

Команда pwd

pwd нужна, чтобы узнать путь к текущему рабочему каталогу (папке), в котором вы находитесь от корневого каталога. Команда вернёт абсолютный (полный) путь, который по сути является путём всех каталогов, начинающийся с косой черты (/). Примером абсолютного пути является /home/username:

```
den@vm355614:~$ pwd
/home/den
den@vm355614:~$
```

```
den@vm355614:/var/log/apt$ pwd
/var/log/apt
den@vm355614:/var/log/apt$
```

Команда cd

Для навигации по файлам и каталогам Linux используется команда cd. Она требует либо полный (абсолютный) путь, либо, в зависимости от текущего рабочего каталога (абсолютный).

```
den@vm355614:/var/log$ cd apt
den@vm355614:/var/log/apt$
```

Также есть несколько шорткодов для более быстрой навигации:

cd .. (с двумя точками), чтобы переместиться на один каталог вверх cd, чтобы перейти прямо в домашнюю папку cd- (с дефисом), чтобы перейти к предыдущему каталогу Также стоит отметить, что оболочка Linux чувствительна к регистру. Важно точно вводить имена каталогов.

```
den@vm355614:/var/log/apt$ cd ..
den@vm355614:/var/log$
```

```
den@vm355614:/root$ cd
den@vm355614:~$ pwd
/home/den
den@vm355614:~$
```

Команда ls

Команда ls используется для просмотра содержимого каталога. По умолчанию эта команда отобразит содержимое вашего текущего рабочего каталога.

```
den@vm355614:/$ ls
bin  dev  home  lib32  libx32  media  opt  root  sbin  swapfile  tmp  var
boot  etc  lib   lib64  lost+found  mnt  proc  run  srv  sys      usr
den@vm355614:/$
```

Если вы хотите просмотреть содержимое других каталогов, введите `ls`, а затем путь к каталогу. Например, введите `ls /home/username/Documents` для просмотра содержимого в `Documents`.

Варианты использования команды Linux `ls`:

`ls -R` также выведет список всех файлов в подкаталогах `ls -a` покажет скрытые файлы `ls -al` выведет список файлов и каталогов с подробной информацией, такой как разрешения, размер, владелец и т. д.

```
den@vm355614:/var/log$ ls -a -l
total 736
drwxrwxr-x  7 root  syslog      4096 Nov 22 00:00 .
drwxr-xr-x 11 root  root        4096 Apr 27 2020 ..
drwxr-xr-x  2 root  root        4096 Nov 21 00:00 apt
-rw-r----- 1 syslog adm       127108 Nov 22 20:13 auth.log
-rw-r----- 1 syslog adm       54862 Nov 20 23:17 auth.log.1
-rw-r----- 1 syslog adm        232 Nov 19 20:09 auth.log.2.gz
-rw-----  1 root  root          0 Nov 19 20:09 boot.log
-rw-----  1 root  root       12998 Nov 19 20:09 boot.log.1
-rw-rw----  1 root  utmp     100224 Nov 22 20:07 btmp
-rw-r--r--  1 syslog adm       68195 Nov 19 20:09 cloud-init.log
-rw-r----- 1 root  adm        5355 Nov 19 20:09 cloud-init-output.log
-rw-r--r--  1 root  root          0 Nov  1 02:41 debug
drwxr-xr-x  2 root  root        4096 Apr  8 2020 dist-upgrade
-rw-r--r--  1 root  adm       52876 Nov 19 20:09 dmesg
-rw-r--r--  1 root  adm       52952 Nov  1 02:40 dmesg.0
-rw-r--r--  1 root  adm      14413 Nov  1 02:34 dmesg.1.gz
-rw-r--r--  1 root  root          0 Nov 21 00:00 dpkg.log
-rw-r--r--  1 root  root      21508 Nov 20 18:45 dpkg.log.1
-rw-r--r--  1 root  root     32032 Nov 20 03:11 faillog
drwxr-sr-x+ 3 root  systemd-journal 4096 Apr 27 2020 journal
-rw-r----- 1 syslog adm          0 Nov 19 20:09 kern.log
-rw-r----- 1 syslog adm     63964 Nov 19 20:09 kern.log.1
-rw-rw-r--  1 root  utmp    292292 Nov 22 20:01 lastlog
-rw-r--r--  1 root  root          0 Nov  1 02:41 messages
drwx----- 2 root  root        4096 Apr 27 2020 private
-rw-r----- 1 syslog adm     12685 Nov 22 20:01 syslog
-rw-r----- 1 syslog adm     13447 Nov 22 00:00 syslog.1
-rw-r----- 1 syslog adm      2718 Nov 21 00:00 syslog.2.gz
-rw-r----- 1 syslog adm      1332 Nov 20 00:00 syslog.3.gz
-rw-r----- 1 syslog adm     19820 Nov 19 20:09 syslog.4.gz
-rw-----  1 root  root       3984 Nov 22 15:35 ubuntu-advantage.log
-rw-----  1 root  root        664 Nov 19 20:36 ubuntu-advantage.log.1
drwxr-x---  2 root  adm        4096 Oct 17 08:46 unattended-upgrades
-rw-rw-r--  1 root  utmp     17280 Nov 22 20:01 wtmp
den@vm355614:/var/log$
```

Команда cat

cat (сокращение от concatenate) — одна из наиболее часто используемых команд в Linux. Используется для вывода содержимого файла в командной строке (stdout). Чтобы запустить эту команду, введите cat, а затем имя файла и его расширение. Например: cat file.txt.

Вот другие варианты использования команды Linux cat:

```
den@vm355614:/var/log$ sudo cat syslog
Nov 22 00:00:23 vm355614 systemd[1]: logrotate.service: Succeeded.
Nov 22 00:00:23 vm355614 systemd[1]: Finished Rotate log files.
Nov 22 00:00:33 vm355614 systemd[1]: man-db.service: Succeeded.
den@vm355614:/var/log$
```

cat> filename создаёт новый файл cat filename1 filename2>filename3 объединяет два файла (1 и 2) и сохранит их содержимое в новом файле (3) Чтобы преобразовать файл в верхний или нижний регистр, cat filename | tr a-z A-Z >output.txt

Команда cp

Используйте команда cp для копирования файлов из текущего каталога в другой каталог.

```
den@vm355614:~$ ls
test_file.txt
```

```
den@vm355614:~$ cp test_file.txt test_file_1.txt
den@vm355614:~$ ls
test_file_1.txt  test_file.txt
den@vm355614:~$
```

Команда mv

Основное предназначение команды mv — перемещение файлов, хотя её также можно использовать для их переименования.

Аргументы в mv похожи на аргументы команды cp:

```
den@vm355614:~$ ls
test_file_1.txt  test_file.txt
den@vm355614:~$ mv test_file_1.txt new_test_file.txt
den@vm355614:~$ ls
new_test_file.txt  test_file.txt
den@vm355614:~$
```

Команда mkdir

Используется для создания каталогов:

```
den@vm355614:~$ mkdir test_dir
den@vm355614:~$ ls
new_test_file.txt  test_dir  test_file.txt
den@vm355614:~$
```

Команда rm

Команда rm используется для удаления файлов. Если вы хотите удалить каталог со всем его содержимым, в качестве альтернативы rmdir используйте rm с опцией -r.

Примечание: будьте очень осторожны с этой командой и всегда проверяйте, в каком каталоге вы находитесь. Она удаляет всё и её невозможно отменить.

```
den@vm355614:~$ touch test_dir/file1
den@vm355614:~$ touch test_dir/file2
den@vm355614:~$ tree
.
├── new_test_file.txt
├── test_dir
│   ├── file1
│   └── file2
└── test_file.txt

1 directory, 4 files
den@vm355614:~$
```

```
den@vm355614:~$ rm test_dir/file1
den@vm355614:~$ tree
.
├── new_test_file.txt
├── test_dir
│   └── file2
└── test_file.txt

1 directory, 3 files
```

```
den@vm355614:~$ rm -r test_dir/
den@vm355614:~$ tree
.
├── new_test_file.txt
```

```
└─ test_file.txt

0 directories, 2 files
```

Команда touch

Команда touch позволяет создать новый пустой файл через командную строку Linux:

```
den@vm355614:~$ touch file1 file2
den@vm355614:~$ ls
file1 file2 new_test_file.txt test_file.txt
den@vm355614:~$
```

Команда sudo

Сокращенно от «SuperUser Do», эта команда позволяет выполнять задачи, требующие прав администратора или root. Однако не рекомендуется использовать эту команду для повседневных задач, так как неправильное её использование может легко стать причиной появления ошибок.

```
den@vm355614:~$ apt-get install vim
E: Could not open lock file /var/lib/dpkg/lock-frontent - open (13: Permission
denied)
E: Unable to acquire the dpkg frontend lock (/var/lib/dpkg/lock-frontent), are you
root?
den@vm355614:~$
```

```
den@vm355614:~$ sudo apt-get install vim
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  alsa-topology-conf alsa-ucm-conf libasound2 libasound2-data libcanberra0 libgpm2
libogg0 libtdb1 libvorbis0a
libvorbisfile3 sound-theme-freedesktop vim-common vim-runtime vim-tiny
```

Команда df

Используйте команду df, чтобы получить отчёт об использовании дискового пространства в системе в процентах и килобайтах. Если вы хотите просмотреть отчёт в мегабайтах, введите df -m:

```
den@vm355614:~$ df
Filesystem      1K-blocks      Used Available Use% Mounted on
udev              730088         0       730088   0% /dev
tmpfs            151908        716       151192   1% /run
```

```
/dev/vda1      6126124 3015980  2789820  52% /
tmpfs          759520      0    759520   0% /dev/shm
tmpfs          5120      0     5120   0% /run/lock
tmpfs          759520      0    759520   0% /sys/fs/cgroup
tmpfs          151904      0    151904   0% /run/user/1000
den@vm355614:~$
```

Команда du

Если вы хотите проверить, сколько места занимает файл или каталог, воспользуйтесь командой du (Disk Usage). Однако вместо размера в обычном формате, в сводке вы увидите количество блоков диск. Если вы хотите посмотреть информацию в байтах, килобайтах и мегабайтах, добавьте аргумент `-h` в командную строку.

```
den@vm355614:/var/log$ du auth.log
128      auth.log
den@vm355614:/var/log$
```

Команда head

Команда head используется для просмотра первых строк любого текстового файла. По умолчанию она покажет первые десять строк:

```
root@vm355614:/var/log# head auth.log
Nov 21 00:17:01 vm355614 CRON[4139]: pam_unix(cron:session): session opened for
user root by (uid=0)
Nov 21 00:17:01 vm355614 CRON[4139]: pam_unix(cron:session): session closed for
user root
Nov 21 01:17:01 vm355614 CRON[4151]: pam_unix(cron:session): session opened for
user root by (uid=0)
Nov 21 01:17:01 vm355614 CRON[4151]: pam_unix(cron:session): session closed for
user root
Nov 21 01:20:03 vm355614 sshd[4154]: Connection closed by 111.111.111.111 port
34124 [preauth]
Nov 21 01:28:28 vm355614 sshd[4158]: Received disconnect from 111.111.111.111 port
50918:11: Bye Bye [preauth]
Nov 21 01:28:28 vm355614 sshd[4158]: Disconnected from 111.111.111.111 port 50918
[preauth]
Nov 21 02:17:01 vm355614 CRON[4167]: pam_unix(cron:session): session opened for
user root by (uid=0)
Nov 21 02:17:01 vm355614 CRON[4167]: pam_unix(cron:session): session closed for
user root
Nov 21 02:46:11 vm355614 sshd[4174]: Received disconnect from 111.111.111.111 port
56590:11: Bye Bye [preauth]
root@vm355614:/var/log#
```

Команда tail

Эта команда имеет функцию, аналогичную команде `head`, но вместо отображения первых строк `tail` выводит последние десять строк текстового файла:

```
root@vm355614:/var/log# tail auth.log
Nov 22 20:27:15 vm355614 sudo:      den : TTY=pts/2 ; PWD=/home/den ; USER=root ;
COMMAND=/usr/bin/apt-get install vim
Nov 22 20:27:15 vm355614 sudo: pam_unix(sudo:session): session opened for user
root by den(uid=0)
Nov 22 20:27:18 vm355614 sudo: pam_unix(sudo:session): session closed for user
root
Nov 22 20:27:33 vm355614 su: (to den) den on pts/2
Nov 22 20:27:33 vm355614 su: pam_unix(su:session): session opened for user den by
den(uid=1000)
Nov 22 20:28:14 vm355614 sudo:      den : TTY=pts/2 ; PWD=/home/den ; USER=root ;
COMMAND=/usr/bin/apt-get install vim
Nov 22 20:28:14 vm355614 sudo: pam_unix(sudo:session): session opened for user
root by den(uid=0)
Nov 22 20:28:41 vm355614 sudo: pam_unix(sudo:session): session closed for user
root
Nov 22 20:30:37 vm355614 su: (to root) den on pts/2
Nov 22 20:30:38 vm355614 su: pam_unix(su:session): session opened for user root by
den(uid=1000)
root@vm355614:/var/log#
```

Команда `chmod`

`chmod` — ещё одна команда Linux, используемая для изменения разрешений на чтение, запись и выполнение файлов и каталогов. Поскольку это довольно сложная команда, рекомендуем прочитать полное руководство (англ) по её применению.

Команда `chown`

В Linux все файлы принадлежат конкретному пользователю. Команда `chown` позволяет изменить или перенести владельца файла на указанное имя пользователя. Например, `chown linuxuser2 file.ext` сделает `linuxuser2` владельцем `file.ext`.

Команда `jobs`

Команда `jobs` отображает все текущие задачи вместе с их статусами. Задача — это процесс, запущенный в фоновом режиме.

```
den@vm355614:~$ nano test_file.txt

Use "fg" to return to nano.

[1]+  Stopped                  nano test_file.txt
den@vm355614:~$
```

```
den@vm355614:~$ jobs
[1]+  Stopped                  nano test_file.txt
den@vm355614:~$
```

Команда kill

Если у вас есть не отвечающая программа, вы можете завершить её вручную, используя команду `kill`. Команда отправит определённый сигнал неверно работающему приложению и даст ему команду прекратить работу.

```
den@vm355614:~$ ps aux
USER          PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root          6521   0.0   0.0      0      0 ?        I    20:25   0:00 [kworker/1:0-
events]
root          6522   0.0   0.0      0      0 ?        I    20:25   0:00 [kworker/1:3-
events]
root          6541   0.0   0.2  11392  4080 pts/2    S    20:27   0:00 su den
den           6542   0.0   0.2   9844  4012 pts/2    S    20:27   0:00 bash
root          6555   0.0   0.0      0      0 ?        I    20:28   0:00 [kworker/u4:2-
events_power_efficient]
root          6573   0.0   0.2  11360  4064 pts/2    S    20:30   0:00 su
root          6576   0.0   0.2   9844  4156 pts/2    S    20:30   0:00 bash
root          6581   0.0   0.2  11020  3840 pts/2    S    20:32   0:00 su den
den           6582   0.0   0.2   9844  4060 pts/2    S    20:32   0:00 bash
den           6584   0.0   0.2   9452  3908 pts/2    T    20:32   0:00 nano
test_file.txt
den           6585   0.0   0.2  11500  3412 pts/2    R+   20:33   0:00 ps aux
den@vm355614:~$
```

```
den@vm355614:~$ kill 6584
den@vm355614:~$
```

В общей сложности вы можете использовать шестьдесят четыре сигнала (англ), но люди обычно используют только два сигнала:

`SIGTERM` (15) — просит программу прекратить работу и даёт ей некоторое время, чтобы сохранить весь прогресс. Если вы не указали сигнал при вводе команды `kill`, этот сигнал будет использоваться по умолчанию. `SIGKILL` (9) — принудительно останавливает программы. Несохрани́нный прогресс будет потерян. Помимо знания сигналов, вам также необходимо знать числовой идентификатор процесса (PID) программы, которую вы хотите уничтожить. Если вы не знаете PID, просто запустите команду `ps` их.

Узнав, какой сигнал вы хотите использовать и PID программы, введите следующий синтаксис:

`kill [опция сигнала] PID.`

Команда ping

Используйте команду `ping` для проверки состояния подключения к серверу. Например, просто введя `ping google.com`, команда проверит, можете ли вы подключиться к Google, а также измерить время ответа.

```
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.  
64 bytes from 8.8.8.8: icmp_seq=1 ttl=112 time=15.3 ms  
64 bytes from 8.8.8.8: icmp_seq=2 ttl=112 time=15.4 ms  
64 bytes from 8.8.8.8: icmp_seq=3 ttl=112 time=15.3 ms  
64 bytes from 8.8.8.8: icmp_seq=4 ttl=112 time=15.2 ms  
  
--- 8.8.8.8 ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3005ms  
rtt min/avg/max/mdev = 15.178/15.292/15.371/0.075 ms  
den@vm355614:~$
```

Команда history

Регулярно пользуясь Linux, вы заметите, что запускаете сотни команд каждый день. Команда `history` позволяет просмотреть команды, которые вы вводили ранее.

```
den@vm355614:~$ history  
 1 cd  
 2 ls  
 3 nano test_file.txt  
 4 jobs  
 5 ps aux  
 6 kill 6584  
 7 jobs  
 8 ping 8.8.8.8  
 9 man ping  
10 man ping -c 4  
11 ping 8.8.8.8 -c 4  
12 history  
den@vm355614:~$
```

Команда man

Неуверены в функциях некоторых команд Linux? Не беспокойтесь, вы можете легко научиться использовать их прямо из оболочки Linux с помощью команды `man`.

```
den@vm355614:~$ man history  
HISTORY(3) Library Functions Manual  
HISTORY(3)  
  
NAME  
    history - GNU History Library
```

COPYRIGHT

The GNU History Library is Copyright (C) 1989-2017 by the Free Software Foundation, Inc.

DESCRIPTION

Many programs **read** input from the user a line at a time. The GNU History library is able to keep track of those lines, associate arbitrary data with each line, and utilize information from previous lines **in** composing new ones.

Команда hostname

Если вы хотите узнать имя вашего хоста/сети, просто введите hostname.

```
den@vm355614:~$ hostname
vm355614
den@vm355614:~$
```

Добавление -I в конце выведет IP-адрес вашей сети.

