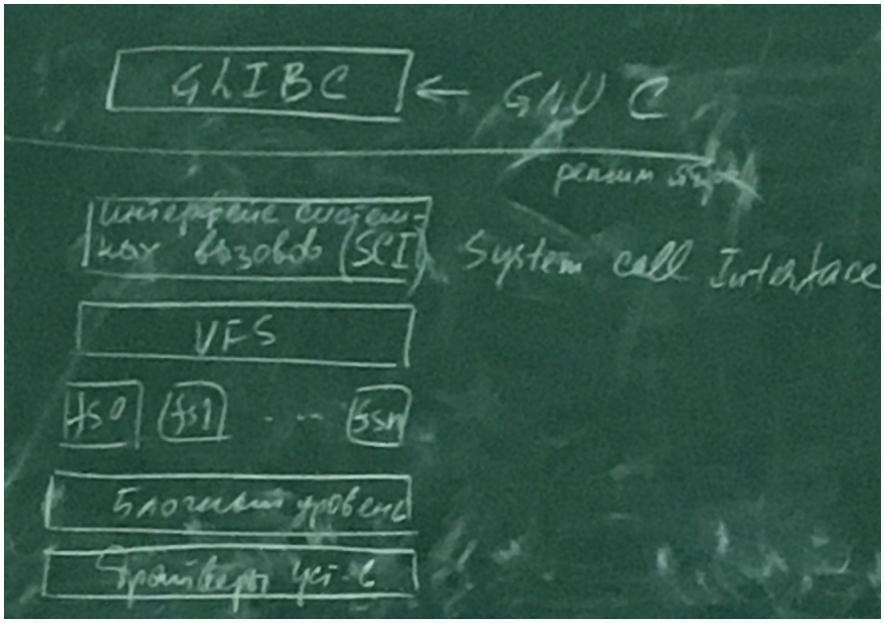
Сб, 23 марта

Рязанова лекция.
Файловая система UNIX/LINUX.

Всегда подчеркивают, что это многофайловая система. Для того, чтобы иметь возможность поддерживать такое большое количество фс в ос реализован специальный интерфейс. В UNIX он называется VFS/vnode, в LINUX просто VFS. LINUX не декларируют vnode, виртуального узла в нем нет.



VFS представляет уровень абстракции, отделяющий POSIX API от подробностей работы конкретной файловой системы. Ключевой момент - системные вызовы, такие как open, read write работают одинаково, независимо от того, какая фс располагается ниже. VFS представляет собой общую файловую модель, которую наследуют низлежащие файловые системы, фактически реализующие действия различных POSIX API.

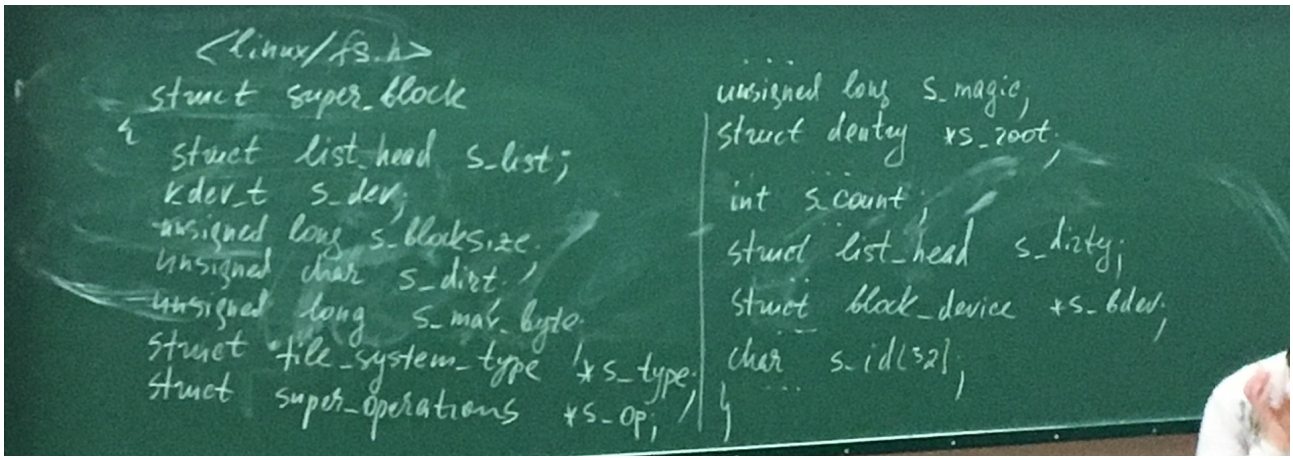
В основе работы VFS лежат 4 базовые структуры - superblock, inode, запись каталога (dentry), файл.

Superblock.

Суперблок содержит высокоуровневые метаданные о файловой системе. Задание - посмотреть в толковом словаре значение приставки мета. **Суперблок - это структура, которая содержит информацию, необходимую для монтирования и управления файловой системой. Суперблок это структура, которая находится на диске, каждая файловая система имеет один суперблок, но на диске суперблок находится в нескольких экземплярах для надежности хранения данных, т. к. Суперблок есть ключевая структура файловой системы.**

Очевидно, суммарное число блоков, свободное число блоков, корневой inode и тд.
Суперблок существует не только на диске. Он предоставляет системе информацию и файловой системе для ядра, чтобы ядро могло работать с фс. Суперблок в памяти предоставляет информацию, нужную для работы с монтированной фс.

// там где в описании интерфейса суперблока три точки упущено поле
// struct list_head s_inodes; - это все inode'ы.
// struct dentry_operations *s_d_op - определяет dietary operations для директории



Поскольку линукс поддерживает одновременно большое количество фс, значит в системе будет соответствующее количество суперблоков, и они хранятся в списке struct list_head, т. е. Это список суперблоков.

s_dev - указывает устройство, на котором находится файловая система.

blocksize - размер блока в байтах.

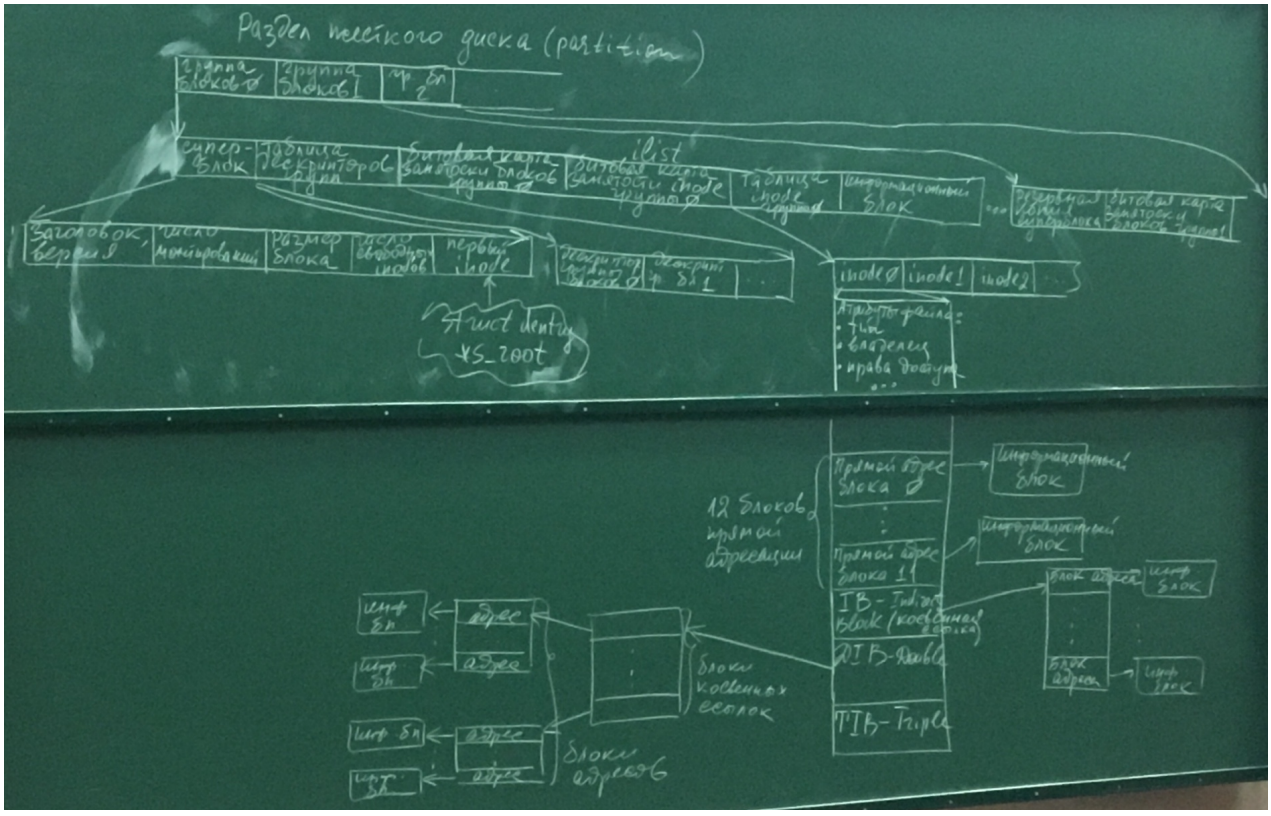
• s_dirt - флаг, показывающий, что суперблок был изменен
Суперблок описывает смонтированные фс, но каждая фс описывается структурой struct file_system_type. Это тип файловой системы, соответственно тип в системе может быть один. Мы можем смонтировать несколько фс ext, но тип у этих фс будет один.

• struct super_operations - действия, определенные на суперблоках.

• s_magic - магический номер файловой системы

- s_root - точка монтирования файловой системы или каталог монтирования фс.
- s_count - счетчик ссылок на суперблок. (it appears to be the mounting count???)
- list_head s_dirty - список измененных индексов.
- Struct block_device *s_bdev - драйвер соответствующего блочного устройства
- s_id - строка имени.

Раздел жесткого диска.



Файлы в системе именуются номерами айнодов. Чтобы разделять адресное пространство диска, надо иметь информацию, которую эффективнее всего хранить в битовой карте - о занятых и свободных айнодах.

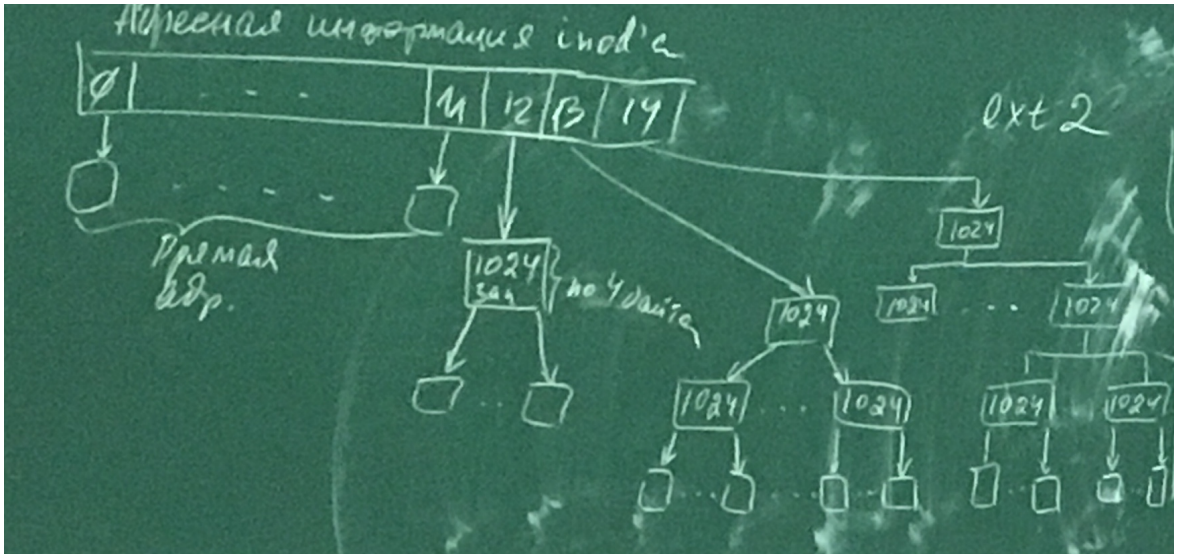
В соответствии с описанной структурой суперблок, что в суперблоке хранится информация, а в частности struct dentry_root - это указатель на корневой каталог. **В системе LINUX все файл. ВСЕ ФАЙЛЫ В СИСТЕМЕ ИМЕЮТ АЙНОДЫ.**

Struct dentry (directory entry) создается на лету, создается на основе информации, которая хранится в айнодах. Файлы ищутся по их полным именам, то есть сначала рассматривается путь к файлу, то есть все составляющие пути к файлу должны быть просмотрены. Dentry кэшируются.

Дисковый inode хранит информацию о физическом файле и позволяет получить доступ к информации, записанной в файле. LINUX поддерживает файлы очень больших размеров.

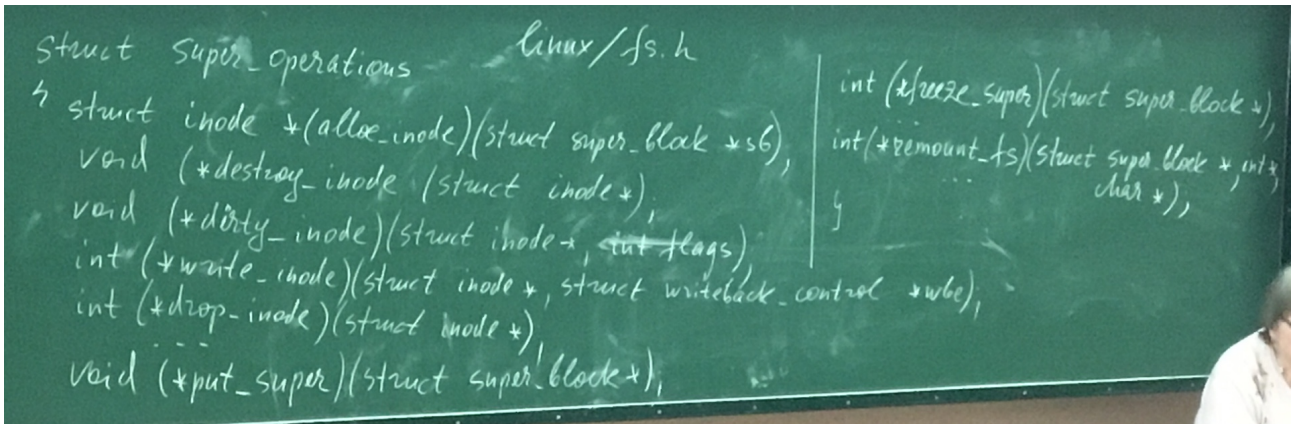
Блок косвенной адресации ссылается на блок в котором хранятся адреса блоков по тому же принципу.

Double Indirect Block - двойная косвенная адресация, дальше тройная косвенная.



— —

Struct super_operations - операции, определенные на суперблоке.



dirty_inode - функция вызывается, когда в файл вносятся изменения.

Журналируемые фс, например, EXT3 используют эту функцию для обновления журнала.

write_inode - записывает inode на диск и одновременно помечает его как грязный
drop_inode - сбрасывает inode.

Функция drop_inode вызывается, когда исчезает последняя ссылка на индекс, и vfs ее просто удаляет.

Когда файловая система нуждается в выполнении операции над суперблоком, то она следует за указателем на желаемый метод объекта суперблока.

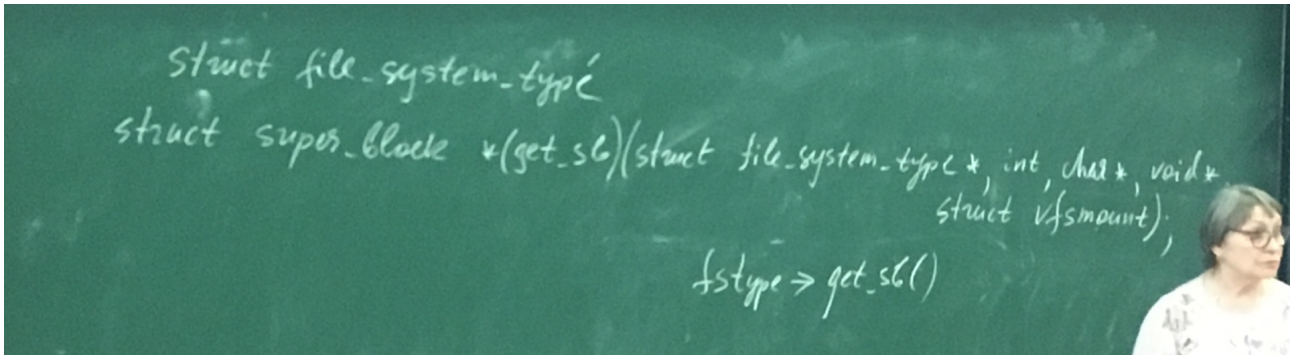
Например
sb -> s_op -> put_super(sb);

Код для создания управления и ликвидации объектов суперблока находится в файле fs/super.c

Объект суперблок создается и инициализируется функцией alloc_super(). Эта функция выделяет и инициализирует новую структуру суперблок, и возвращает указатель на новый суперблок, и null если выделение супероблока завершилось аварийно.

Суперблок описывает информацию, которая нужна системе чтобы управлять смонтированной фс. Любая фс описывается структурой file_system_type. В этой структуре есть поле

Struct super_block *(get_sb)



Эта функция вызывается вовремя монтирования файловой системы. Ядро называет эту функцию <нижняя строчка фотки>

И эта функция инициализирует поля данных и устанавливает суперблок