Операционные системы

Система Linux поддерживает 7 типов файлов.

В UNIX системах все является файлов. В таких системах имя файла не является идентификатором. Изначально система могла поддерживать файлы большого размера

Файл – некоторая поименованная совокупность данных, содержимое файлов никак не интерпретируется, хранится во вторичной памяти (жесткие диски, файлы).

Файлы имеют имя, с целью организации к ним доступа. Безымянная совокупность данных недоступно.

Файловая подсистема

Файловая система имеет иерархическую структуру. Файлы находятся на диске, диск – это физический уровень.

Inode – это структура ядра. Для того, чтобы получить доступ к файлу необходимо получить доступ к идентификатору. Файл идентифицируется в системе номером inode.

|  |
| --- |
| Символьный уровень |
| Базовый уровень |
| Проверка прав доступа |
| Логический уровень |
| Физический уровень |

уровни файловой подсистемы

Важнейшим моментов является защита информации в системе и для этого отдельное внимание уделяется к защите прав доступа. Они отдельно устанавливаются для пользователя.

Операционная система поддерживает 2 типа файлов: текстово ориентированные и блочно ориентированные.

Блочные файлы – хранят информацию блоками равного размера. Из-за такой организации файлов у программиста появляется возможность получить доступ к конкретному блоку. Блочные файлы поддаются к сортировке.

Файловая система – это часть операционной системы, которая отвечает за возможность длительного хранения информации и доступа к ней. Доступ подразумевает создание, чтение, запись, именование, переименование, установка изменения прав доступа, удаление и т.п.

Каждая индивидуальная идентифицируемая единица информации называется файлом.

Файловая система

1. определяет формат сохраняемой информации и способ ее физического хранения
2. связывает физический носитель информации и системные вызовы (API) для доступа к файлам.

Файл – любая поименованная совокупность данных или информации, хранящиеся во вторичной памяти.

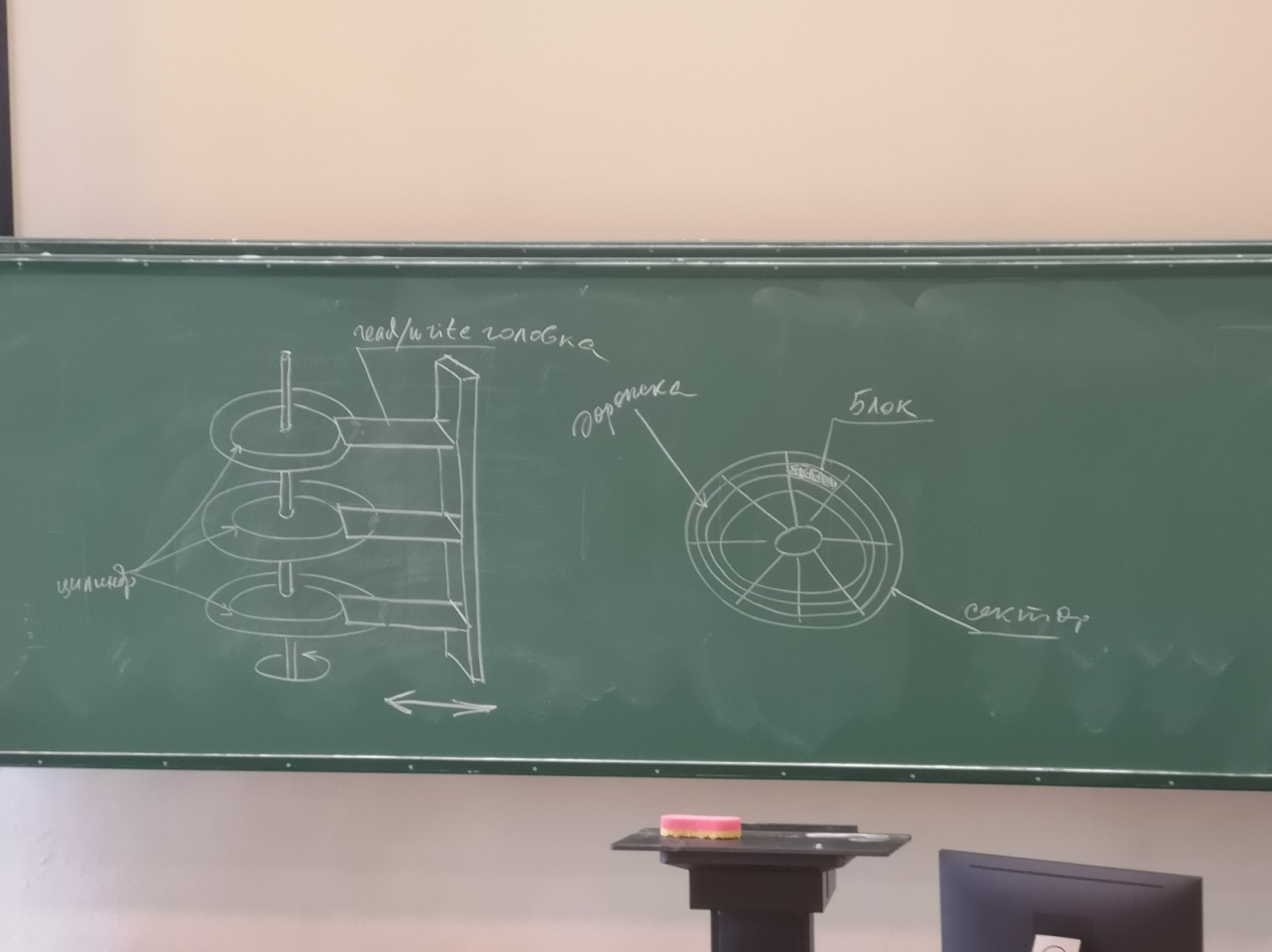
Задачи файловой системы:

* отображение логической модели файлов
* обеспечение длительного хранения файлов
* доступ к файлам
* защита от несанкционированного доступа
* обеспечение совместного использования файлов

Логическое адресное пространство файла аналогично логическому адресному пространству процесса. Начинается с нулевого адреса и представляет непрерывную последовательность адресов. Логический уровень позволяет обеспечить доступ к данным хранящимся в файле в формате, который отличается от формата их физического хранения. В результате запроса на запись или чтение данных из файла, этот запрос преобразуется в запрос на физическую последовательность байтов, которая хранится на физическом носителе. Структурами данных записываемых в файл управляет пользователь. Система данные записанные в файл никак не интерпритирует.

Например в бинарные файлы мы можем записывать объявленную в программе структуру и только в этой программе определены значения полей данных. В текстовых файлах информация хранится в виде символов и любой такой файл может быть отредактирован в текстовом редакторе.

Физический уровень

Рассмотрим организацию жесткого диска. Жесткий диск состоит из одной или более пластин и обычно рисуется следующим образом: 

Блок – наименьшая адресуемая порция данных, к которой можно обратиться на жестком диске. Файл может храниться на диске в виде непрерывной области адресного пространства диска. Существует несвязное распределение адресного пространства диска – для хранения файлов используются блоки, находящиеся в памяти не последовательно.

Ядро реализует некоторый слой абстракции над своим низкоуровневым интерфейсом.

VFS – virtual file system.

Для файловой системы UNIX, Linux – характерна поддержка VFS. Благодаря наличию интерфейса VFS/vnode (Unix), Linux отказался от vnode. Вирутальной файловой системой называется набор структур, при этом базовыми являются 4 структуры.

* Супер блок struct superblock
* Индексный узел struct inode
* Элемент каталога struct dentry
* Файл struct file

/”root

* /bin
* /boot
* /ets
* /usr
* /var
* /dev
* /home
* …

Каждая файловая система определяет как обратиться к файлу, как прочитать или обратиться к файлу и т.п. В каждой отдельной файловой системе код скрывает от пользователя физические особенности работы с файлом. Однако общим для различных файловых систем явлется то, что они поддреживают такие понятия как файлы, каталоги. Поддреживают такие действия как создание файла, удаление файла, переименование файла, чтение файла и т.п. Другими словами несмотря на огромное разнообразие файловой системы все они выполняют одинаковые действия. В результате в юникс и линукс сформирован абстрактный интерфейс VFS, который является ожидаемым конкретными файловыми системами. В результате соответствующие действия конкретной файловой системе отражаются на файле.1

