

Bash: команды, фишки и примеры

Someanonimcoder

Содержание

Intro

В этом документе я хочу собрать основные команды bash и объяснить их для тех, кто работает на mac или linux

Любые дополнения, исправления и уточнения приветствуются.

Basic

Особое

Некоторые "магические" папки:

```
~      # home dir
..     # one dir up
-      # one dir back
.      # current dir
```

Перенаправление вывода команды:

```
# this is a comment. bash doesn't read it
# $ is a prompt for command input
# print output not in terminal but in file.txt
$ command > file.txt
# send output of command1 to command2 as (n+1)th argument
$ command1 | command2 arg1 arg2 ... argn
```

Читай мануалы, `$&#%*!`

```
# print help for command
$ man command
```

Как красиво попросить помощи у товарища? Читать километр вывода в vk или tg ну очень неудобно. Поэтому пожалуйста, используйте *pastein.it*. Это сайт, где ваш код или вывод терминала красиво оформится. Пример: *pastebin.com/hLw6DhQm*.

Еще есть команда *pastebinit*, она требует установки, но позволяет делать так:

```
$ command
<A wall of strange errors>
$ command | pastebinit
pastebin.com/AbCdEfGh
```

Потом можно скопировать ссылку и отправить мне, ИВС, ММ, кому угодно еще. Так мы быстрее прочтает Эту вашу стену сообщений об ошибках и поможем.

ОСТОРОЖНО!!! Все тексты публичны, так что пароли, ключи, персональные данные... ну вы поняли.

Навигация

Сюда я собрал команды для поиска и перемещения файлов, перехода по папкам, и всего такого

cd

Переход в папку выполняется командой

```
$ cd <name>
```

cp

Скопировать

```
# copy file to dir
$ cp file dir
#copy file to dir as file2
$ cp file dir/file2
```

mv

Переместить(move)

Идентично cp по логике работы

rm

Удаляет файл. ВНИМАНИЕ!!! Удаляет не в корзину, а по-взрослому. Без *-rf* не удаляет папки

```
#remove file
$ rm file
#remove directory
$ rm -rf dir
```

ls

Список файлов и папок можно просмотреть командой:

```
$ ls <name>
```

<name> здесь - имя той папки, список файлов в которой вы хотите увидеть

file

Эта команда выдает информацию о файле:

```
$ file MeCl.inp
MeCl.inp: ASCII text, with CRLF line terminators
$ file DZ1.zip
DZ1.zip: Zip archive data, at least v2.0 to extract
$ file doc.docx
doc.docx: Microsoft Word 2007+
```

find

Чтобы найти файл из терминала, используйте эту команду. Немного примеров:

```
$ find .
#show all files in current dir
$ find Docs -name "*.pdf"
#show files by name "*.pdf" in "Docs"
$ find Docs -name "*.pdf"
#same without register(file=fIle)
$find Docs -not -name "*.pdf"
#same, but NOT "*.pdf" files
```

Работа с текстом

Команды для эффективного просмотра, редактирования, анализа текста из командной строки

less

Показывает файл. Выход - *q*. Навигация - стрелочки и *pgup/pgdn*. Больше сказать и нечего.

more

Более примитивный вариант less, что иронично с учетом их названий

cat

Печатает файл в терминал. Пример, выводящий код программы лежащий в файле *rk.c*. Вывод сокращен

```
$ cat rk.c
#ifdef _OPENMP
#include <omp.h>
#endif

#include <iostream>

<..5000 lines..>

    std::cout << result << std::endl;
}
```

Не всегда удобно, особенно если файл на пару тысяч строк.

head

Печатает «голову»- 10 первых строк

```
$ head rk.c
#ifdef _OPENMP
#include <omp.h>
#endif

#include <iostream>

constexpr int size = 1000000;

inline double f(double x, double y) { return x * x; }
```

tail

Печатает «хвост» - 10 последних строк

```
$ tail rk.c
    for (int i = 0; i < threads; i++) {
        std::cout << "Thread_" << i << "_" << 0 + (double)size / (double)thr
            << "_" << (double)size / (double)threads * (i + 1) << "\n"
        result +=
            solveoderungekutta((double)size / (double)threads * i,
                               (double)size / (double)threads * (i + 1), 0,
        }

    std::cout << result << std::endl;
}
```

grep

Ищет заданный паттерн(первый аргумент) внутри второго аргумента. В примере ищет "energy" в выводе первой команды. Именно так grep чаще всего и применяют.

```
#second argument is output of previous command
$ cat NH3.out | grep "energy"
Change in XC energy                                ...      0.000026139
Gradient of the Kohn–Sham DFT energy:
<...>
Change in XC energy                                ...      0.000025435
```

sed

Мощный потоковый редактор файлов. Принимает два аргумента - строку-шаблон и то что хотим отредактировать

```
#s=substitute
$ echo "hello world" | sed "s/hello/goodbye/"
goodbye world
#only first!
$ echo "hello world" | sed "s/l/6/"
he6lo world
#g=global, all occurrences
$ echo "hello world" | sed "s/l/6/g"
he66o wor6d
```

nano

Няшный редактор в командной строке. Простой и интуитивно понятный.

```
$ nano
GNU nano 4.8 New Buffer
```

Text you are editing is here

At the end there is a manul **for** nano – nano shows it!

```
[ line 1/1 (100%), col 1/1 (100%), char 0/0 (0%) ]
^G Get Help      ^O Write Out    ^W Where Is     ^K Cut Text
^J Justify       ^C Cur Pos      M-U Undo
^X Exit          ^R Read File    ^\ Replace      ^U Paste Text
^T To Spell      ^_ Go To Line   M-E Redo
```

vi(vim)

Страшный редактор. Очень удобный отдельным гражданам, но сложный для начала работы. Если зашли случайно - <esc>, потом наберите : q! и забудьте как страшный сон.

Системное

date

Показывает дату в удобном виде. Может и установить ее, но это требует администраторских полномочий

```
$ date
Tue 24 MAR 2020 00:30:00
```

top

Показывает процессы и загрузку машины. Некрасивый но вроде стоит везде "из коробки"

```
Tasks: 263 total,   1 running, 262 sleeping,   0 stopped,
0 zombie
%Cpu(s):  1,4 us,   1,4 sy,   0,0 ni, 96,4 id,   0,0 wa,   0,0 hi,
0,7 si,   0
MiB Mem :   3821,4 total,    450,7 free,   1937,2 used,
1433,6 buff/cach
MiB Swap:   8192,0 total,   7674,7 free,    517,3 used.
1403,6 avail Mem
```

	PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU
%MEM		TIME+							
12,5	125991	someano+	20	0	9352	3972	3296	R	12,5
0,1		0:00.02							
	1	root	20	0	181408	5804	3976	S	0,0
0,1		0:05.15							
	2	root	20	0	0	0	0	S	0,0
0,0		0:00.11							
	3	root	0	-20	0	0	0	I	0,0
0,0		0:00.00							
	4	root	0	-20	0	0	0	I	0,0
0,0		0:00.00							
	8	root	0	-20	0	0	0	I	0,0
0,0		0:00.00							

htop

То же, но куда более красиво. Интерфейс простой и понятный, все удобно и круто, но требует установки

```

1  [ 0.0%] 2  [|||||||||||||||||100.0%]
Mem[|||||||||||||2.08G/3.73G] Tasks: 92, 263 thr; 1 running
Swp[|| 536M/8.00G] Load average: 0.55 0.61 0.54
Uptime: 1 day, 08:58:42

```

	PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	
Command												
	125857	somea	20	0	8332	4324	3412	R	0.7	0.1	0:00.09	hto..
	95156	somea	20	0	3273M	419M	41392	S	0.7	11.0	17:51.77	/us..
	1	root	20	0	177M	5804	3976	S	0.0	0.1	0:05.15	/sb..
	345	root	20	0	78068	104	0	S	0.0	0.0	0:00.00	/us..
	357	root	20	0	8120	1228	1136	S	0.0	0.0	1:14.08	/us..

ping

Проверка связи с хостом. Хост - адрес сайта или ip. Посылает тестовые пакеты пока не убьешь процесс. Ну или до количества переданного с ключом `-c`

```

$ ping -c 3 google.com
PING google.com (209.85.233.139) 56(84) bytes of data.
64 bytes from lr-in-f139.1e100.net (209.85.233.139): icmp_seq=1 ttl=46 t
64 bytes from lr-in-f139.1e100.net (209.85.233.139): icmp_seq=2 ttl=46 t
64 bytes from lr-in-f139.1e100.net (209.85.233.139): icmp_seq=3 ttl=46 t

```

— google.com ping statistics —

```

3 packets transmitted, 3 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 22.776/23.192/24.285/0.633 ms

```

```

$ ping 8.8.8.8

```

```

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=45 time=23.0 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=45 time=22.9 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=45 time=22.9 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=45 time=22.9 ms
^C

```

— 8.8.8.8 ping statistics —

```

4 packets transmitted, 4 received, 0% packet loss, time 3003ms
rtt min/avg/max/mdev = 22.857/22.917/22.971/0.041 ms

```

Здесь собраны команды для получения информации о системе и взаимодействия с ней.

reboot

Перезагрузка машины. Требуется быть администратором при наличии других пользователей залогиненных на эту машину. Не имеет аргументов

shutdown

Выключение. То же примечание что и для reboot. Принимает аргумент - время выключения.

```
$ shutdown now
```

Middle

Pro