

Информационные технологии

Лекция №2



Команды Linux

Команды Linux

- Команды консоли Linux - взаимодействие между пользователем и ОС
- Выполняются через командную строку путем ручного ввода
- За каждой командой - соответствующая программа в системе
- Команды (программы) могут иметь опции (ключики)
- Самый мощный инструмент управления системой

Общие команды

<code>\$ uname -a</code>	Показать версию ядра Linux
<code>\$ date</code>	Текущее время и дата
<code>\$ exit</code>	Завершить сеанс текущего пользователя
<code>\$ passwd</code>	Изменить пароль текущего пользователя
<code>\$ reboot</code>	Перезагрузка системы
<code>\$ halt</code>	Выключение системы
<code>\$ hostname</code>	Сетевое имя компьютера
<code>\$ history</code>	Показывает список последних введенных команд
<code>\$ man <cmd></code>	Справка по команде <cmd>
<code>\$!!</code>	Выполнить последнюю команду

```
spider@sbi:~$ uname -a
Linux sbi 2.6.32-71-generic-pae #138-Ubuntu SMP Thu Dec 18 17:53:29 UTC 2014 i686 GNU/Linux
```

```
spider@sbi:~$ date
Пнд Сен 19 14:49:20 NOVT 2016
```

```
spider@sbi:~$ hostname
sbi
```

Команды работы с файлами

<code>\$ pwd</code>	Показать имя текущей папки
<code>\$ ls</code>	Показать содержимое папки
<code>\$ cd /home/alex/work</code>	Перейти в папку
<code>\$ cat /etc/passwd</code>	Вывести содержимое файла
<code>\$ cp /etc/passwd /home/alex/secret</code>	Скопировать файл
<code>\$ rm diplom_final.doc</code>	Удалить файл <code>diplom_final.doc</code>
<code>\$ mkdir /home/alex/diplom</code>	Создать папку
<code>\$ touch text1.txt</code>	Изменить дату/время файла на текущие
<code>\$ rmdir /home/alex/diplom</code>	Удалить папку
<code>\$ mv text1.txt text2.txt</code>	Переименовать файл <code>text1.txt</code> в <code>text2.txt</code>

Работа с файлами и ресурсами системы

<code>\$ chmod</code>	Изменить права файла
<code>\$ chown</code>	Изменить владельца
<code>\$ df</code>	Просмотр свободного места на дисках
<code>\$ free</code>	Объем свободной оперативной памяти
<code>\$ top</code>	Статистика процессов CPU
<code>\$ lspci</code>	Список подключенных устройств к шине PCI
<code>\$ lsusb</code>	Список подключенных устройств к шине USB
<code>\$ mount</code>	Монтирование новых разделов/устройств
<code>\$ ps</code>	Список процессов запущенных в системе
<code>\$ grep</code>	Поиск образца в файле

Объединение команд

```
# cd /home/name; ls -la # ; - последовательное выполнение команд
# ls file.txt && echo "DATA" >>file.txt # && - выполнение команды при условии успешного завершения предыдущих
# ls file1.txt || echo "DATA" >file1.txt # || - выполнение команды при условии, что предыдущая завершилась с
ошибкой
# touch $(echo $(date "+%Y-%m-%d").txt) # $() - использование выходных данных одной команды при вызове другой
команды
# ls -la /usr/bin | more # | - передача выходных данных одной команды на вход другой команды
# ls > menu.txt # > - перенаправление выходных данных в файл
# wc -l < menu.txt # < - использование содержимого файла в качестве входных данных
# find / -name .profile 2>&1 |less # 2>&1 - поток ошибок туда, куда направлен вывод
```

Потоки ввода/вывода

stdin	0	поток ввода	клавиатура
stdout	1	поток вывода	терминал
stdin	2	поток ошибок	терминал

Файловая система Linux

- Файл - непрерывная последовательность байтов определенной длины
- Имена файлов:
 - До 255 символов
 - Любые символы кроме #0 и /
 - Не рекомендуется использовать:
 - ! @ # \$ % & ~ * () [] { } ' " \ : ; < > ` пробел
 - `$ mkdir "My favorite files"`
 - Точек в имени может быть несколько
 - Никакого смысла расширение файла не имеет
 - Если имя файла начинается с точки (.filename) этот файл становится "невидимым"
 - Символы верхнего и нижнего регистров различаются (myText.doc и MyText.doc - разные файлы)
- Текущая папка - .
- "Верхняя" папка - ..

Файлы группируются в каталоги

Каталоги могут входить в другие каталоги

Иерархическая структура каталогов начинающаяся с корневого /

“Дерево” каталогов: каталог - узел дерева, файл - лист

Единая каталоговая структура для всех носителей

В единую каталоговую структуру можно подключить любое число каталогов, физически расположенных на разных носителях

Полное имя файла - список имен вложенных друг в друга подкаталогов начиная с корневого:

`/home/spider/documents/diplom.doc`

- У каждого пользователя есть свой “домашний каталог”
- В “домашнем каталоге” пользователь обладает всеми правами
- Обычно “домашние каталоги” лежат в папке /home:
 - /home/ivan
 - /home/stepan
 - /home/sveta
- ~ значок обозначающий “домашний каталог” пользователя
 - /home/ivan/docs ~/docs
- Для изменения текущего каталога используется команда cd
 - # cd /home/ivan/disser
 - # cd ..
 - # cd /etc
- Команда ls выводит список файлов в каталоге
 - # ls # список файлов в текущем каталоге
 - # ls /etc # список файлов в каталоге /etc

- Самые простые:
 - Файлы
 - Каталоги
- Специальные типы файлов:
 - Файлы физических устройств
 - Именованные каналы (named pipes)
 - Гнезда (sockets)
 - Символические ссылки (symlinks)

- С точки зрения ОС Linux все подключаемые к компьютеру устройства (диски, терминал, принтер, модем, сетевая карта и пр.) представляются файлами
 - Например, если нужно вывести на экран информацию, можно произвести запись в файл `/dev/tty01`
- Физические устройства:
 - Символьные (байт-ориентированные)
 - Блочные (блок-ориентированные)
- Символьные:
 - Терминал
 - Принтер
 - Монитор
 - Последовательные порты USB, COM
- Блочные:
 - Жесткий диск
 - Компакт-диск
- Взаимодействие с устройствами производится драйверами устройств
- Драйвер устройства создает коммуникационный интерфейс, выглядит как файл в каталоге `/dev`

Примеры файлов устройств

`/dev/console` - Системная консоль, т.е. монитор и клавиатура

`/dev/hd` - Жесткие диски с IDE интерфейсом

`/dev/sd` - Жесткие диски с интерфейсом SCSI

`/dev/fd` - Гибкие диски

`/dev/tty` - Файлы поддержки пользовательских консолей. Виртуальные консоли.

Переключение: `<Alt><F1>...<Alt><Fn>`

`/dev/ttyS` - Файлы последовательных портов

`/dev/null` - “Черная дыра”. Все что записывается в это устройство навсегда теряется

Примеры:

```
# cat /etc/passwd >/dev/null
```

Именованные каналы и гнезда

- Именованные каналы, или буферы FIFO (First In-First Out).
- Файлы этого типа служат в основном для того, чтобы организовать обмен данными между разными приложениями (pipe переводится с английского как труба).
- Канал— это очень удобное и широко применяемое средство обмена информацией между процессами.
- Все, что один процесс помещает в канал, другой может оттуда прочитать.
- Если два процесса, обменивающиеся информацией, порождены одним и тем же родительским процессом (а так чаще всего и происходит), канал может быть неименованным.
- В противном случае требуется создать именованный канал, что можно сделать с помощью программы `mkfifo`.
- При этом собственно файл именованного канала участвует только в инициации обмена данными.
- `# ls | sort | lp`

- Файл в Linux может иметь несколько имен или "жестких ссылок".
- Жесткая ссылка является просто еще одним именем для исходного файла. После создания жесткой ссылки невозможно различить, где исходное имя файла, а где ссылка. Если вы удаляете один из этих файлов (точнее одно из этих имен), то файл еще сохраняется на диске (пока у него есть хоть одно имя-ссылка).
- Одно из применений жестких ссылок состоит в том, чтобы предотвратить возможность случайного удаления файла.
- Другой тип ссылок - символические ссылки.
- Эти ссылки тоже могут рассматриваться как дополнительные имена файлов, но в то же время они представляются отдельными файлами— файлами типа символических ссылок.
- В отличие от жестких ссылок символические ссылки могут указывать на файлы, расположенные в другой файловой системе, например, на монтируемом носителе, или даже на другом компьютере.
- Если исходный файл удален, символическая ссылка не удаляется, но становится бесполезной.
- Используйте символические ссылки в тех случаях, когда хотите избежать путаницы, связанной с применением жестких ссылок.

Права доступа к файлам и каталогам

- Linux— система многопользовательская
- Разграничение доступа к файлам и каталогам
- Механизмы разграничения доступа, разработаны для UNIX в 70-х годах
- очень просты, но очень эффективны

В основе - имена пользователей и имена групп пользователей.

каждый пользователь может быть включен в одну или несколько групп.

Члены разных групп могут иметь разные права по доступу к файлам

- В индексном дескрипторе каждого файла записаны имя так называемого владельца файла и группы, которая имеет права на этот файл.
- Первоначально, при создании файла его владельцем объявляется тот пользователь, который этот файл создал.
- Группа тоже назначается при создании файла— по идентификатору группы процесса, создающего файл.
- Владельца и группу файла можно поменять в ходе дальнейшей работы с помощью команд `chown` и `chgrp`

- Три вида прав доступа:
 - Чтение (read) - r
 - Запись (write) - w
 - Исполнение (execute) - x
- Права действуют на файлы и каталоги
 - Право записи для каталога означает что вы можете создавать там файлы
 - Право на чтение - просмотр содержимого каталога
 - Право на исполнение - возможность “войти” в каталог

- Система проверяет, совпадает ли имя пользователя с именем владельца файла.
- Если эти имена совпадают, то проверяется, имеет ли владелец соответствующее право доступа: на чтение, на запись или на выполнение.
- Если право такое есть, то соответствующая операция разрешается.
- Если же нужного права владелец не имеет, то пользователю выдается сообщение о невозможности выполнения затребованного действия
- Если имя пользователя, обращающегося к файлу, не совпадает с именем владельца, то система проверяет, принадлежит ли владелец к группе, которая сопоставлена данному файлу.
- Если принадлежит, то для определения возможности доступа к файлу используются атрибуты, относящиеся к группе, а на атрибуты для владельца и всех остальных пользователей внимания не обращается.
- Если же пользователь не является владельцем файла и не входит в группу файла, то его права определяются атрибутами для остальных пользователей.
- Таким образом, третья группа атрибутов, определяющих права доступа к файлу, относится ко всем пользователям, кроме владельца файла и пользователей, входящих в группу файла.

Для изменения прав доступа к файлу используется команда `chmod`.

```
# chmod wXr имя-файла
```

- где вместо символа `w` подставляется
 - либо символ `u` (т.е. пользователь, который является владельцем);
 - либо `g` (группа);
 - либо `o` (все пользователи, не входящие в группу, которой принадлежит данный файл);
 - либо `a` (все пользователи системы, т.е. и владелец, и группа, и все остальные).
- Вместо `X` ставится:
 - либо `+` (предоставляем право);
 - либо `-` (лишаем соответствующего права);
 - либо `=` (установить указанные права вместо имеющихся),
- Вместо `r`— символ, обозначающий соответствующее право:
 - `r` (чтение);
 - `w` (запись);
 - `x` (выполнение)
- ```
chmod a+x file_name
```
- ```
# chmod go-rw file_name
```
- ```
chmod ugo+rwX file_name
```