**21. Дескрипторы и атрибуты файлов в ОС Unix.**

Дескриптор файла:

* Имя файла (в операционной системе UNIX имя файла не входит в дескриптор)
* Данные, необходимые для указания на размещение файла
* Способ организации файла
* Тип устройства
* Данные для управления доступом к файлу
* Тип файла (текст, объединенный модуль и др.)
* Время создания, последнего изменения и последнего доступа к файлу

В различных ОС в дескриптор файла может входить и другая информация.

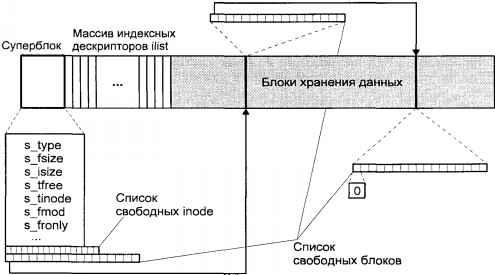
Существует две команды для управления атрибутами файла: lsattr(1) и chattr(1). Команда lsattr выводит список (LiSt) атрибутов, а команда chattr изменяет (CHange) их. Эти атрибуты могут быть установлены только для каталогов и обычных файлов. Доступны следующие атрибуты:

* A («no Access time»): если для файла или каталога установлен этот атрибут, то, всякий раз при обращении к нему для чтения или записи, у него не будет обновляться время последнего доступа. Это может быть полезно, например, для файлов и каталогов, к которым очень часто обращаются для чтения, особенно из-за того, что это единственный параметр в inode, который изменяется при открытии файла только для чтения.
* a («append only»): если для файла установлен этот атрибут, и этот файл открыт для записи, то единственной доступной операцией будет добавление данных к его предыдущему содержимому. Для каталога это означает, что вы сможете только добавить файлы, но не сможете переименовать или удалить ни одного из существующих файлов. Только root может установить или снять этот атрибут.
* d («no dump»): dump (8) - это стандартная утилита UNIX® для резервного копирования. Она делает дамп любой файловой системы, для которой счетчик дампов в файле /etc/fstab (5-е поле) равен 1 (см. главу Глава 8, Файловые системы и точки монтирования). Но если этот атрибут установлен для файла или каталога, то он, в отличие от других, будет будет пропущен при снятии дампа. Обратите внимание, что при установке его для каталогов, это также распространяется на все их подкаталоги и файлы.
* i («immutable»): файл или каталог с установленным этим атрибутом вообще не может быть изменен: он не может быть переименован, на него не может быть создана ссылка[24] и он не может быть удален. Только root может установить или снять этот атрибут. Обратите внимание, что это также предотвращает изменение времени последнего доступа, поэтому вам нет необходимости устанавливать атрибут A, если установлен i.
* s («secure deletion»): когда удаляется файл или каталог с этим атрибутом, блоки, которые он занимал на диске перезаписываются нулями.
* S («Synchronous mode»): если для файла или каталога установлен этот атрибут, все его изменения синхронизируются и немедленно записываются на диск.

**22. Базовая файловая система ОС Unix System V.**

ФС System V состоит из:

* Блока загрузки
* Суперблока
* Списка индексов
* Информационных блоков



**Суперблок:**

* Тип файловой системы (s\_type)
* Размер файловой системы в логических блоках, включая сам суперблок, ilist и блоки хранения данных (s\_fsize)
* Размер массива индексных дескрипторов (s\_isize)
* Число свободных блоков, доступных для размещения (s\_ftree)
* Число свободных inode, доступных для размещения (s\_tinode)
* Размер физического блока (512, 1024, 2048)
* Список номеров свободных inode
* Список адресов свободных блоков

**Структура дискового inode**

di\_mode - Тип файла, дополнительные атрибуты выполнения и права доступа

di\_nlinks - Число ссылок на файл, т. е. количество имен, которые имеет файл в файловой системе

di\_uid, di\_gid - Идентификаторы владельца-пользователя и владельца-группы.

di\_size - Размер файла в байтах. Для специальных файлов это поле содержит старший и младший номер устройства.

di\_atime - Время последнего доступа к файлу.

di\_mtime - Время последней модификации.

di\_ctime - Время последней модификации inode (кроме модификации полей di\_atime, di\_mtime).

di\_addr[13] - Массив адресов дисковых блоков хранения данных.

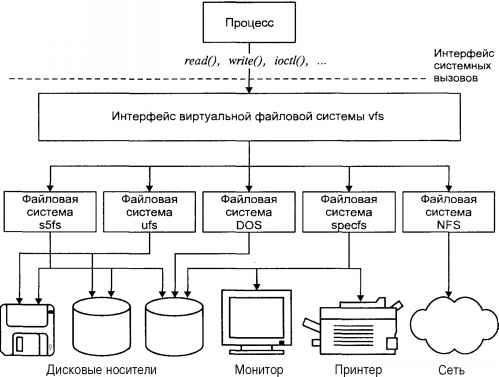
Поле di\_mode хранит несколько атрибутов файла: тип файла (IFREG - для обычных файлов, IFDIR - для каталогов, IFBLK или IFCHR для специальных файлов блочных и символьных устройств соответственно); права доступа к файлу для трех классов пользователей и дополнительные атрибуты выполнения (SUID, SGID и sticky bit). Заметим, что в индексном дескрипторе отсутствует информация о времени создания файла. Вместо этого inode хранит три значения времени: последнего доступа (di\_atime), время последней модификации содержимого файла (di\_mtime) и время последней модификации метаданных файла (di\_ctime). В последнем случае не учитываются модификации полей di\_atime и di\_mtinie. Таким образом, di\_ctime изменяется, когда изменяется размер файла, владелец, группа или число связей.

Блоки хранения данных. Данные обычных файлов и каталогов хранятся в блоках. Обработка файла осуществляется через inode, содержащего ссылки на блоки данных. Блоки хранения данных занимают большую часть дискового раздела, и их число определяет максимальный суммарный объем файлов данной файловой системы. Размер блока кратен 512 байтам, например, файловая система S51K SCO UNIX использует размер блока в 1 Кбайт (отсюда и название).

**23. Архитектура виртуальной файловой системы**

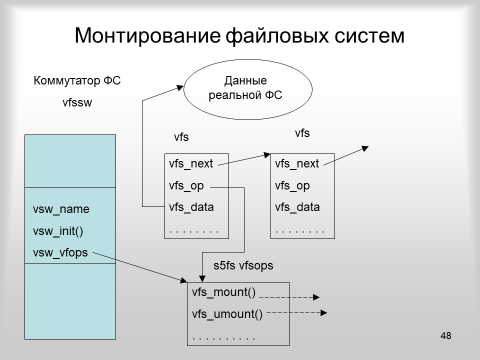
Виртуальная файловая система — уровень абстракции поверх конкретной реализации файловой системы. Целью VFS является обеспечение единообразного доступа клиентских приложений к различным типам файловых систем. VFS может быть использована, например, для прозрачного доступа к локальным и сетевым устройствам хранения данных без использования специального клиентского приложения (независимо от типа файловой системы). VFS определяет интерфейс между ядром и конкретной файловой системой, таким образом, можно легко добавлять поддержку новых типов файловых систем, внося изменения только в ядро операционной системы.

Виртуальная файловая система определяет интерфейс взаимодействия между ядром ОС и конкретной файловой системой, что упрощает поддержку новых типов файловых систем, просто дополняя интерфейс взаимодействия. Определения в интерфейсе могут приводить к изменению совместимости от релиза к релизу, что требует поддержки файловой системой перекомпиляции и возможной модификации перед компиляцией; или же, поставщик ОС может вносить изменения только с обратной совместимостью, так что поддержка конкретной файловой системы, построенной для данного релиза ОС, будет работать с будущими версиями. Иногда виртуальными файловыми системами называют псевдо-файловые системы, которые не предназначены для хранения данных. Примером такой системы является procfs(виртуальная файловая система, используемая в UNIX-подобных операционных системах).



Операция VFS\_MOUNT выполняет традиционное для UNIX монтирование файловой системы на указанный каталог уже смонтированной файловой системы для образования общего дерева, а операция VFS\_UNMOUNT отменяет монтирование. Операция VFS\_ROOT используется при разборе полного имени файла, когда встречается дескриптор vnode, который связан со смонтированной на него файловой системой. Операция VFS\_ROOT помогает найти vnode, который является корнем смонтированной файловой системы. Операция VFS\_STATVFS позволяет получить независимую от типа файловой системы информацию о размере блока файловой системы, о количестве блоков и количестве свободных блоков в единицах этого размера, о максимальной длине имени файла и т.п. Операция VFS\_SYNC выталкивает содержимое буферов диска из оперативной памяти на диск. Операция VFS\_MOUNTROOT позволяет смонтировать корневую файловую систему, то есть систему, содержащую корневой каталог / общего дерева. Для указания того, какая файловая система будет монтироваться как корневая, в UNIX System V Release 4 используется переменная rootfstype, содержащая символьное имя корневой файловой системы, например "ufs".

**24. Монтирование и доступ к файловым системам.**



**Структура виртуальной файловой системы (VFS)**

* Struct vfs \*vfs\_next – Следующая файловая система в списке
* Struct vfsops \*vfs\_op – Операции файловой системы
* Struct vnode \*vfs\_vnjdecovered – vnode, перекрываемой файловой системы
* Int vfs\_flag – флаги доступа
* Int vfs\_bsize – Размер блока файловой системы
* Caddr\_t vfs\_data – Указатель на специфические данные, относящиеся к реальной файловой системы.

**Операции файловой системы**

* VFS\_MOUNT монтирование файловой системы
* VFS\_UNMOUNT размонтирование файловой системы
* VFS\_ROOT получение vnode для корня файловой системы
* VFS\_STATVFS получение статистики файловой системы
* VFS\_SYNC выталкивание буферов файловой системы на диск
* VFS\_VGET получение vnode по номеру дескриптора файла
* VFS\_MOUNTROOT монтирование корневой файловой системы