По функциональности, ядра почти не различаются: поддерживают многопроцессность, многопоточность, виртуальную память. Управление памятью – страницами по запросам. Осноное отличие ОСей – файловые системы. ФС – иерархические.

Уровни:

|  |  |  |
| --- | --- | --- |
| 1 | символьный | верхний, для именования файлов |
| 2 | базовый | позволяет обратиться к inode |
| 3 | проверки прав доступа | возможен только на уровне inode |
| 4 | логический | объясняет, как получить доступ к отдельному блоку/байту |
| 5 | физический | уровень устройства, подсистема ввода/вывода |

Файл описывается inode (index node) – большая структура. Их несколько копий: в ядре (содержит все поля, для оперирования файлом), дисковый (??).

inode содержит информацию о физическом расположении.

Файлы бывают:  
- блокориентированные;  
- байториентированные (символьные, текстовые - на уровне пользователя).

Устойства:  
- символьные;  
- блочные (ЖД).

Декларация: в UNIX всё файл. Следствие этого: работа с файлами и устройствами одними и теми же системными вызовами позволила сократить количество системных вызовов.

В Unix можно установить любое количество файловых систем. Эта идея получила название vnode/vfs. Этот интерфейс имеет объектно ориентированный подход. На основе 1 класса можно породить 1 или более новых классов наследников. Класс наследник может стать базовым для его наследников.

Архитектура vfs:

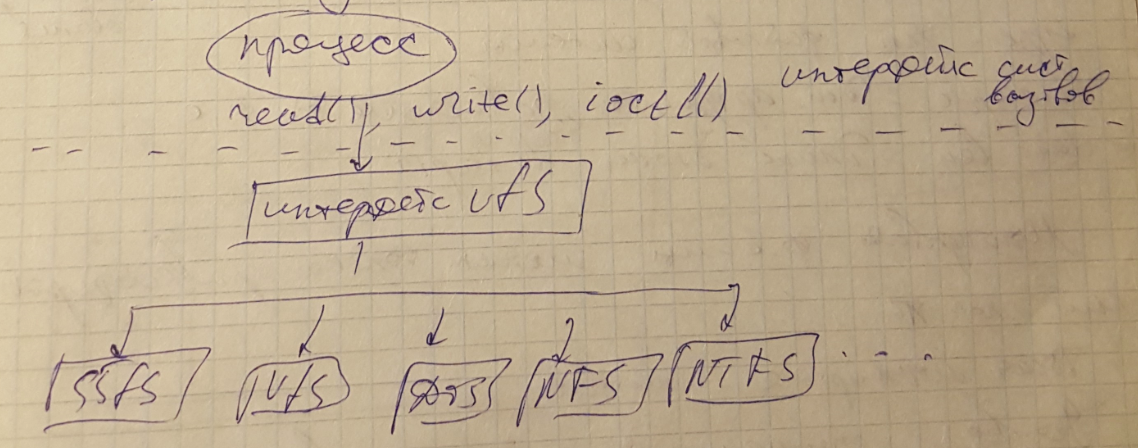


Рисунок 1.

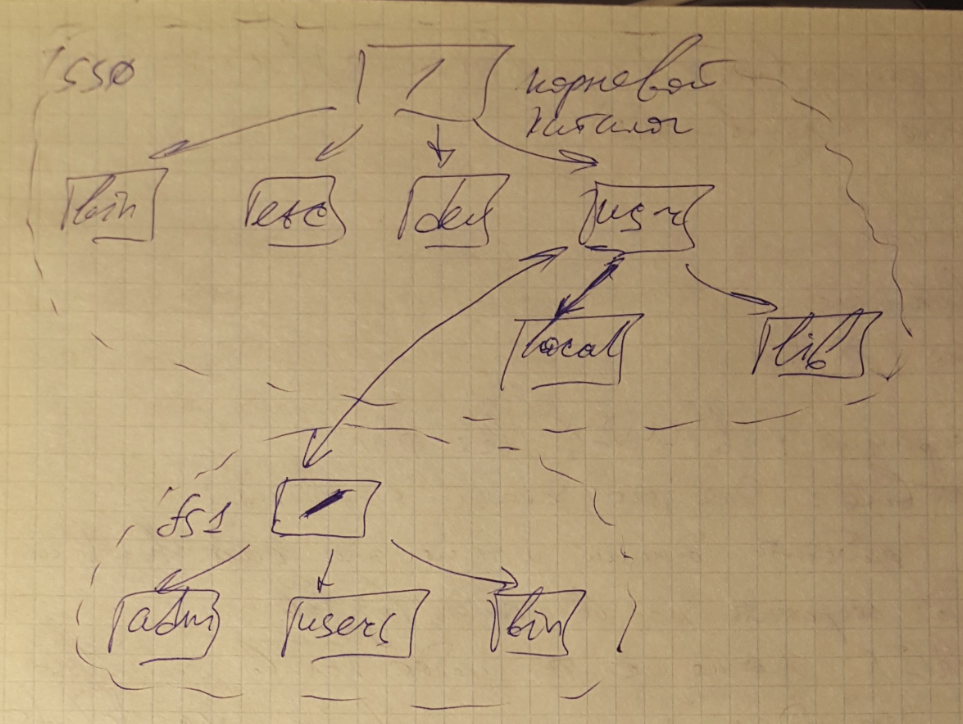


Рисунок 2.

/usr/lib – абсолютное имя

./lib – относительное имя

По Вахалии (стр 366):  
mount (spec, dir, flags, type, dataptr, datalen);  
spec – имя файла – устройства, представляющего ФС;  
dir – полное имя каталога;  
type – тип файловой системы;  
dataptr – дополнительные аргументы, зависящие от ФС;  
datalen – длинна блока dataptr.

Монтировать ФС может только привилегированный пользователь. Точка монтирования – каталог. Устройство – специальный файл, с помощью которого система получает доступ к физическому устройству.

В командной строке функция mount выглядит так:   
# mount устройство точка\_монтирования

Даже ФС в разделе ЖД нужно монтировать. Отметим, что при инсталляции Linux и создании на ЖД раздела Linux, автоматически конфигурируется на монтировании основных ФС-м при каждом запуске.

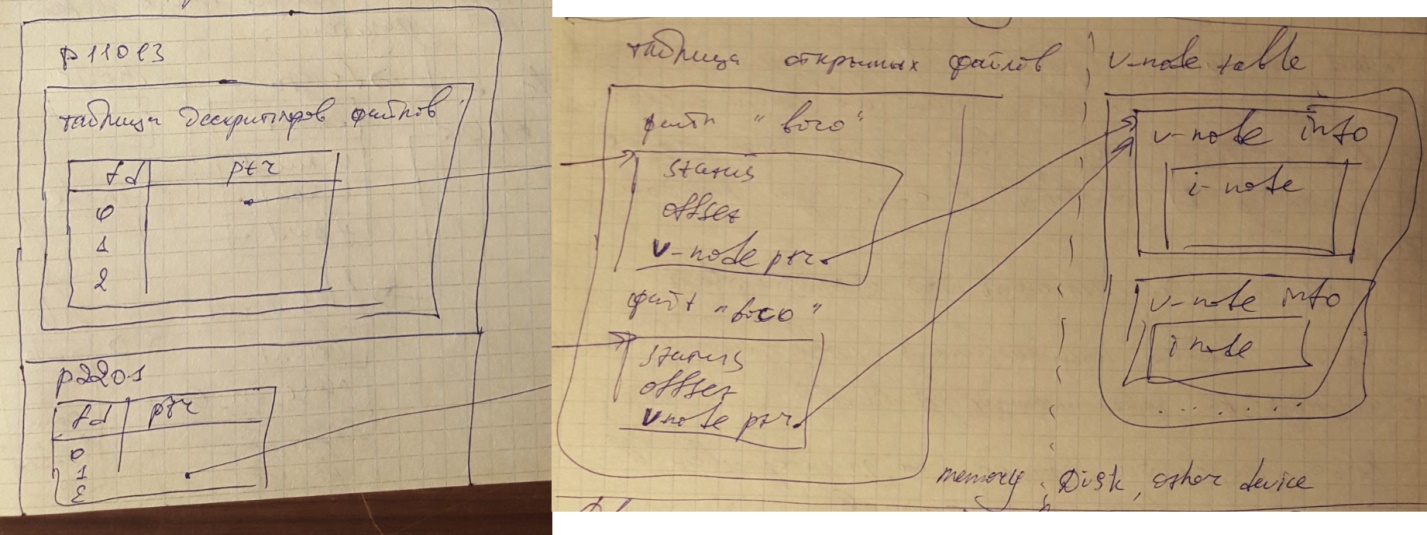
umount – команда, чтобы отмонтировать.

Файл – любая поименованная совокупность данных, размещенная на запоминающих устройствах.

Когда запускается процесс, он заинтересован в чтении/записи файлов.

Библиотечные функции написаны для минимизации системных вызовов, т.к. системные вызовы переводят ОС в режим ядра.

Любой запущенные процесс имеет дескриптор в таблице процессов. Все открытые файлы описаны в таблице открытых файлов, которая одна на систему.

  
Рисунок 3.

Два процесса решили одновременно открыть один и тот же файл. Будет создано 2 дескриптора, описывающие открытый файл.

В vnode ссылка на inode.

task\_struct – описывает процесс Linux. В ней есть указатели:

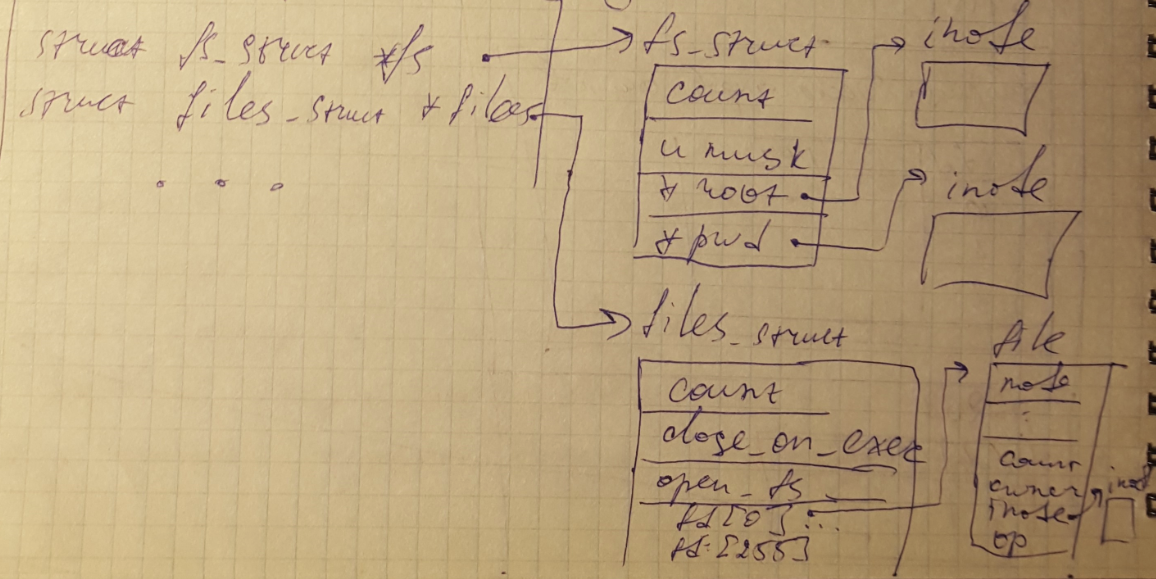


Рисунок 4.

