



Российский разработчик и поставщик  
решений на основе программного обеспечения  
с открытым исходным кодом



# Работа с файлами и каталогами в РЕД ОС

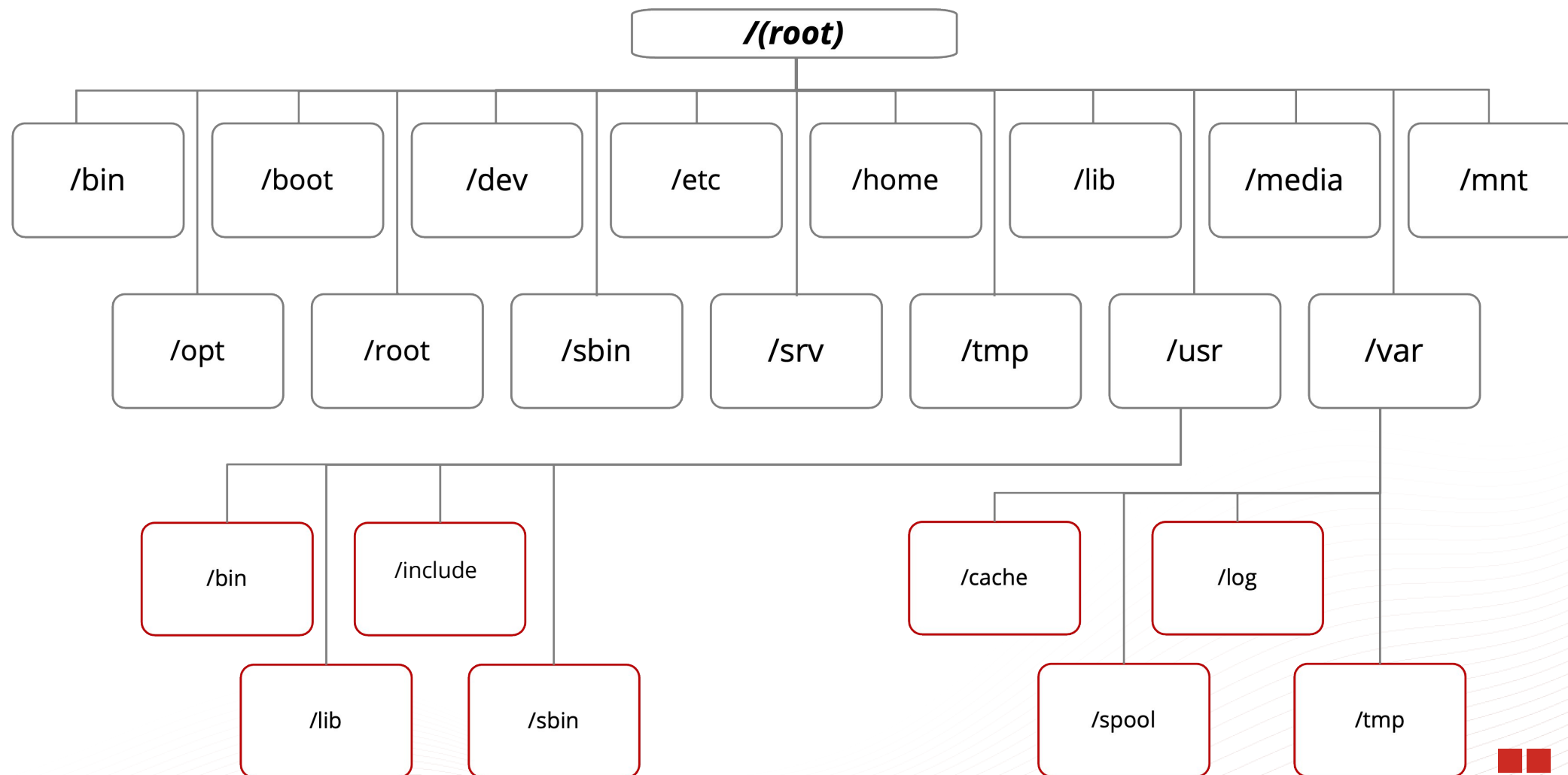
# Системные вызовы и системные библиотеки

- Системные вызовы представляют собой программный интерфейс, который используют программы для получения доступа к функциям ядра ОС (создание нового процесса, запись данных в файл и т.д.)
- Системный вызов определяет точку входа в ядро для запроса прикладным процессом требуемых функций, которые в дальнейшем выполняются ядром от имени процесса
- Библиотечные функции - это код, используемый для решения типовых задач (например, форматированный вывод на дисплей - `printf()` или вычисление синуса `sin()`)
- Код библиотечных функций присоединяется к коду программы

# Понятие дистрибутива

- Дистрибутив Linux — это комплект взаимосогласованного программного обеспечения, включающий ОС на базе ядра Linux, системное программное обеспечение и приложения, а также систему управления программным обеспечением
- Большая часть программного обеспечения в дистрибутиве является свободным (распространяется по свободным лицензиям)
- Минимальным объектом дистрибутива является программный пакет
- Программные пакеты могут собираться как в двоичном виде, так и в исходных кодах
- Дистрибутивы Linux размещаются в специальном хранилище - репозитории (в локальном или сетевом)

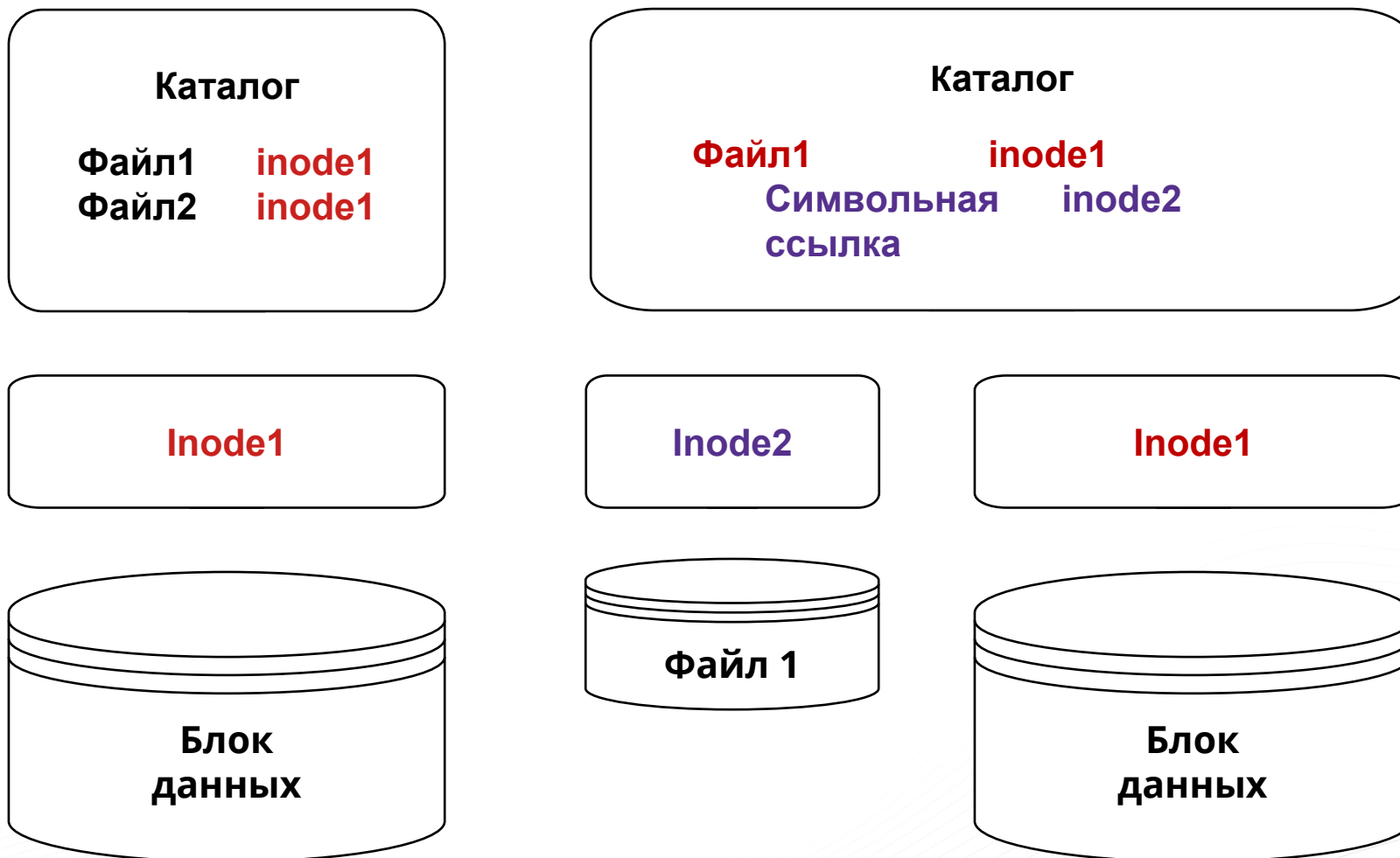
# Логическая структура файловой системы



# Типы специальных файлов

- Регулярный файл
- Каталог – сопоставляет имя файла с номером inode
- Символическая ссылка – в области данных содержится имя файла, на который указывает ссылка
- Жесткая ссылка - не является специальным файлом, это еще одно имя файла. В inode хранится количество жестких ссылок (link count)
- Файл устройств (блочных и символьных) – вызов драйвера
- Локальные (UNIX) сокеты (socket) и именованные (pipe) каналы

# Символические и жесткие ссылки



# Команды навигации по файловой системе

- **cd имя\_каталога** – переход в каталог
- **pwd** – абсолютное имя текущего
- **ls (-l -a -d -i)** – получение информации о файлах/каталогах
- **dir** - получение информации о файлах/каталогах (основные параметры те же, что и у ls)
- **tree** - вывод информации о содержимом каталогов в древовидном



# Команды создания файлов

- **touch имя\_файла** - создание пустого регулярного файла
- **mkdir имя\_каталога (-p)** — создание нового каталога
- **ln -s имя\_файла имя\_ссылки** — создание символической ссылки
- **ln имя\_файла имя\_жесткой\_ссылки** — создание жесткой ссылки
- **mknod** - создание специального файла устройств
- **mkfifo** - создание именованного канала

# Операции с файлами

- **cp (-a, -r, -s, -d)** – копирование файлов и каталогов
- **mv** – перемещение/переименование файлов и каталогов
- **rm (-r, -f)** – удаление файлов
- **rmdir имя\_каталога** – удаление пустого каталога
- Пустой каталог содержит две записи (. и ..)

# Операции с файлами

- **file** - информация о характере содержимого файла
- **which** - поиск программы по имени в каталогах, перечисленных переменной PATH
- **find точка\_начала\_поиска параметры\_поиска** - поиск файлов по атрибутам файлов. Точка\_начала\_поиска - имя каталога, с которого начинается поиск файлов

# Часто используемые параметры поиска утилиты find:

- **name (-iname)** поиск по имени с учетом регистра и без (можно использовать символы \* ? []);
- **size** - поиск по размеру файла(перед размером можно ставить символы + (больше), - (меньше));
- **user, -group** - поиск по владельцу и группе файла;
- **perm** - поиск по правам доступа;
- **mtime** - поиск по времени изменения файла;
- **type** - поиск по типу (f - обычный файл, d -каталог, l символическая ссылка);
- **exec** командная строка \; - выполнение действий над найденными файлами. Место, куда должны помещаться имена найденных файлов обозначаются символами {}. Командная строка, передаваемая утилите find, должна заканчиваться символом ; (точка с запятой).

# Команды для просмотра текстовых файлов

- **cat имена\_файлов** – объединение (конкатенация) файлов и перенаправление их в стандартный выходной поток
- **tac имена\_файлов** – работает как cat, но выводит строки с конца файла
- **head имя\_файла** – выводит n (по умолчанию 10) первых строк файла
- **tail имя\_файла** – выводит n (по умолчанию 10) последних строк файла
- **tail -f имя\_файла** – выводит добавляемые в файл строки в режиме реального времени
- **less имя\_файла** — постраничный вывод страниц файла

# Команды-фильтры

- **wc** — подсчет количества строк, слов
- **sort** - сортировка строк
- **cut** - вырезание символов или полей из текстовых строк
- **uniq** - показывает или убирает повторяющиеся строки
- **tr** - замена одних символов на другие или удаление символов
- **tee имя\_файла** — одновременное перенаправление стандартного входного потока в файл и в стандартный выходной поток
- **nl** — нумерация строк с использованием разных форматов
- **xargs** — создание командных строк с использованием данных, получаемых из стандартного входящего потока

# Утилита **grep**

Утилита соответствующих регулярному выражению, в текстовых потоках

Формат использования:

**grep [параметры] 'регулярное\_выражение' [файлы]**

Команды из семейства **grep**:

- **grep** — поиск с использованием базового набора метасимволов регулярных выражений
- **egrep (grep -E)** — поиск с использованием расширенного набора метасимволов регулярных выражений
- **fgrep (grep -F)** — поиск по «фиксированной» строке (специальные значения «метасимволов игнорируется»)

# Команда awk

- Утилита awk — предназначена для построчной обработки текстовых потоков
- Формат использования awk:

**awk [параметры] 'правила\_отбора{действия}' [файлы]**

Утилита awk работает с полями строк с помощью встроенных переменных (поля разделяются любым количеством пробелов или табуляций, можно задать разделитель с помощью -F)



# Практическая работа

1. Найдите строку, содержащую слово student в файле /etc/group
2. Отобразите все строки, содержащие слово root из файлов /etc/passwd и /etc/group
3. Подсчитайте количество строк, со словом student в файле /etc/group
4. Подсчитайте количество строк, в которых НЕТ слова student в файле /etc/group
5. Используя команду grep, отобразите все строки, содержащие слово List или слово System в файле /etc/passwd
6. Найдите в директории /usr все файлы и директории, имя которых заканчивается на ln
7. Найдите в домашней директории пользователя student только файлы
8. Найдите в домашней директории пользователя student только директории
9. У кого из пользователей в качестве оболочки используется bash
10. Найдите все файлы принадлежащие пользователю student
11. Найдите все файлы принадлежащие группе ftp
12. Найдите все файлы несуществующих пользователей или групп
13. Найдите все файлы являющиеся символическими ссылками в директории /etc
14. Найдите все файлы в системе размером более 200M
15. Найдите все файлы с расширением .html в директории /usr/share/doc
16. Найдите в директории /tmp файлы нулевого размера, а затем удалите не нужные из них (нужные файлы система удалить не даст)



**Спасибо за внимание!**

**[www.red-soft.ru](http://www.red-soft.ru)**  
**[redos@red-soft.ru](mailto:redos@red-soft.ru)**

