

ПОР → Р



”



”””



”



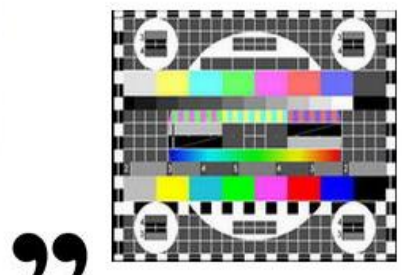
，



3 = 3

+ Б

4 = E



В



2



1 = Д

Я

Файловая система Linux



В **Windows** наиболее распространенными системами являются **FAT 32** и **NTFS**.

NTFS – более "продвинутая" система, одной из особенностей которой является обеспечение более высокой скорости работы на дисках относительно больших размеров.

В **Linux** используются другие файловые системы. В основном, это **Ext2** и **Ext3** (вторая и третья расширенные файловые системы).

Ext3 – это аналог Ext2, однако имеющая ряд модернизаций, одной из которых является поддержка журналирования.



► Unix базируется на двух основных понятиях: **процесс** и **файл**.

Процессы являют собой динамическую сторону системы, это субъекты; а **файлы** – статическую, это объекты действия процессов.

Почти весь интерфейс взаимодействия процессов с ядром и друг с другом выглядит как **запись/чтение файлов**.

Типы процессов

- ▶ **Задача**– это процесс, который выполняет свою работу, стремясь побыстрее закончить ее и завершиться
- ▶ **Демон**– ждет событий, которые он должен обработать, обрабатывает произошедшие события и снова ждет; завершается он как правило по приказу другого процесса, чаще всего его убивает пользователь, дав команду **kill номер_процесса**



Типы имени файла в Unix

- ▶ **Краткое имя** – идентифицирует файл в пределах одного каталога. Оно может содержать так называемый суффикс, отделяемый точкой. Например **test.c** – имя файла
- ▶ **Полное имя** однозначно определяет файл. Оно состоит из цепочки имен каталогов, через которые проходит маршрут от корневого каталога до данного файла. Например, **/mnt/rk2/test.c**, где **mnt** и **rk2** – имена каталогов, а **test.c** – имя файла.

Типы имени файла в Unix

- ▶ Каждому **полному имени** в ОС соответствует **только один файл**, однако **файл может иметь несколько различных имен**, так как ссылки на один и тот же файл могут содержаться в разных каталогах (жесткие связи).

Типы имени файла в Unix

- ▶ Имя файла относительно текущего каталога называется **относительным**.
- ▶ Оно представляет собой цепочку имен каталогов, через которые проходит маршрут от текущего каталога до данного файла.
- ▶ Относительное имя в отличие от полного не начинается с символа "/".

Например, если принять за текущий каталог **/mnt**, то относительное имя файла **test.c** будет **rk2/test.c**.

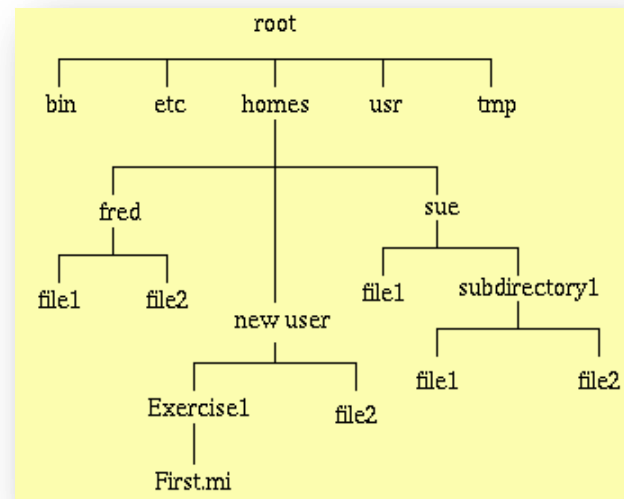
Данные в файлах

Программы, выполняемые под управлением системы UNIX, **не содержат никакой информации относительно внутреннего формата**, в котором ядро хранит файлы данных, **данные в программах представляются как бесформатный поток байтов**.

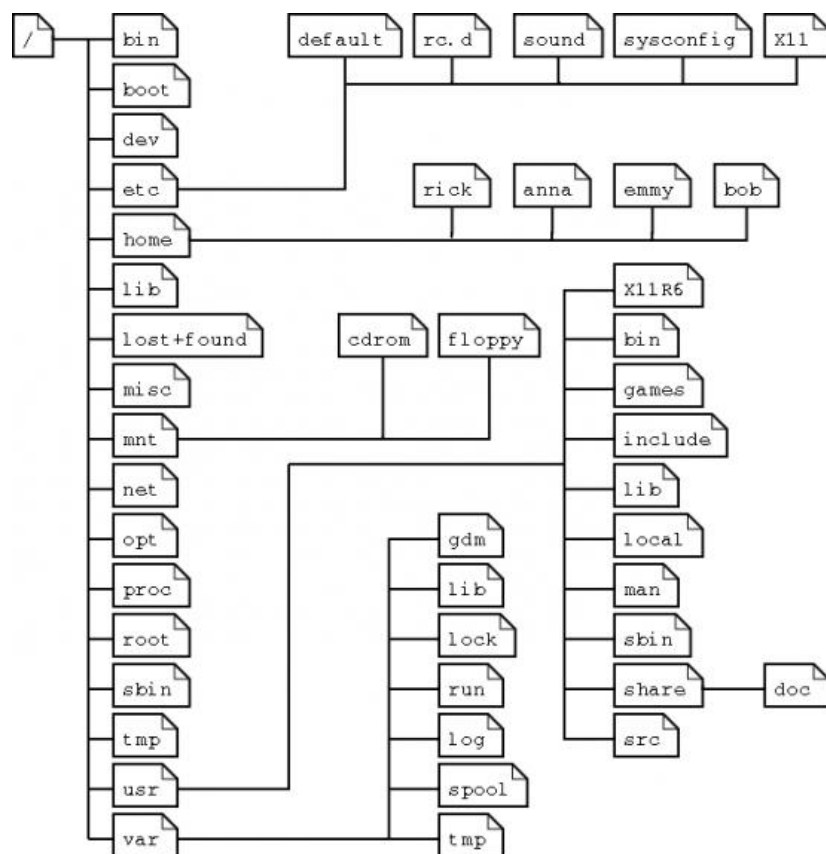
Программы могут интерпретировать поток байтов по своему желанию, при этом любая интерпретация никак не будет связана с фактическим способом хранения данных в операционной системе.

Каталоги (директории)

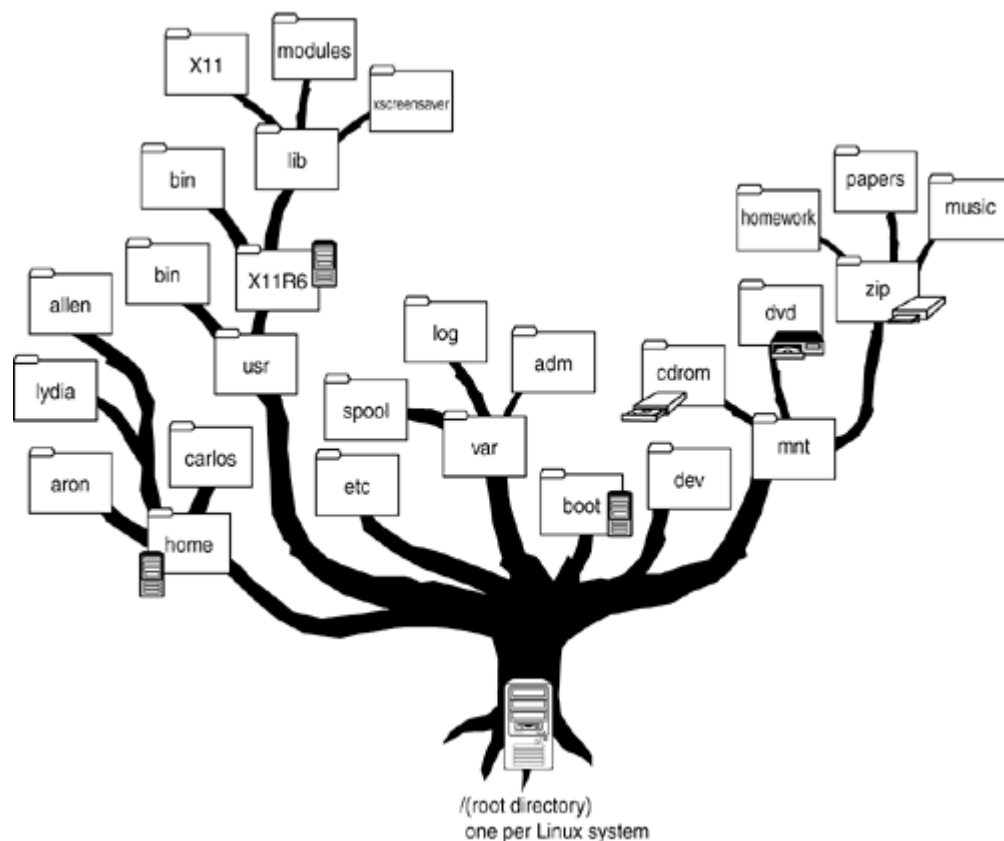
Каталоги похожи на обычные файлы в одном отношении: система представляет информацию в каталоге набором байтов, но эта информация включает в себя имена файлов в каталоге в объявленном формате для того, чтобы операционная система и программы, могли их обнаружить.



- ▶ **Каждый файл** (но не директория) может быть известен под **несколькими именами**, но **обязательно лежащими на одном разделе**. Все ссылки на файл равноправны; файл стирается, когда удаляется последняя ссылка на файл.



- ▶ Пользователи **могут создавать файлы**, если **разрешен доступ к каталогу**.
- ▶ Вновь созданные файлы **становятся листьями в древовидной структуре файловой системы**.



Учетные записи в Linux



- ▶ **Учетная запись пользователя** – это необходимая для системы *информация* о пользователе, хранящаяся в специальных файлах. *Информация* используется Linux для аутентификации пользователя и назначения ему прав доступа.
- ▶ **Аутентификация** – системная процедура, позволяющая Linux определить, какой именно *пользователь* осуществляет вход.

Вся информация о пользователе обычно хранится в файлах **/etc/passwd** и **/etc/group**.

1) **/etc/passwd** – этот *файл* содержит информацию о пользователях.

Запись для каждого пользователя занимает одну строку:

root: \$1\$OvdTRu2w\$Cm5gk.6w6nmNdUierh5pu:0:0:root:/root:/bin/bash

имя пользователя зашифрованный пароль UID GID Настоящее имя пользователя домашний каталог оболочка

- ▶ **UID** – числовой **идентификатор пользователя**. Система использует его для распределения прав файлам и процессам.
- ▶ **GID** – числовой **идентификатор группы**. Имена групп расположены в файле **/etc/group**. Система использует его для распределения прав файлам и процессам.

2) **/etc/group** – этот файл содержит информацию о группах, к которым принадлежат пользователи:

project: \$1\$QydTRu2w\$Cm5gk.6w6nmNdUjerh5pu:100:root,bin,daemon

The diagram shows the entry "project: \$1\$QydTRu2w\$Cm5gk.6w6nmNdUjerh5pu:100:root,bin,daemon" with brackets underneath. The first bracket under "project:" is labeled "Имя группы". The second bracket under "\$1\$QydTRu2w\$Cm5gk.6w6nmNdUjerh5pu:" is labeled "Шифрованный пароль". The third bracket under "100:" is labeled "GID". The fourth bracket under "root,bin,daemon" is labeled "Пользователи, включенные в несколько групп".

- ▶ **GID** – числовой идентификатор группы. Система использует его для распределения прав файлам и процессам.
- ▶ **Пользователи, включенные в несколько групп** – В этом поле через запятую отображаются те пользователи, у которых по умолчанию (в файле **/etc/passwd**) назначена другая группа.

- ▶ На сегодняшний день хранение паролей в файлах `passwd` и `group` считается **ненадежным**.
- ▶ В новых версиях Linux применяются так называемые теньевые файлы паролей – **shadow** и **gshadow**. Права на них назначены таким образом, что даже чтение этих файлов без прав **суперпользователя** невозможно.
- ▶ При использовании теньевых паролей в `/etc/passwd` и `/etc/group` вместо самого пароля **устанавливается символ 'x'**, что и является указанием на хранение пароля в `/etc/shadow` или `/etc/gshadow`.

- ▶ Файл **shadow** хранит защищенную информацию о пользователях, а также обеспечивает *механизмы* устаревания паролей и учетных записей. Вот структура файла **shadow**

cisco:\$1\$oAJZcVg0\$EGORy8Mh3swT1RfJeX.UR0:13770:10:99999:7:30:99999:

The diagram shows a line of text representing a shadow file entry: 'cisco:\$1\$oAJZcVg0\$EGORy8Mh3swT1RfJeX.UR0:13770:10:99999:7:30:99999:'. Below this line, several brackets are drawn to group parts of the string. Brackets labeled 'а' and 'б' are under the first two fields. Brackets labeled 'в', 'г', 'д', 'е', 'ж', 'з', and 'и' are under the last seven fields. A long bracket labeled 'б' spans from the start of the second field to the end of the sixth field.

- ▶ а – **имя пользователя** ;
- ▶ б – **шифрованный пароль**
- ▶ в – число дней с последнего изменения пароля, начиная с 1 января 1970 года;
- ▶ г – число дней, перед тем как пароль может быть изменен;
- ▶ д – число дней, после которых пароль должен быть изменен;
- ▶ е – число дней, за сколько пользователя начнут предупреждать, что пароль устаревает;
- ▶ ж – число дней, после устаревания пароля для блокировки учетной записи;
- ▶ з – дней, отсчитывая с 1 января 1970 года, когда учетная запись будет заблокирована;
- ▶ и – зарезервированное поле;

- ▶ Файл **gshadow** так же накладывает дополнительную функциональность, вкуже с защищенным хранением паролей групп. Он имеет следующую структуру:

root:\$1\$QydTRu2w\$Cm5gk.6w6nmNdUjerh5pu:root:cisco,oem

Имя группы Шифрованный пароль администратор группы Список пользователей

Пользователь Root

- ▶ **Root**– пользователь с неограниченными правами
- ▶ Идентификаторы **UID** и **GID** такого пользователя всегда **0**
- ▶ Для пользователя **root** *права доступа* к файлам и процессам не проверяются системой.

Права пользователей

- ▶ Для распознавания пользователя используется пароль.
- ▶ Сегодня в большинстве версий Linux пароли **шифруются** по алгоритмам **3DES** и **MD5**. Когда *алгоритм 3DES* является обратимым, то есть такой *пароль* можно расшифровать, **MD5** – это необратимое преобразование.
- ▶ Пароли, зашифрованные по алгоритму **3DES** **не применяются** при использовании **теневых файлов для хранения паролей**.