ИУ-10
Системное
Программное
Обеспечение

Администрирование Linux
Взаимодействие с оболочкой Bash

Москва, 2022

На этом уроке

1. Узнаем, как подключиться к ОС, используя протокол удалённого управления SSH.
2. Изучим, как работает навигация по файловой системе
3. Познакомимся с утилитами работы с папками
4. Познакомимся с утилитами работы с файлами
5. Научимся искать информацию в документации с man

Содержание

Знакомство с Bash

Как мы узнали на первом уроке Linux может быть использован как в десктопной версии, так и в серверной. В десктопной версии используется графический интерфейс, и вы можете работать в окружении рабочего стола, схожего с тем, что вы используете в Windows. Так как основное достоинство Linux — применение в серверном администрировании, администраторы для работы используют интерфейс командной строки (Command Line Interface - CLI) или консоль, только в нем доступен весь функционал Linux.

Консоль позволяет нам вводить текстовые команды, получать ответ системы на них в текстовом виде и таким образом управлять операционной системой. **Важно!** *Оболочка Bash - это интерпретатор команд, который используется по умолчанию, то есть любая вводимая команда будет обрабатываться Bash. Оболочка предоставляет набор полезных функций, которые мы изучим на уроке.*

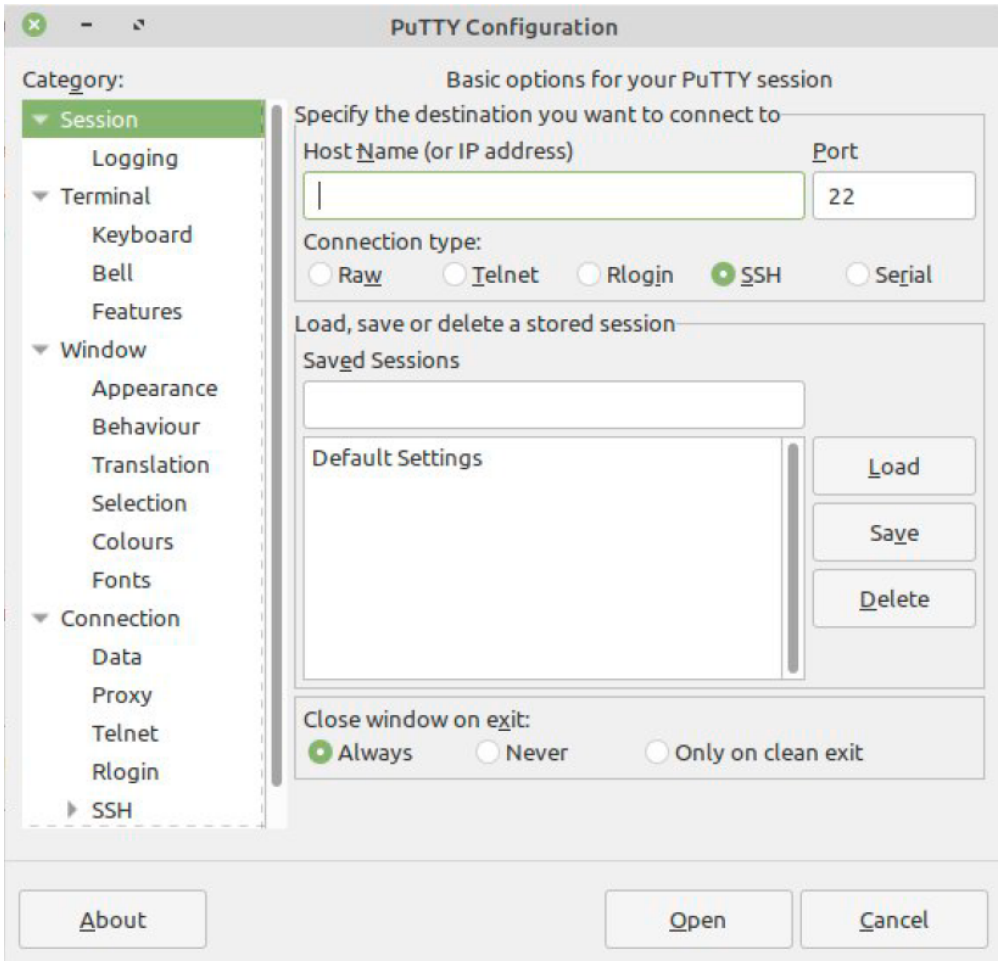
После загрузки ОС нам становится доступно семь терминалов, переключаться между которыми можно, используя комбинацию клавиш.

1. В случае с физической машиной: **Ctrl + Alt + F(1–7)**, где клавиши F1–F7 — номера виртуальных терминалов

2. В случае с виртуальной машиной (под VirtualBox) переключение между терминалами будет осуществляться при помощи комбинации **host_key + F(1–7)**, где host_key в большинстве случаев — клавиша правый Ctrl.

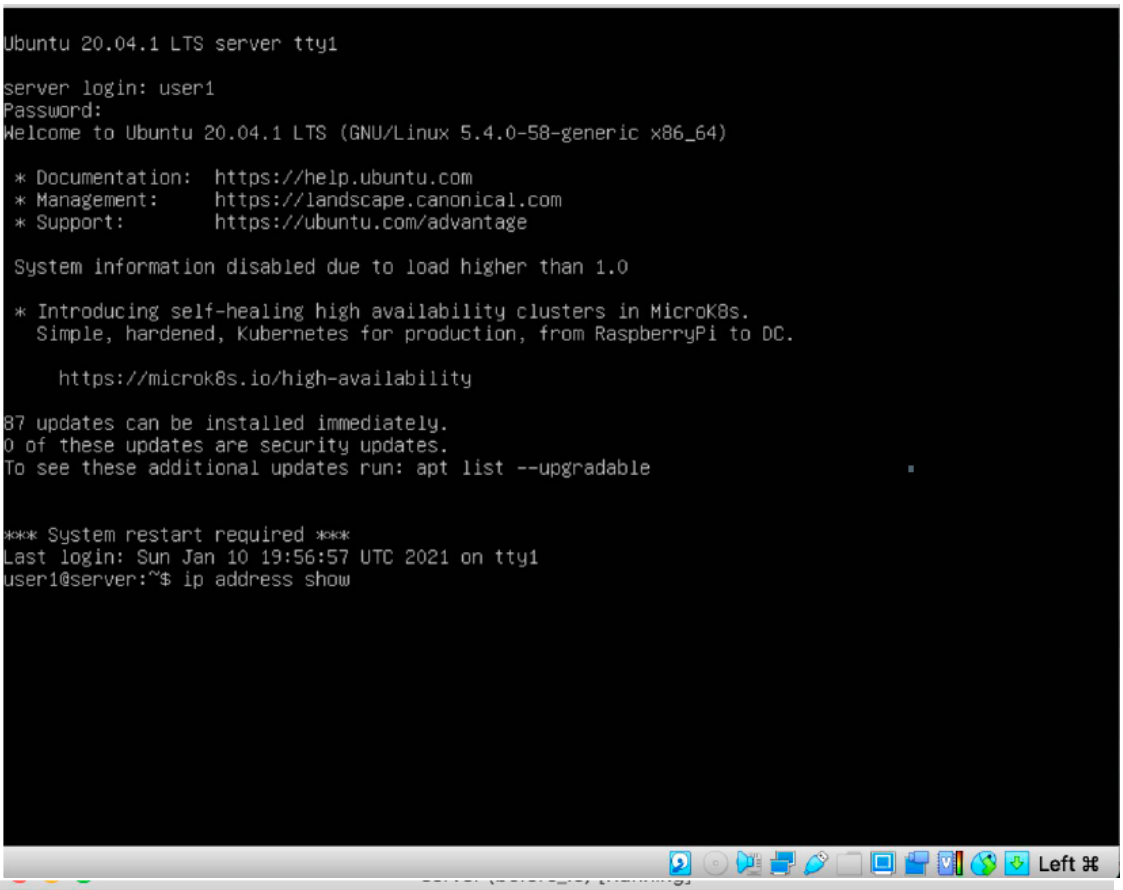
Для последующей работы будем использовать подключение к серверу через протокол SSH. Для этого нам понадобится установить на свой компьютер клиент SSH. Самый распространенный клиент для Windows — PuTTY, в последних версиях ОС Windows существует встроенный клиент. Если вы работаете из-под Linux или macOS, то для подключения к удалённому серверу можно использовать предустановленный в системе клиент.

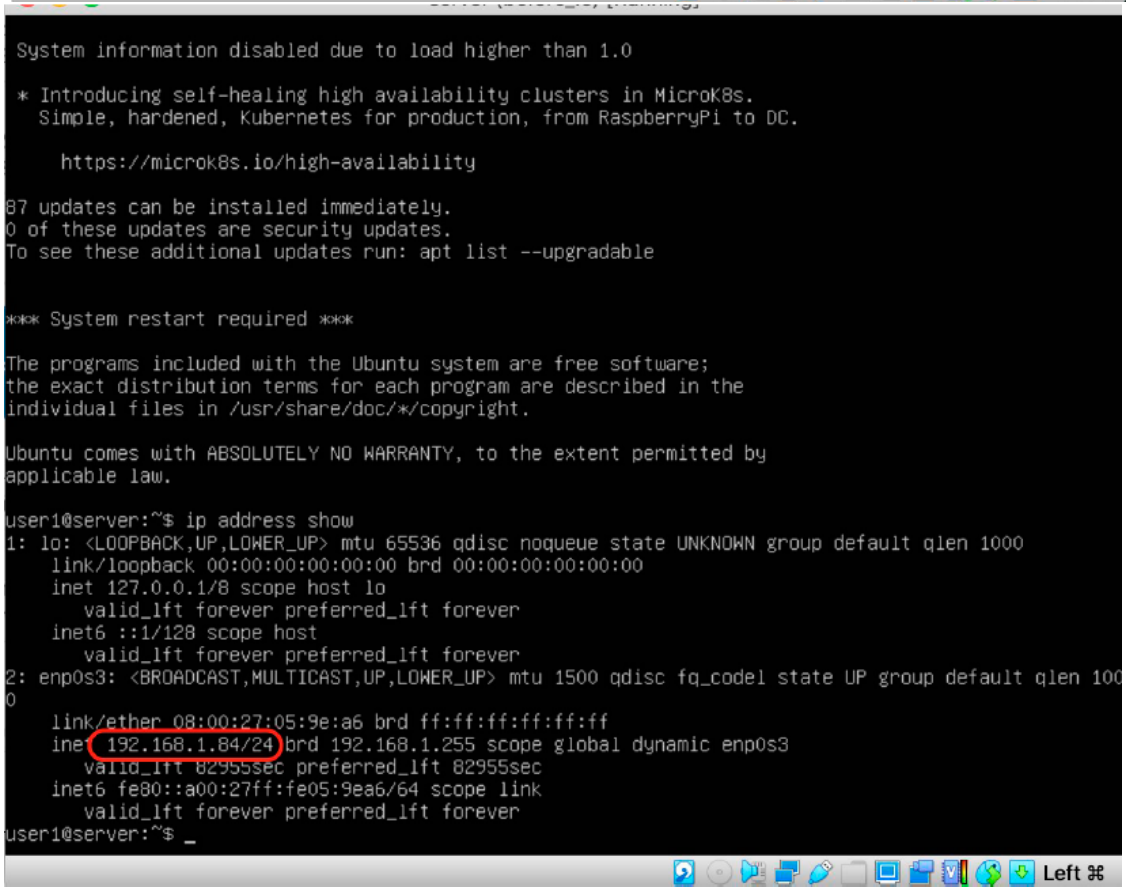
1. **Подключение к серверу, используя PuTTY**. Запускаем программу. В поле Host Name (or IP address) вводим IP-адрес нашей виртуальной машины или сервера. Далее нажимаем кнопку Open, вводим логин и пароль. **Важно:** *при вводе пароля символы не отображаются.*



2. Подключение к серверу, используя terminal/iTerm. Запускаем программу и в окне вводим команду `SSH your_user@ip_server`, далее вводим пароль и получаем приглашение в командную строку.

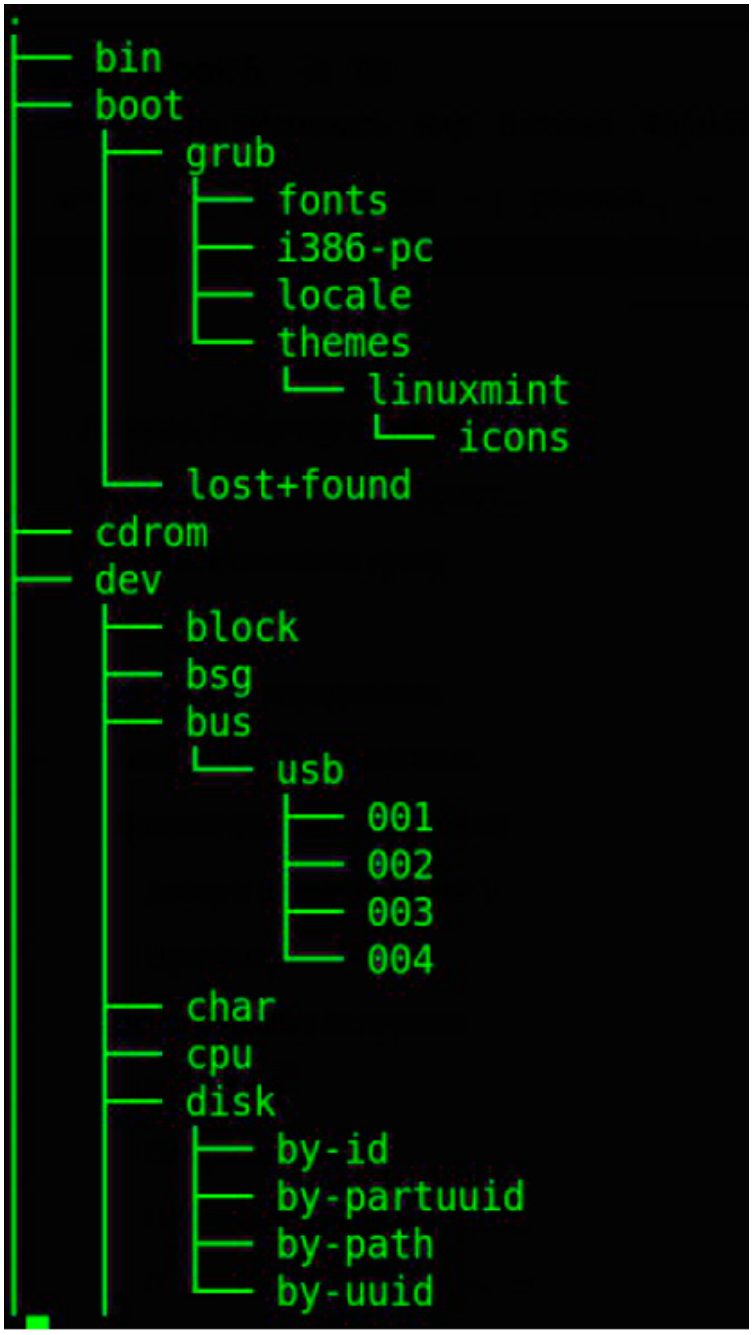
Примечание. *Найти IP адрес сервера можно с помощью команды* ***ip address show****, как показано ниже на рисунке. Команду требуется ввести после логина через терминал VirtualBox.*


Ubuntu 20.04.1 LTS server tty1
server login: user1
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-58-generic x86_64)
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage
System information disabled due to load higher than 1.0
* Introducing self-healing high availability clusters in MicroK8s.
Simple, hardened, Kubernetes for production, from RaspberryPi to DC.
https://microk8s.io/high-availability
87 updates can be installed immediately.
0 of these updates are security updates.
To see these additional updates run: apt list --upgradable
*** System restart required ***
Last login: Sun Jan 10 19:56:57 UTC 2021 on tty1
user1@server:~$ ip address show
Left ⌘


System information disabled due to load higher than 1.0
* Introducing self-healing high availability clusters in MicroK8s.
Simple, hardened, Kubernetes for production, from RaspberryPi to DC.
https://microk8s.io/high-availability
87 updates can be installed immediately.
0 of these updates are security updates.
To see these additional updates run: apt list --upgradable
*** System restart required ***
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
user1@server:~$ ip address show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
inet 127.0.0.1/8 scope host lo
valid_lft forever preferred_lft forever
inet6 ::1/128 scope host
valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 100
0
link/ether 08:00:27:05:9e:a6 brd ff:ff:ff:ff:ff:ff
inet 192.168.1.84/24 brd 192.168.1.255 scope global dynamic enp0s3
valid_lft 82955sec preferred_lft 82955sec
inet6 fe80::a00:27ff:fe05:9ea6/64 scope link
valid_lft forever preferred_lft forever
user1@server:~$ _
Left ⌘

Навигация по файловой системе и основные операции с файлами и каталогами

Хранение файлов в Linux организовано в виде древовидной структуры. В ней есть корневой каталог, от которого «растут» все остальные каталоги и файлы. Корневой каталог носит название «/» (root), при этом важно не путать его с домашним каталогом суперпользователя /root. Утилита `tree` выведет на экран иерархию на экран.



В оконном интерфейсе Windows навигация по системе интуитивно понятна, так как используются визуальные образы и нажатия клавиш мыши. В интерфейсе командной строки подобное представление отсутствует, по-

этому для успешной навигации по файловой системе в первую очередь необходимо разобраться в понятии пути до файла или каталога.

Путь до файла - это набор символов, показывающий расположение файла или каталога (папки) в файловой системе. По сути путь указывает нам, в какой точке дерева в данный момент мы находимся. Путь может быть полным (абсолютным) — это путь, который указывает на одно и то же место в файловой системе, вне зависимости от текущего рабочего каталога. **Примечание!** *Полный путь всегда начинается с корня - "/"(слэш), например /usr/local/bin/*. Путь может быть также относительным — это путь по отношению к текущему рабочему каталогу пользователя (то есть по отношению к директории в которой пользователь сейчас находится). В отличие от абсолютного пути, относительный меняется, если мы меняем нашу текущую директорию. **Примечание!** *Относительный путь никогда не начинается с корня - "/"(слэш), например local/bin/*.

Команда `pwd — print working directory` (показать рабочий каталог) — это первая команда, с которой мы познакомимся. Она покажет текущий каталог (каталог, в котором мы сейчас находимся), при этом покажет полный путь. Команда необходима, чтобы понять, в каком месте дерева файловой системы мы находимся.

```
user@server:~$ pwd
/home/user
user@server:~$
```

Перемещение между каталогами осуществляется при помощи команды **cd — change directory**. Данная команда позволит нам сменить текущую директорию, используя полный или относительный путь. Например:

- используем полный путь: `cd /var/log` ;

```
user@server:~$ pwd
/home/user
user@server:/var/log$ cd /var/log/
user@server:/var/log$ pwd
/var/log
```

- используем относительный путь: `cd journal/` ;

```
user@server:/var/log$ cd journal/
user@server:/var/log/journal$ pwd
/var/log/journal
```

- для перехода на уровень выше можно использовать `cd ..`

```
user@server:/var/log/journal$ cd ..
user@server:/var/log$ pwd
/var/log
```

- быстро вернуться в домашний каталог: `cd ~` или просто `cd`

```
user@server:/var/log$ cd
user@server:~$ pwd
/home/user1
user@server:~$
```

- Переход в предыдущую директорию: `cd -`

```
user@server:~$cd -
user@server:/var/log$
```

Полезные функции Bash

При работе в командном интерпретаторе bash есть несколько комбинаций клавиш, позволяющих существенно облегчить работу. Например, в параметрах команд можно не набирать имена существующих файлов или каталогов целиком, достаточно набрать их начало и нажать клавишу табуляции.

```
user@server:~$ cd f<TAB>
```

Дальше bash выполнит поиск среди файлов, начинающихся на f, и в случае однозначного совпадения имя будет сразу дополнено:

```
user@server:~$ cd files/
```

Если найденных вариантов несколько, они все будут показаны:

```
user@server:~/files$ cd 20<TAB>
2000/ 2002/ 2004/ 2006/ 2008/ 2010/ 2012/ 2014/
2001/ 2003/ 2005/ 2007/ 2009/ 2011/ 2013/
```

Можно продолжить набор, сократив количество вариантов:

```
user@server:~/files$ cd 201<TAB>
2010/ 2011/ 2012/ 2013/ 2014/
```

Аналогично работает автодополнение по именам команд:

```
user@server:~/files$ ma<TAB
```

```
mailmail3    man          manifes      man-recode

make-bcache  mandb        manpat       mapfil       mawk
```

Кроме автодополнения существует возможность повторять ранее набранные команды с помощью стрелочек вверх-вниз на клавиатуре. Это называется историей или стеком команд. Команда history выведет на экран все ранее вводимые команды и с помощью комбинации !<номер> команду можно повторить.

```
user@server:~$ history
1  cd /var/log/
2  pwd
3  cd /var/log/
4  pwd
5  cd journal/
6  pwd
7  cd
8  pwd
9  cd /var/log/
10  ls
11 clear
user1@server:~$ !3
cd /var/log/
user@server:/var/log$
```

Можно выполнять поиск по истории команд с помощью комбинации Ctrl-R (поиск назад).Введите строку поиска и если в истории команд была команда с такой подстрокой, она будет найдена и предложена к выполнению.

```
(reverse-i-search)‘cd’: cd /var/log/
```

Используйте Enter для немедленного выполнения команды или <ESC>, чтобы найденную команду редактировать.

Кроме того, существуют команды для быстрого перемещения по командной строке и быстрого удаления текста:

- Ctrl-A — в начало строки.
- Ctrl-E — в конец строки.
- Alt-F — на слово вперед.
- Alt-B — на слово назад.

- Ctrl-U — удалить в строке все символы от текущей позиции до начала строки.
- Ctrl-K — удалить в строке все символы от текущей позиции до конца строки.

Примечание! *Заглавные буквы показаны только для наглядности, комбинации работают со строчными символами.*

Взаимодействие с файлами и каталогами

Просмотр и создание

Просмотреть содержимое каталога нам поможет команда **ls <путь к каталогу>**. Если путь к каталогу не указан, то будет показано содержимое папки, в которой находится пользователь в данный момент. У команды есть ряд полезных ключей (параметров):

1. `ls -l` покажет подробный список содержимого, сюда будут включены дата изменения, владелец и группа владельца, права и другие свойства файлов или каталогов в директории.
2. `ls -a` покажет скрытые файлы и каталоги. В Unix-подобных системах такие файлы и каталоги начинаются с точки. Этот параметр очень часто используют в сочетании с параметром **-l**, например `ls -al /home/user`.

```
user@server:~$ ls -al
total 24
drwxr-xr-x 2 user user 4096 Dec 30 14:55 .

drwxr-xr-x 6 root root 4096 Dec 14 18:04 ..

-rw------- 1 user user 153 Dec 24 17:11 .bash_history

-rw-r--r-- 1 user user 220 Feb 25 2020 .bash_logout

-rw-r--r-- 1 user user 3771 Feb 25 2020 .bashrc
-rw-rw-r-- 1 user user    0 Dec 30 14:55

-rw-rw-r-- 1 user user    0 Dec 30

-rw-r--r-- 1 user user 807 Feb 25 2020 .profile
```

```
-rw-r--r-- 1 user user    0 Dec 13 18:42 .
  sudo_as_admin_successful
```

Для создания файлов в ОС Linux есть несколько способов:

1. Используя утилиту **touch** — она создаст пустой файл.
2. Используя перенаправление потока вывода, например, с помощью утилит **cat** или **echo** (рассмотрим их на следующем уроке).
3. Используя текстовый редактор.

Создание каталогов — команда **mkdir** (в некоторых дистрибутивах **md**, make directory). Например, `mkdir /home/user/dir1` создаст каталог с именем dir1 в домашнем каталоге пользователя user.

Бывают случаи, когда нам необходимо создать каталог и вложенные подкаталоги, для решения этой задачи используют параметр **-p (parents)**, например, `mkdir -p /home/user/dir1/dir2/` создаст в домашнем каталоге пользователя user каталог dir1 и вложенный подкаталог dir2.

```
user@server:~$ mkdir /home/user/dir1/dir2/
mkdir: cannot create directory ‘/home/user/dir1/dir2’/: No such
    file or
directory
user@server:~$ mkdir -p /home/user/dir1/dir2/
user@server:~$ tree
.
├── dir1
│   └── dir2
├── file1
└── file2
```

Действия с существующими объектами

Копирование файлов или каталогов — команда **cp (copy): cp file1 file2**. При операции копирования можно использовать как полный, так и относительный путь. Например:

- `cp /usr/local/etc/file /tmp/` скопирует файл с именем file из каталога /usr/local/etc/ в каталог /tmp, сохранив название файла.
- `cp /usr/local/etc/file /tmp/file1` скопирует файл с именем file из каталога /usr/local/etc/ в каталог /tmp, изменив имя файла на file1.

- `cp /usr/local/etc/file .` копирует файл из каталога /usr/local/etc/ в текущий каталог.
- `cp file file1` создаст копию файла в текущем каталоге.
- Копирование директорий происходит немного иначе, поскольку может содержать поддиректории, поэтому необходимо использовать параметр **-r (рекурсивно)**, например, `cp -r /dir1 .` скопирует каталог /dir1 в текущую директорию.

```
user1@server:~$ cp dir1/ copy_dir1
cp: -r not specified; omitting directory 'dir1/' user1@server:~
   $ cp -r  dir1/ copy_dir1
user1@server:~$ tree
.
├── copy_dir1
│   └── dir2
├── dir1
│   └── dir2
├── file1
└── file2
```

Перемещение файлов или каталогов — команда **mv (move)**. `mv /home/user/file /home/user1/file` переместит файл из каталога /home/user в каталог /home/user1. Команда **mv**, применённая к файлу или каталогу в текущей директории, переименует файл или каталог. Например: `mv file1 file2`, `mv dir1 dir2`. **Примечание!** *Относительно каталогов операция mv не требует параметра -r, поскольку никак не воздействует на поддиректории.*

Удаление файлов или каталогов — команда **rm (remove)**. Например, `rm file1` удалит файл. Для удаления каталогов необходимо использовать параметр **-rf** (recursive, forced) — удалить со всем содержимым, не спрашивая подтверждения.

Внимание! *Операция удаления — необратимое действие. Debian-подобные дистрибутивыне спрашивают подтверждения действия. Ошибочное удаление файлов или каталогов может привести к неработоспособности системы.*

Функция globbing в Bash

globbing - это функция оболочки, которая помогает в сопоставлении имен файлов. Его не следует путать с регулярными выражениями, тема, о которой мы поговорим позже, и которая поможет найти текстовые шаблоны.

Давайте разберем несколько примеров, как пользоваться globbing.

```
kglushen@server:~$ ls -l /etc/host*
-rw-r--r-- 1 root root 92 Dec 5 2019 /etc/host.conf

-rw-r--r-- 1 root root   7 Jan 28 14:22 /etc/hostname

-rw-r--r-- 1 root root 221 Jan 28 14:22 /etc/hosts

-rw-r--r-- 1 root root 411 Jan 28 14:23 /etc/hosts.allow
-rw-r--r-- 1 root root 711 Jan 28 14:23 /etc/hosts.deny
```

Можно видеть, что были показаны все файлы, имена которых начинаются с host, за которыми следует что угодно (символ *). С помощью "?" шаблон будет подставлять один любой символ, а в квадратных скобках задается перечень ожидаемых символов, или символов, которые нужно исключить (для этого применяется восклицательный знак).

```
kglushen@server:~$ touch cat kglushen@server:~$ touch hat
   kglushen@server:~$ touch rat
kglushen@server:~$ touch canat
kglushen@server:~$ ls -l ?at
-rw-rw-r-- 1 kglushen kglushen 0 Jan 30 13:24 cat
-rw-rw-r-- 1 kglushen kglushen 0 Jan 30 13:24 hat
-rw-rw-r-- 1 kglushen kglushen 0 Jan 30 13:24 rat

kglushen@server:~$ ls -l [ch]at
-rw-rw-r-- 1 kglushen kglushen 0 Jan 30 13:24 cat
-rw-rw-r-- 1 kglushen kglushen 0 Jan 30 13:24 hat
   kglushen@server:~$ ls -l [!ch]at
-rw-rw-r-- 1 kglushen kglushen 0 Jan 30 13:24 rat
```

С помощью globbing в шаблоне можно указывать диапазон чисел.

```
kglushen@server:~$ touch {1..5}_file
kglushen@server:~$ ls -l *_file
-rw-rw-r-- 1 kglushen kglushen 0 Jan 30 13:35 1_file
-rw-rw-r-- 1 kglushen kglushen 0 Jan 30 13:35 2_file
-rw-rw-r-- 1 kglushen kglushen 0 Jan 30 13:35 3_file
-rw-rw-r-- 1 kglushen kglushen 0 Jan 30 13:35 4_file
-rw-rw-r-- 1 kglushen kglushen 0 Jan 30 13:35 5_file
kglushen@server:~$ ls -l [1-3]_file
-rw-rw-r-- 1 kglushen kglushen 0 Jan 30 13:35 1_file
-rw-rw-r-- 1 kglushen kglushen 0 Jan 30 13:35 2_file
-rw-rw-r-- 1 kglushen kglushen 0 Jan 30 13:35 3_file
```

Архиваторы и компрессоры в Linux

В Windows вы наверняка использовали программы-архиваторы (rar и zip), которые позволяют упаковать каталог с файлами в один сжатый архивный файл. В Linux тоже есть версии этих архиваторов, которыми можно распаковать архивы, созданные пользователями Windows. Также есть графические программы-оболочки, облегчающие пользователям работу с архивами. Сейчас мы поговорим о других программах из мира Linux, которые появились задолго до своих Windows-собратьев. Чем же они уникальны?

Во-первых, есть различие в терминах. Архиваторами в Linux и Windows называют программы, немного отличающиеся по назначению. В Linux архиваторы выполняют следующие задачи:

- создавать архивы как объединения заданных файлов в виде одного
- добавлять в архив или извлекать из него отдельные файлы или группы файлов.

В Windows архиваторы делают то же что в Linux, плюс дополнительно занимаются сжатием архивов. В Linux сжатием занимаются отдельные утилиты, называемые компрессорами. Такое разделение функций является характерным для философии unix way: каждой задаче — свой продвинутый инструмент, плюс комбинация разных инструментов для решения новых задач. Подход оказался оправданным, он позволяет использовать новые, более мощные компрессоры в сочетании со старыми архиваторами. Рассмотрим конкретные примеры программ для архивации и сжатия файлов.

Утилита tar (Tape ARchiver)

Судя по названию, предназначалась для создания архивах на магнитных лентах, но также прекрасно работает с архивами в виде файлов. Имеет три основных варианта использования:

Создание архивов с tar

`tar опции arc_file file_or_dirname ...`
Опции имеют следующее значение:

- c — создать архив.
- z — сжимать файл архива с помощью gzip .
- j — сжимать файл архива с помощью bzip2 .

- v — (verbose) выдает в терминал имя добавляемого файла. Когда файлов много, обычно эту опцию не указывают и tar работает молча.
- f — после этой опции указывают имя архивного файла.
- file_or_dirname — список архивируемых файлов или каталогов. Каталоги архивируются рекурсивно вместе со всем содержимым.

Пример:

```
tar cf foo.tar ~/foo
```

Просмотр оглавления архива

```
tar tvf arc_file
```

- t — прочитать оглавление архива. Остальные опции как в случае создания архива. Пример:

```
tar tvf foo.tar
```

Извлечь файлы из архива

```
tar xvf arc_file [filename]
```

- x — извлечь файл(ы). Если последний параметр не указывать, будут извлечены все файлы из архива в текущий каталог. Имена извлекаемых файлов надо указывать, так как они показаны в листинге оглавления архива (с тем же относительным путём).

Компрессоры

В Linux семейство компрессоров представлено сразу несколькими программами: compress, gzip, bzip2. Они используют разные алгоритмы сжатия и в разной степени эффективны. Каждый компрессор включает в себя три утилиты: утилита сжатия (compress, gzip, bzip2), утилита декомпрессии (uncompress, gunzip, bunzip2,) и ещё одна утилита декомпрессии, которая по умолчанию выдаёт результат в стандартный вывод. Утилиты разных компрессоров имеют унифицированный интерфейс, так что нет необходимости запоминать ключи для каждого компрессора отдельно.

gzip

gzip — это GNU-версия программы compress, созданная, чтобы избежать патентных ограничений. Ключи у gzip такие же, как у compress. Дополнительно есть полезная опция, задающая с помощью числового значения от 1 до 9 степень сжатия. Это позволяет управлять соотношением сжатие/время в зависимости от потребностей, например можно задать -5 для больших файлов и выиграть на времени исполнения при достаточной степени сжатия. Суффиксы сжатых файлов, которые добавляет gzip — .gz. Для декомпрессии есть утилиты gunzip и gzcat, аналогичные uncompress и zcat. gunzip ищет файлы с суффиксами .gz и .Z, .z, tgz, taz.

```
gzip [ к  л  ю  ч  и ] [ ф  а  й  л ]...
```

- -c — направляет результат в стандартный вывод, исходные файлы не меняются. Обычно для использования совместно с архиватором.
- -d — выполнить декомпрессию указанных файлов.

Без ключей gzip выполняет сжатие каждого файла из заданного списка и записывает результат в новый файл, добавляя к имени суффикс .gz. Исходные несжатые файлы удаляются.

Пример использования совместно с архиватором:

```
tar cf - ./data | gzip -c > data.tar.gz
```

gunzip

Декомпрессия с gunzip:

```
gunzip [ ф  а  й  л ]...
```

Выполняет декомпрессию файлов из списка, если надо, добавляя к именам суффикс gz. Разархивированный файл записывается с именем без .gz. Исходный сжатый файл удаляется.

zcat

Декомпрессия с выдачей в стандартный вывод zcat:

```
zcat [ ф  а  й  л ]...
```

Этот вариант обычно используют совместно с архиватором для декомпрессии сжатых архивов:

```
zcat data.tar.gz | tar xf -
```

Ещё один популярный компрессор — bzip2. Превосходит на большинстве файлов gzip по эффективности сжатия, но проигрывает в скорости. Опции такие же, как у gzip. Расширения архивов — .bz2.

Встроенная документация

Ранее при запуске команд использовался набор параметров или так называемых ключей, и при использовании каждого ключа был получен различный результат. Подробная документация в Linux поставляется вместе с утилитами и доступна каждому пользователю.

–help

Самым простым способом для поиска является выполнение команд с параметром –help:

```
user@server:~$ mkdir --help
Usage: mkdir [OPTION]... DIRECTORY...
Create the DIRECTORY(ies), if they do not already exist.

Mandatory arguments to long options are mandatory for short
   options too.
-m, --mode=MODE   set file mode (as in chmod), not a=rwx -
   umask
-p, --parents     no error if existing, make parent directories
    as needed
-v, --verbose     print a message for each created directory
-Z                   set SELinux security context of each
   created directory to the default type
    --context[=CTX]  like -Z, or if CTX is specified then set
   the SELinux or SMACK security context to CTX
    --help                      p and exit
    --version  output version information and exit
GNU coreutils online help: <https://www.gnu.org/software/
   coreutils/> Full documentation at: <https://www.gnu.org/
   software/coreutils/   ir> or available locally via: info '(
   coreutils) mkdir invocation'
```

На экране будет выведена краткая информация о возможных ключах и их влиянии на команды.

Man

В случае необходимости более подробной информации в Linux существует встроенная система документации, обратившись к которой, всегда можно найти название нужной утилиты, описание параметра команды или пример ее использования. Например, если требуется подробное описание команды cp, используем команду man (от MANual).

```
user@server:~$ man cp
```

Справка по командам обычно занимает несколько страниц. Для перехода к следующей странице используйте клавишу <PgDn>, обратно — <PgUp>. Если эти клавиши не срабатывают, используйте для перехода пробел и клавишу u соответственно. Для выхода из man служит клавиша q. Есть еще пара полезных клавиш: g — переход в начало, G — в конец документа. Для поиска вперед используйте /чтоискать, для обратного поиска — ?что искать, повторить поиск в том же направлении — n.

Теперь пара слов о стандартных разделах справочной страницы.

NAME — краткое описание команды.

```
NAME
    cp - copy files and
```

Раздел SYNOPSYS описывает различные варианты синтаксиса. В этом разделе могут использоваться специальные обозначения. Например, если опции или параметры заключены в квадратные скобки, это означает, что они не обязательны для использования. Троеточие после параметра говорит, что он может повторяться многократно, как в случае, когда несколько файлов копируются в один каталог назначения.

```
SYNOPSIS
    cp [OPTION]... [-T] SOURCE DEST
    cp [OPTION]... SOURCE... DIRECTORY
    cp [OPTION]... -t DIRECTORY
```

В разделе DESCRIPTION описываются опции (иногда их называют ключами) программы. Бывает, что в описании через запятую перечислены 2 опции, например -v, –verbose. В таких случаях мы имеем дело с разными стилями одной и той же опции. -v — классический вариант, описанный в стандарте POSIX. –verbose — опция в стиле GNU. Можно использовать то, что больше нравится.

```
    -v, --verbose
        explain what is being done
```

Еще один полезный раздел обычно находится в конце справочной странице, он называется SEE ALSO. Как правило, здесь приводят ссылки на

другие утилиты с похожим функционалом.

```
SEE ALSO
    Full documentation at: <http://www.gnu.org/software/
   coreutils/cp> or available locally via: info '(coreutils) cp
   invocation'
```

Apropos

Кроме man есть еще несколько полезных справочных утилит. Если мы не знаем точного имени утилиты, можно попробовать найти ее с помощью apropos, которая выполнит поиск по ключевым словам в описании утилит. Например найдем как называется утилита, которая меняет пароль пользователя:

```
user@user-virtual-machine:~$
apg (1)                  - generates several random passwords
chage (1)                - change user password expiry
   information
chgpasswd (8)            - update group passwords in batch mode
chpasswd (8)             - copy with locking the given file to
   the
cpgr (8) or g.. cppw (8) or g..         - copy with locking the
    given file to the
cracklib-check (8)       - Check passwords using
create-cracklib-dict (8) - Check passwords using libcrack2
```

Практическое задание

1. **Навигация по файловой системе.**
   (a) Переместитесь в каталог /home
   (b) Просмотреть содержимое каталога /home и всех папок внутри
   (c) С помощью –help найти, как выполнить задание (b) одной командой (использовать команду ls)
2. **Управление файлами и каталогами.**
   (a) Создать несколько пустых файлов. Создайте директорию, переместите файлы туда.
   (b) Скопируйте файл /var/log/syslog в созданную папку

(c) Сделайте из созданной папки tar архив с произвольным названием

(d) Удалите архив.

(e) Выведите историю команд

3. С помощью man понять, как искать файлы по имени (утилита find). *В папке /etc найти расположение всех файлов с расширением .yaml.

4. *С помощью globbing в одну команду создать перечень папок по годам от 2015 до 2021, внутри которых будут находиться папки по месяцам от 1 до 12.

Глоссарий

Терминал — интерфейс взаимодействия между пользователем и операционной системой.

SSH-клиент — программное обеспечение, позволяющее подключиться к серверу, используя протокол SSH. Рекомендуемые из-за простоты установки SSH-клиенты: в Windows можно использовать PowerShell с 10 -й версии или PuTTY - в более старых версиях, в macOS — iTerm2, в Linux — встроенное приложение GNOME Terminal.

Аутентификация — процедура проверки подлинности, например сравнением введённого пароля пользователя с паролем, сохранённым в базе данных паролей.

Файловая система — часть операционной системы, которая обеспечивает чтение и запись файлов на дисковых носителях информации. Файловая система устанавливает физическую и логическую структуру файлов, правила их создания и управления ими, а также сопутствующие данные файла и идентификацию. Конкретная файловая система определяет размер имени файла и максимально возможный размер файла.

Файл — именованная область данных на носителе информации.

Каталог — объект файловой системы, упрощающий организацию файлов. В Linux реализован как специальный файл, где регистрируется информация о других файлах и каталогах файловой системы.

Дополнительные материалы

1. Авторизация по ключу

2. Настройка авторизации по ключу, используя программу PuTTY

3. Небольшой обзор редакторов

4. https://ru.wikipedia.org/wiki/Философия_UNIX
5. `https://losst.ru/tsikly-bash`

Используемые источники

1. https://ru.wikipedia.org/wiki/Философия_UNIX
2. Статья, посвящённая работе с терминалом
3. Статья, посвящённая текстовым редакторам vi/Vim
4. Костромин В. Linux для пользователя