

Лекция 2.

# **Командная строка Linux**

## Лекция 2. Командная строка Linux

- Виды компьютерных интерфейсов
- Командные оболочки Linux
- Оболочка Bash
- Общая структура команды
- Программа Gnome Terminal
- Приглашение командной строки.
- Обычный пользователь и суперпользователь
- Команды su и sudo
- Команды man и info

- ❑ **Текстовый интерфейс пользователя (интерфейс командной строки, Command Line Interface, CLI).** Оператор вводит команды в текстовом виде, компьютер выполняет их и отвечает тоже текстовыми сообщениями.
- ❑ **Графический интерфейс пользователя с использованием мыши (графический интерфейс пользователя, Graphic User Interface, GUI).** На экране компьютера отображается графическое изображение некоторой системы, оператор выполняет команды, указывая мышью на специальные зоны изображения (экранные кнопки, ссылки и т.п.).
- ❑ **Графический интерфейс пользователя с использованием мультисенсорных панелей.** Отличается от предыдущего тем, что вместо мыши используется прямое указание пальцами или стилусом, а программа может обрабатывать действия сразу нескольких пальцев одновременно.
- ❑ **Голосовой интерфейс пользователя (Voice User Interface).** Пользователь отдает компьютеру команды голосом, а компьютер их интерпретирует и выполняет.

Достоинства **графических интерфейсов**:

- ❑ Такие интерфейсы существенно упрощают процесс изучения программы и для **пользователей-новичков являются предпочтительными**.
- ❑ Возможность реализации принципа **WYSIWYG (What You See Is What You Get** – что вижу, то и получаю).

Достоинства **текстовых интерфейсов**:

- ❑ Текстовые интерфейсы **более эффективны для опытных пользователей**.
- ❑ Интерфейсы командной строки предоставляют **широкие возможности для автоматизации**.
- ❑ Такие интерфейсы командной строки, как правило, **более лаконичны**.

Командный интерфейс в Linux и других UNIX-подобных ОС реализуется специальной программной, называемой **командной оболочкой**, **shell**, **командным интерпретатором** или **командным процессором**.

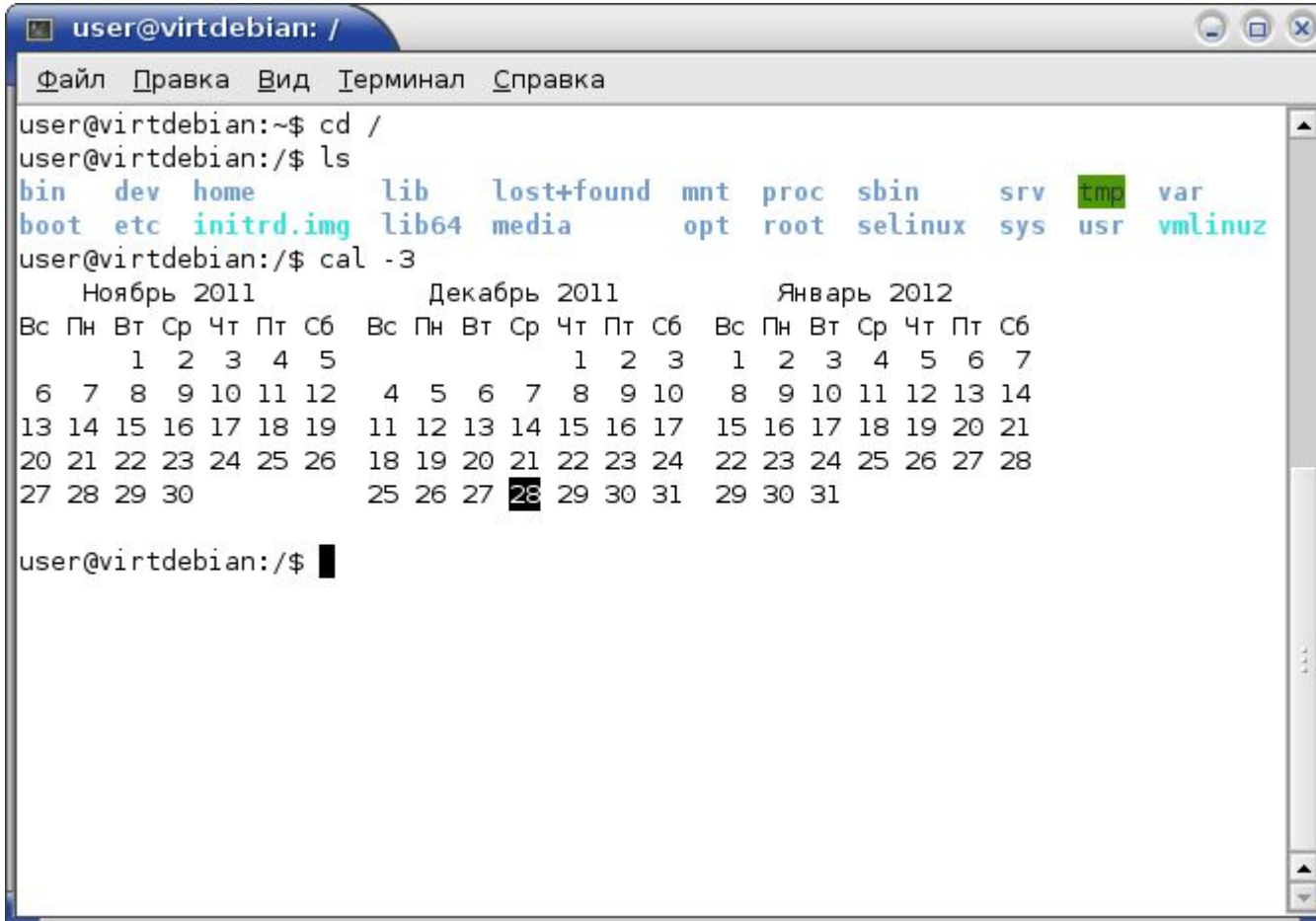
Исторически первой появилась оболочка Борна – **Bourne Shell** (**shell**, **sh**).

Несколько позже в рамках BSD Unix появилась оболочка **C-shell** (**csh**), созданная в университете Беркли. Она для написания сценариев использовала синтаксис, приближенный к языку C.

Все остальные оболочки делятся на **sh-совместимые** и **csh-совместимые**, в зависимости от того синтаксиса, который они поддерживают.

В настоящее время наибольшее распространение получила оболочка **bash** (**Bourne Again Shell** – Возрожденная оболочка (Борна)) – усовершенствованная версия оболочки Борна, созданная в рамках проекта **GNU**. Она используется по-умолчанию в большинстве дистрибутивов Linux.

Еще одной распространенной sh-совместимой оболочкой является **Z-shell** (**zsh**), содержащей широкие интерактивные возможности в некоторых моментах превосходящие возможности bash.



```
user@virtdebian: /
Файл Правка Вид Терминал Справка
user@virtdebian:~$ cd /
user@virtdebian:/$ ls
bin  dev  home  lib  lost+found  mnt  proc  sbin  srv  tmp  var
boot etc  initrd.img  lib64  media  opt  root  selinux  sys  usr  vmlinuz
user@virtdebian:/$ cal -3
      Ноябрь 2011          Декабрь 2011          Январь 2012
Вс Пн Вт Ср Чт Пт Сб   Вс Пн Вт Ср Чт Пт Сб   Вс Пн Вт Ср Чт Пт Сб
                1  2  3  4  5                1  2  3   1  2  3  4  5  6  7
 6  7  8  9 10 11 12   4  5  6  7  8  9 10   8  9 10 11 12 13 14
13 14 15 16 17 18 19  11 12 13 14 15 16 17  15 16 17 18 19 20 21
20 21 22 23 24 25 26  18 19 20 21 22 23 24  22 23 24 25 26 27 28
27 28 29 30           25 26 27 28 29 30 31  29 30 31
```

user@virtdebian:/\$ █

Каждая команда Linux соответствует следующему формату:

**команда** **-опции** **параметры**

где **опции** – это некоторые текстовые слова или символы, уточняющие логику работы команды, а **параметры**, как правило, определяют к какому объекту команду нужно применить.

Например: `ls -l /home/user/dir1/`

В строке команды сама команда, опции и параметры команды отделяются друг от друга разделителями, которыми по умолчанию являются символы **пробела** и **табуляции**.

Некоторые опции кроме активизации некоторого режима требуют задания вместе с ними параметра или параметров (**параметров опций**). Параметры опций являются обязательными.

`grep -f file1 -v file2`

Так же некоторые опции могут задаваться не одной буквой, а целым словом. В этом случае они предваряются не одним, а двумя дефисами.

`wc --help`

# Обычный пользователь и суперпользователь. Команды `su` и `sudo` (1)

В Linux, как и в любой другой Unix-среде, один пользователь выделяется особо. Это пользователь с номером 1, называемый **администратором**, **суперпользователем** или пользователем **root**. Суперпользователь отличается от остальных тем, что только суперпользователю разрешается выполнять ряд действий по настройке системы, например, задавать параметры сетевых устройств или подключать и отключать дисковые устройства. Часть команд, предназначенных для настройки системы, недоступны обычным пользователям.

В режим суперпользователя входят на короткое время для выполнения необходимых операций. Для временного входа в режим суперпользователя для выполнения привилегированных команд предназначены команды **su** и **sudo**.

Команда **su** позволяет временно подменить идентификатор текущего пользователя на идентификатор пользователя **root**. По умолчанию остальные настройки пользователя остаются неизменными. После ввода команды **su** система запрашивает пароль пользователя **root**. Для возврата в режим обычного пользователя после выполнения необходимых команд нужно ввести команду **exit**.



```
$ ifconfig
```

```
bash: ifconfig: команда не найдена
```

```
$ su
```

```
Пароль:
```

```
# ifconfig
```

```
eth0      Link encap:Ethernet  HWaddr 08:00:27:46:af:57
```

```
          inet addr:10.0.2.15  Bcast:10.0.2.255
```

```
          Mask:255.255.255.0
```

```
...
```

```
          RX bytes:732 (732.0 B)  TX bytes:732 (732.0 B)
```

```
# exit
```

```
exit
```

```
$
```

# Обычный пользователь и суперпользователь. Команды `su` и `sudo` (3)

Команда `sudo` отличается от команды `su` двумя моментами.

Во-первых, она по умолчанию **не переходит в отдельный режим** ввода привилегированных команд, а выполняет только одну такую команду. И эта команда указывается в качестве параметра в команде `sudo`:

```
$ sudo команда параметры
```

Если вам нужно выполнить несколько привилегированных команд, то можно перейти в специальный режим подобно тому, как это делает команда `su`:

```
$ sudo -i
```

Во-вторых, при выполнении команды требуется ввести не пароль пользователя `root`, а свой **собственный пароль**. Кроме этого команда `sudo` является более гибкой, чем команда `su` в том смысле, что можно разрешить пользователю выполнять не все, а только некоторые привилегированные команды. Сами настройки команды `sudo` содержатся в файле `/etc/sudoers`.

```
$ ifconfig
```

```
bash: ifconfig: команда не найдена
```

```
$ sudo ifconfig
```

```
[sudo] password for user:
```

```
eth0      Link encap:Ethernet  HWaddr 08:00:27:46:af:57
```

```
          inet addr:10.0.2.15  Bcast:10.0.2.255
```

```
          Mask:255.255.255.0
```

```
          inet6 addr: fe80::a00:27ff:fe46:af57/64 Scope:Link
```

```
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```

```
...
```

```
          RX bytes:732 (732.0 B)  TX bytes:732 (732.0 B)
```

```
$
```

Команда **man** (сокращение от англ. manual – руководство, справочник) предназначена для доступа к встроенным руководствам (так называемым **man-страницам**). Формат запуска команды:

```
$ man [раздел] имя [[раздел] имя ...]
```

Параметр **имя** задает имя команды, утилиты или функции для которой нужно отобразить руководство.

Все страницы разбиты на **разделы**, а в пределах одного раздела каждое имя является уникальным. Раздел задается в команде номером.

При отображении справки в команде **man** вы входите в специальный режим данной команды. Выйти из просмотра справочной страницы и вернуться в режим ввода команд можно нажав клавишу **q**.

Номер раздела	Описание раздела
1	Исполняемые программы или команды
2	Системные вызовы (функции, предоставляемые ядром)
3	Библиотечные вызовы (функции, предоставляемые программными библиотеками)
4	Специальные файлы устройств (обычно находящиеся в каталоге /dev)
5	Форматы стандартных файлов системы, например /etc/passwd
6	Игры и прочие несистемные программы
7	Разное (включает пакеты макросов и соглашения), например, man(7), groff(7)
8	Команды администрирования системы (обычно, запускаемые только суперпользователем)
9	Процедуры ядра (раздел для системных программистов)