D-Link

Лекция 3.

Файлы и каталоги в Linux. Команды для работы с файлами и каталогами

Лекция 3. Файлы и каталоги в Linux. Команды для работы с файлами и каталогами

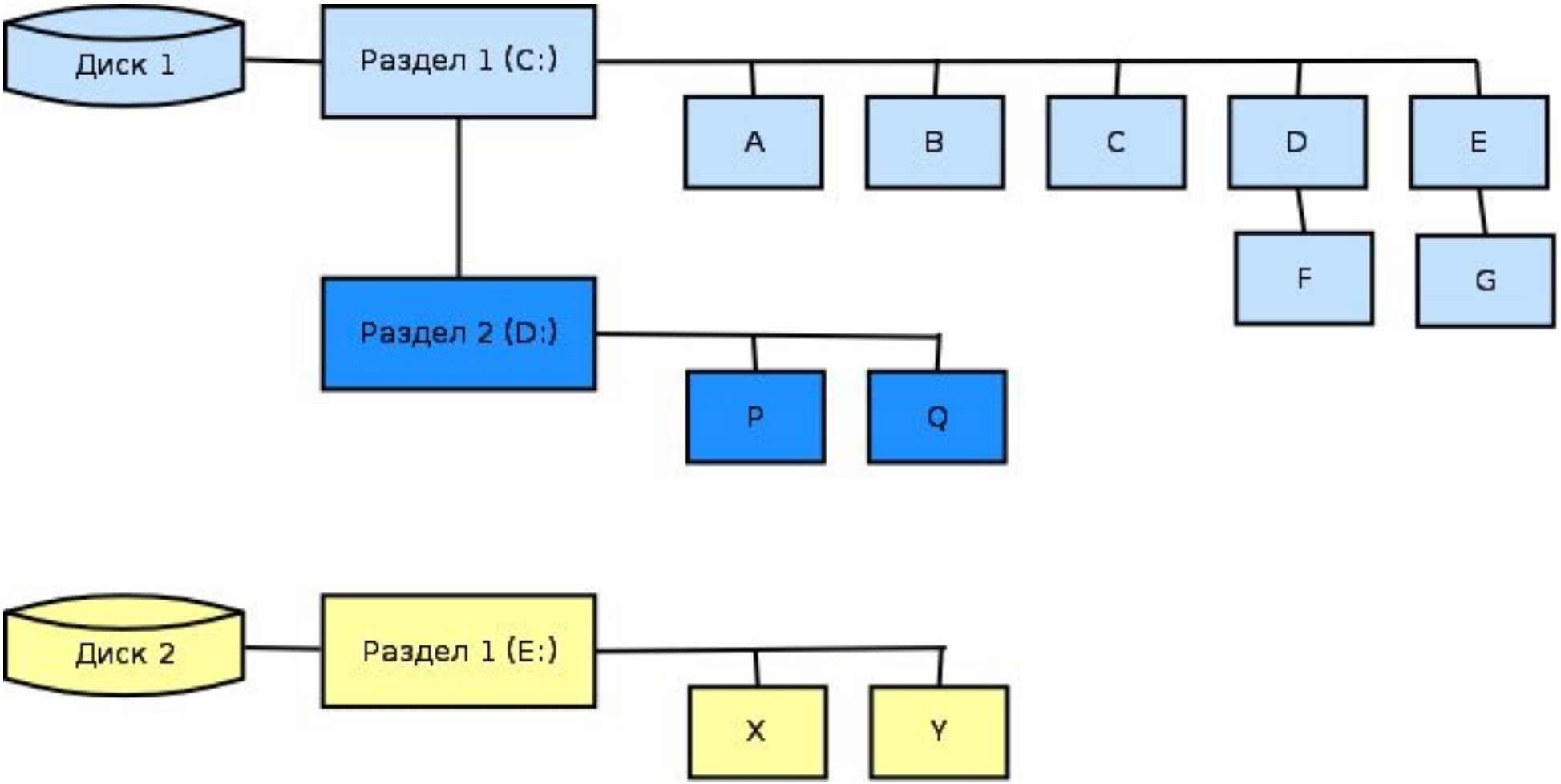
- Типы файлов в Linux
- Дерево каталогов Linux . Монтирование файловых систем
- Шаблоны имен файлов
- Команда ls
- Команды cd и pwd
- Команды mkdir и rmdir
- Команда cat
- Команды mv и cp
- Команда rm
- Работа со ссылками
- Понятия владельца файла и прав доступа
- Команда chmod

Типы файлов в Linux

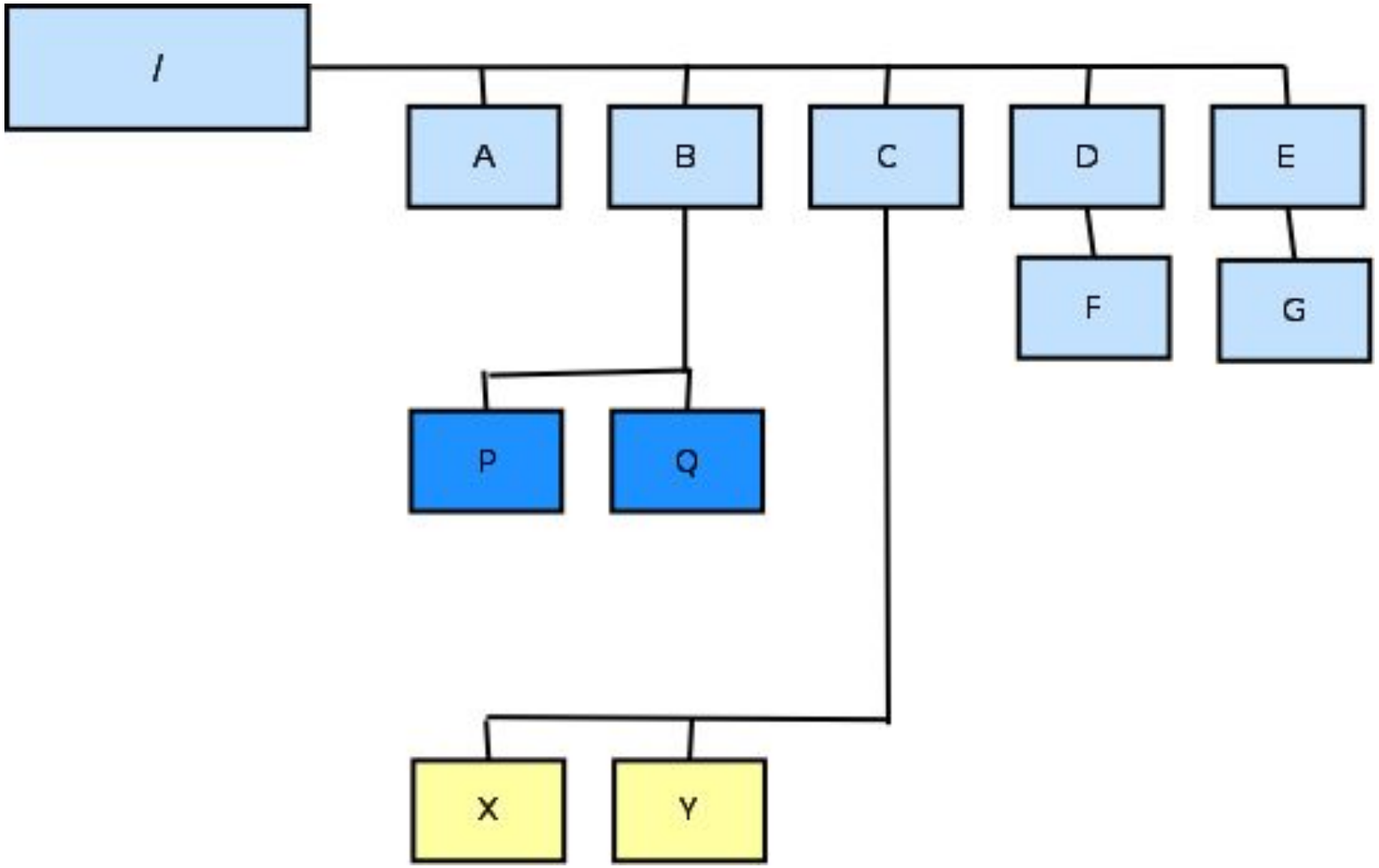
Одним из основных принципов построения Linux и Unix является принцип “**Everything is file**” (Все является файлом). Данный принцип приводит к тому, что типов файлов в Linux довольно много. Их шесть:

- **Обычный файл** (**file**) – это обозначенная некоторым именем последовательность данных, которые хранятся на диске (устройстве хранения данных).
- **Каталог** (**директория**, **directory**) формально тоже считается файлом, содержащим данные о файлах, хранящихся в нем.
- **Символьная** или **мягкая ссылка** (**символическая ссылка**, **symbolic link**, **simlink**) во многом похожа на ярлык Windows. Но предназначена она не для быстрого запуска программ, а для исключения дублирования и более оптимального использования места на диске.
- **Файл устройства** представляет собой способ обращения к различным устройствам с помощью файловых операций. Файлы устройств отображаются в каталоге /dev дерева каталогов Linux. Существуют два типа устройств – символьные (character) и блочные (block), работа с которыми происходит по-разному.
- **Канал** (**pipe**) представляет собой специальный механизм для обмена данными между процессами (исполняющимися программами), которые выполняются на одной машине.
- **Сокет** (**socket**) похож на канал в том смысле, что тоже является механизмом для обмена данными между двумя разными процессами. Однако сокет имеет два важных отличия от канала – процессы чаще всего запущены на разных компьютерах, а обмен данными осуществляется по сети с использованием стека протоколов TCP/IP.

Структура каталогов в Windows



Структура каталогов в Linux



Дерево каталогов Linux (1)

Каталог **/bin** содержит основные системные утилиты, необходимые в однопользовательском режиме, а так же при обычной работе всем пользователям.

Каталог **/boot** содержит файлы загрузчика, например, образ ядра Linux.

Каталог **/dev** содержит файлы устройств.

Каталог **/etc** содержит общесистемные конфигурационные файлы и конфигурационные файлы программ.

Каталог **/home** содержит домашние каталоги пользователей. При обычной работе чаще всего используется этот каталог.

Каталог **/lib** содержит файлы общесистемных разделяемых библиотек и модулей ядра.

Каталог **/media** содержит точки монтирования для сменных носителей, таких как CD и DVD-диски, flash-накопители и т.п.

Каталог **/mnt** содержит точки монтирования для временных разделов.

Каталог **/opt** содержит дополнительные пакеты приложений.

Дерево каталогов Linux (2)

Каталог **/proc** содержит виртуальную файловую систему proc, которая отображает в виде структуры файлов и каталогов различную информацию о работе системы.

Каталог **/root** является домашним каталогом пользователя root.

Каталог **/sbin** содержит основные системные программы для администрирования и настройки системы.

Каталог **/sys** является точкой монтирования еще одной виртуальной файловой системы sys, являющейся расширением proc.

Каталог **/usr** содержит большинство пользовательских приложений и утилит, использующихся в многопользовательском режиме.

Каталог **/var** содержит изменяемые файлы, такие как файлы регистрации, временные файлы и т.п.

Шаблоны имен файлов

При задании имени файла в Linux можно использовать **шаблоны имен файлов**. Они задаются с помощью специальных символов.

Символ “*****” заменяет несколько любых символов, в том числе и ни одного.

Шаблону “*****” соответствует любое имя файла, а шаблону “**a*b**” соответствуют имена “**ab**”, “**a1b**”, “**ahtb**”, “**a_file_b**” и т.п.

Символ “**?**” в шаблоне заменяет один любой символ в имени файла.

Шаблону “**a?b**” соответствуют имена файлов “**aab**”, “**abb**”, “**axb**”, “**a8b**” и т.п.

Символы квадратных скобок позволяют задать символ из заданного набора. Например, шаблону “**a[abc]b**” соответствуют имена файлов “**aab**”, “**abb**” и “**acb**”. В квадратных скобках можно указать диапазон.

Команда ls (1)

Команда **ls** выводит на экран содержимое заданного каталога.

```
$ ls [флаги] [каталог]
```

Если каталог не указан, то отображается содержимое текущего каталога.

```
$ ls /home/user/Prog
file.c fileout
$ ls
inst.txt ls.txt Prog
```

Команда содержит множество ключей, которые определяют:

- в каком формате выводить информацию,
- какую именно информацию об этих объектах нужно вывести,
- как упорядочить выводимую информацию.

За то, каким образом выводить содержимое каталога, отвечают опции -1, -m, -x, -C и -l.

За то, какая именно информация выводится на экран, отвечают опции -l, -a, -R.

Третья группа ключей отвечает за сортировку списка выводимых файлов. Это ключи -f, -t, -v, -S, -X, -r.

D-Link

Команда ls (2)

```
$ ls -al
итого 52
drwxr-xr-x 2 user user 4096 Янв 26 14:29 .
drwxr-xr-x 3 user user 4096 Янв 26 14:24 ..
-rw-r--r-- 1 user user   22 Янв 26 14:26 example2.txt
-rw-r--r-- 1 user user   22 Янв 26 14:25 example.c
-rw-r--r-- 1 user user   22 Янв 26 14:25 example.o
-rw-r--r-- 1 user user   22 Янв 26 14:25 example.out
-rw-r--r-- 1 user user   22 Янв 26 14:25 file100.txt
-rw-r--r-- 1 user user   22 Янв 26 14:25 file1.txt
-rw-r--r-- 1 user user   22 Янв 26 14:25 file20.txt
-rw-r--r-- 1 user user   22 Янв 26 14:25 file.a.txt
-rw-r--r-- 1 user user   22 Янв 26 14:25 fileb.txt
-rw-r--r-- 1 user user   22 Янв 26 14:25 textfile
-rw-r--r-- 1 user user   22 Янв 26 14:26 textfile2
```

Команды cd и pwd

Команда **cd** изменяет текущий каталог (сокр. от change directory).

```
$ cd [каталог]
```

Примеры:

$ cd /usr - перейти в каталог **/usr**,

$ cd .. - перейти в родительский каталог (находящийся в иерархии каталогов на один уровень выше),

$ cd - перейти в рабочий каталог пользователя.

$ cd – - перейти в предыдущий каталог.

Вывести на экран текущий каталог можно с помощью команды **pwd** (сокр. от print working directory). Команда не имеет параметров.

```
$ pwd
/home/user
```

D-Link

Команды mkdir и rmdir

Для создания одного или нескольких новых каталогов используется команда **mkdir**.

```
$ mkdir [ключи] имя_каталога …
```

```
$ mkdir dir_a dir_b dir_c
```

Используя ключ **-m** можно при создании каталога сразу задать ему права доступа.

```
$ mkdir -m 777 dir1
```

Если в команде **mkdir** задать ключ **-p**, то можно сразу создать цепочку вложенных друг в друга подкаталогов.

```
$ mkdir -p dir1/dir2/dir3
```

Для удаления пустых каталогов предназначена команда **rmdir**.

```
$ rmdir [ключи] каталог …
```

```
$ rmdir dir1 dir2
```

Если задать в команде ключ **-p**, то она позволяет удалять вложенные друг в друга пустые подкаталоги.

```
$ rmdir -p ./dir1/dir2/dir3
```

Команда cat

Очень полезной командой является команда **cat**. Она позволяет объединять файлы, но не только это.

```
$ cat [ключи] имя_файла …
```

Команду **cat** можно использовать для:

1. Вывода файла на экран

   ```
   $ cat file
   ```

2. Ввода файла с клавиатуры. В конце ввода нужно нажать Ctrl+D.

   ```
   $ cat > file
   ```

3. Объединения нескольких файлов в один.

   ```
   $ cat file1 file2 file3 > file123
   ```

Например:

```
$ cat file1
This is file1
$ cat file2
This is file2
$ cat file1 file2 > file3
$ cat file3
This is file1
This is file2
```

Команды mv и cp

Переименование или перемещение файла выполняется командой **mv**.

```
$ mv [флаги] исходный_файл файл_назначения
$ mv [флаги] исходный_файл… каталог_назначения
```

Если последний параметр команды является именем каталога , то команда mv перемещает файлы в этот каталог.

Команда **cp** позволяет выполнить копирование указанного файла под новым именем или копировать несколько файлов под старыми именами в указанный каталог.

```
$ cp [флаги] файл1 файл2
$ cp [флаги] файл… каталог_назначения
```

Если последним параметром команды является имя каталога, то в данный каталог копируются указанные в команде файлы с сохранением их имен.

Ключи команды:

-l - вместо копирования файлов будут созданы их жесткие ссылки.

-s - вместо копирования файлов будут созданы их символьные ссылки.

-r - рекурсивный режим копирования, когда вместе с каждым копируемым каталогом копируются вложенные в него файлы и подкаталоги.

Команда rm

Команда **rm** позволяет удалить указанные файлы или каталоги. Она имеет формат:

```
$ rm [флаги] файл…
```

В Linux удаленный файл восстановить чаще всего **невозможно**.

Достаточно мощным, но в то же время опасным ключом команды является ключ **-r**, включающий режим рекурсивного удаления каталогов, позволяющий удалять каталоги вместе с вложенными в них файлами и подкаталогами. Например, следующая команда удаляет каталог prog и все его подкаталоги:

```
$ rm –r prog
```

Работа со ссылками

Для создания ссылок на файлы в Linux предназначена команда **ln**. Она может создавать как жесткие, так и символьные ссылки.

```
$ ln [ключи] имя_файла [имя_ссылки]
$ ln [ключи] имя_файла … каталог
```

Первый вариант команды создает (по умолчанию - в текущем каталоге) ссылку с именем ссылка на заданный файл с именем имя_файла.

Второй вариант команды создает в заданном каталоге ссылки на указанные файлы, причем имена файлов ссылок будут совпадать с именами исходных файлов.

По-умолчанию создается жесткая ссылка. Если требуется создать символьную ссылку, то в команде нужно указать ключ **-s**.

D-Link

Понятия владельца файла и прав доступа

В Linux каждый файл имеет своего **владельца**. Им обычно является пользователь, создавший этот файл. Однако владелец файла может быть изменен администратором системы.

Владелец файла определяет **права доступа** к файлу. Для каждого файла могут быть заданы: **право на чтение**, **право на запись** и **право на выполнение**. Права задаются отдельно для трех категорий пользователей:

1. для владельца файла,
2. для пользователей, входящих в группу владельца,
3. для всех остальных пользователей.

Основные права доступа часто задают в виде трех восьмеричных цифр, например 764. Первая цифра (7) определяет права владельца файла, вторая (6) – права группы владельца, а третья – права остальных пользователей. Каждая восьмеричная цифра прав складывается из трех двоичных разрядов – права на исполнение (в разряде единиц), права на запись (в разряде двоек) и права на чтение (в разряде четверок).

Кроме перечисленных есть еще три дополнительных бита прав доступа. Это так называемые **SUID-**, **SGID-** и **sticky-биты**. Они хранятся в старшей тройке битов прав доступа.

Команда chmod (1)

Для изменения прав доступа к файлу предназначена команда **chmod**.

```
$ chmod [ключи] режим файл …
```

Параметр режим, задающий права доступа, может задаваться в двух форматах – **числовом** и **символьном**.

```
$ chmod 4764 report.txt
```

Символьная форма описания режима прав доступа задается в следующем формате:

```
[ugoa] [+-=] [rwxstugo]
```

Первая часть строки режима определяет, для кого изменяются права доступа – для владельца (**u**), группы владельца (**g**), остальных пользователей (**o**) или всех пользователей (**a**).

Вторая часть строки режима определяет, как изменяются права доступа – они добавляются к уже имеющимся (**+**), устанавливаются заново (**=**) или удаляют часть имеющихся прав (**-**).

Третья часть строки режима задает новые права доступа для пользователя, заданного первой строкой: **r** – чтение, **w** – запись, **x** – выполнение, **s** – SUID- или SGID-биты, **t** – sticky-бит.

Команда chmod (2)

Установка файлу f1 прав доступа на чтение и запись для всех пользователей:

```
$ chmod a=rw f1
```

Установка для файлов f1 и f2 полных прав для владельца, прав на чтение и выполнение для группы владельца и права на чтение для остальных пользователей.

```
$ chmod u=rwx,g=rx,o=r f1 f2
```

Добавление группе владельца файла f1 права на запись данного файла.

```
$ chmod g+w f1
```

Установка SUID-бита для программы

```
$ chmod u+s prog
```

Установка SGID-бита для программы

```
$ chmod g+s prog
```