
10	12	
11	14	
12	-1	
13		
14	-1	
15		← Неиспользуемый блок

Реализация файлов – индексные узлы (i-узлы)

- С каждым файлом связывается структура данных, называемая индексным узлом
- Эта структура содержит атрибуты и адреса блоков файла
- Для решения проблемы роста файла последний дисковый адрес резервируется не для блока данных, а для адреса косвенного блока



Реализация каталогов

- Каталог состоит из списка элементов, по одному на файл; содержит имя файла, атрибуты, и
 - дисковый адрес начала файла (FAT)
 - номер первого блока (связный список)
 - номер i -узла
- Атрибуты можно хранить в i -узлах

Производительность

- Кэширование — набор блоков диска хранится в памяти. При запросе блока происходит проверка его наличия в кэше. Размер кэша ограничен, используются алгоритмы замещения блоков, аналогичные виртуальной памяти.
- Опережающее чтение блока — последовательное считывание блоков до того, как они понадобятся.
- Снижение времени перемещения блока головок:
 - помещение блоков, к которым высока вероятность обращения близко друг к другу, на один цилиндр
 - помещение i -узлов не в начале диска, а перед каждым файлом
 - помещение i -узлов в середине диска

Примеры файловых систем: CP/M

- CP/M — операционная система, изначально работала на процессорах Intel 8080, Z80
- Один каталог. Файловая запись — набор данных, описывает один файл
- CP/M v1.4 (для дисков 250K, 77 дорожек, 26 секторов по 128 байт на дорожку, 2 резервных дорожки, два 1K блока для файловых записей, двести сорок 1K блоков данных)
 - статус файла (SS, 1 байт, существует, удален, скрыт)
 - имя файла (Fn, 8 байт)
 - расширение (Tn, 3 байта)
 - номер файловой записи (EX, 1 байт, для файлов более 16K)
 - резерв (Sn, 2 байта)
 - количество записей (RC, 1 байт), занятых файлом в данной файловой записи, одна запись — 128 байт, служит для указания размера файла (с округлением до 128 байт).
 - размещение файла (AL, 16 байт, список номеров блоков)

SS	F1	F2	F3	F4	F5	F6	F7	F8	T1	T2	T3	EX	S1	S2	RC	.FILENAME.TYP....	
AL	AL	AL	AL	AL	AL	AL	AL	AL	AL	AL	AL	AL	AL	AL	AL	

Примеры файловых систем: MSDOS

- MSDOS — изначально 16-битный клон CP/M с FAT12
- Иерархическая файловая система (MS-DOS 2.0)
- Смежные сектора диска логически объединяются в кластеры, файл хранится в кластерах
- Обязательно есть корневой каталог, размещается на диске непосредственно после таблицы FAT (FAT12/FAT16)
- Пустые кластера в FAT помечаются 0, последний кластер файла помечается меткой EOC (End Of Clusterchain) (FAT12: $\geq 0x0FF8$, $0x0FFF$; FAT16: $\geq 0xFFFF8$;...), кластер с поврежденным блоком - $0x0FF7$

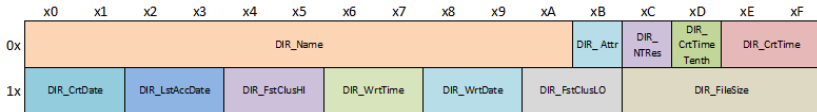
FAT32: файловая запись

- Структура файловой записи FAT32

- DIR_Name — имя файла (8.3)
- DIR_Attr — атрибуты файла (каталог, только чтение, скрытый, системный, архивный, ...)
- DIR_CrtTimeTenth, DIR_CrtTime, DIR_CrtDate, DIR_LstAccDate, DIR_WrtDate — метки времени и даты
- DIR_FstClustHI, DIR_FstClustLO — первый и второй байт номера кластера, где начинается содержимое файла (одновременно индекс в FAT для номера второго кластера)
- DIR_FileSize — размер файла

- Создание каталога

- DIR_FileSize = 0
- Каталог отводится один кластер данных (корневой каталог в FAT12/FAT16 занимает фиксированное количество секторов)
- Создаются две записи: . и .. для текущего и родительского каталога



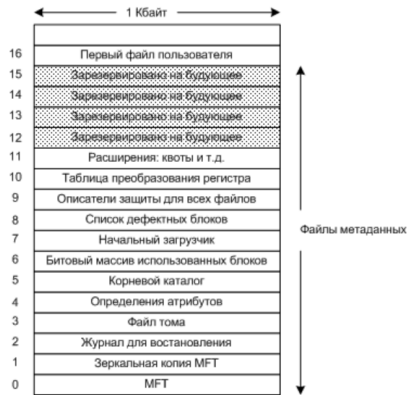
Примеры файловых систем: NTFS

- NTFS (New Technology File System) — файловая система для семейства операционных систем Winndows NT
- Информация о файлах хранится в MFT (Master File Table) — главная файловая таблица
- Поддерживает разграничение доступа к данным для пользователей и групп пользователей
- Позволяет назначать дисковые квоты для пользователей (3.0)
- Файлы хранятся в кластерах (512 байт — 64Kб)



NTFS, структура MFT

- Полная структуризация: всё есть файл; MFT — файл
- MFT представляет собой набор записей по 1Kb
- Каждая запись описывает один файл, но один файл может описываться более чем одной записью
- Первые 16 записей (описывают метафайлы, размещаются в корневом каталоге, начинаются с символа \$) — служебные
 - журнал (\$LogFile): данные для осуществления транзакции
 - файл тома (\$Volume): серийный номер, время создания и т.п. для раздела (тома)
 - определения атрибутов (\$AttrDef): список допустимых атрибутов в записях MFT
 - таблица преобразования регистра (\$UpCase): для преобразования имен файлов/каталогов



NTFS, запись MFT

- Запись в MFT представляет собой совокупность атрибутов, каждый из которых хранится как отдельный поток (streams) байтов
- Атрибут состоит из тела и заголовка; бывают резидентными (значения хранятся в записи MFT) и нерезидентными
- Запись состоит из набора атрибутов
- Типовые атрибуты:
 - \$STANDARD_INFORMATION: стандартная информация о файле (время, права доступа)
 - \$FILE_NAME: полное имя файла
 - \$SECURITY_DESCRIPTOR: дескриптор безопасности и списки прав доступа (ACL)
 - \$DATA: основные данные файла
 - \$BITMAP: карта свободного пространства
 - \$ATTRIBUTE_LIST: расположение дополнительных записей MFT, когда потоки не умещаются в одну запись MFT

NTFS, запись MFT: примеры

Запись для
файла



Запись для
каталога



Ext2 – Extended Filesystem

- При создании файловой системы можно изменить размер блока
- Объединяет блоки диска в группы, каждая группа включает блоки данных и индексные дескрипторы (для ускорения доступа к данным)
- Заранее выделяет блоки данных для файла
- Улучшенные алгоритмы обновления (изменения, удаления) файлов — минимизация сбоев системы
- Поддержка автоматической проверки непротиворечивости при загрузке

Ext2

Раздел разбит на группы блоков, каждый из которых содержит один из элементов информации:

- копия суперблока файловой системы (обновляется при проверке файловой системы)
- копия дескрипторов групп блоков (обновляется при проверке файловой системы)
- битовая карта блоков данных
- битовая карта индексных дескрипторов
- данные файлов

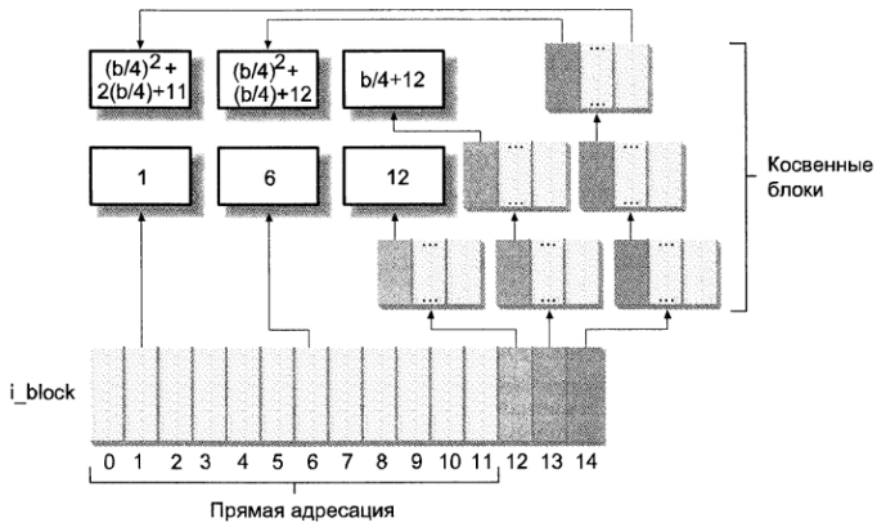


Ext2: таблица индексных дескрипторов

Таблица индексных дескрипторов состоит из ряда блоков, каждый из которых содержит определенное количество дескрипторов (по 128 байт)

- тип файла и права доступа
- ИД владельца
- длина файла
- время обращения, изменения, удаления файла
- количество блоков данных в файле
- флаги файла
- указатели на блоки данных (массив из указателей на логические номера блоков, обычно 15); первые 12 — содержат логические номера, 12,13 и 15 элементы — косвенная адресация; логический номер блока занимает 4 байта
- список доступа к файлу (указывает на блок, содержащий расширенные атрибуты)
- список доступа к каталогу
- адрес фрагмента
- ...

Ext2: i-блоки



Ext2: каталоги

- Каталог — специальный тип файла, содержат имена и номера индексных дескрипторов
- Элементы каталога имеют переменную длину и содержат:
 - номер индексного дескриптора
 - длина элемента каталога
 - длина имени файла
 - тип файла
 - имя файла

	inode	rec_len	file_type	name_len	name
0	21	12	1	2	· \0 \0 \0
12	22	12	2	2	.. \0 \0
24	53	16	5	2	h o m e 1\ 0\ 0 \0
40	67	28	3	2	u s r\ 0
52	0	16	7	1	o l d f i l e \0
68	34	12	4	2	s b i n

Ext3

- Совместимость с Ext2
- Журналирование: каждое изменение выполняется за два шага
 - вначале копии всех блоков сохраняются в журнале
 - затем блоки сохраняются в файловой системе
 - если сбой произошёл до фиксирования в журнале — выдается ошибка
 - если сбой происходит после фиксирования в журнале — повторная запись

Ext4

- 48 битная адресация блоков (файл до 16ТВ, файловая система до 1EB)
- Неограниченное число подкаталогов
- Вместо отображения блоков используется отображение «экстентов» — набора последовательных блоков
- Изменяемый размер i-узла (по-умолчанию — 256 байт, хранение времени с повышенной точностью, доп. атрибутов непосредственно в i-узле)
- Усовершенствования при выделении блоков