



[Главная](#) >> [Команды](#) >> Команда ps в Linux

Команда ps в Linux

Опубликовано: 22 июня, 2020 от [admin](#), 7 комментариев, время чтения: 9 минут

Обнаружили ошибку в тексте? Сообщите мне об этом. Выделите текст с ошибкой и нажмите Ctrl+Enter.

Утилита ps одна из самых простых и в то же время часто используемых программ для просмотра списка процессов в Linux. Она не поддерживает интерактивный режим, зато имеет множество опций для настройки вывода тех или иных параметров процессов в Linux.

В этой статье мы рассмотрим как пользоваться ps, что с помощью неё можно посмотреть, а также какие основные опции можно использовать.

Содержание статьи

- [Что такое процесс?](#)
- [Команда ps в Linux](#)
- [Выводы](#)

Что такое процесс?

Чтобы понять что отображает команда ps сначала надо разобраться что такое процесс. Процесс Linux - это экземпляр программы, запущенный в памяти. Все процессы

[Конфиденциальность](#) - [Условия использования](#)

Privacy

разделить на обычные и фоновые. Более подробно об этом написано в статье [управление процессами Linux](#). Linux - это многопользовательская система, каждый пользователь может запускать одни и те же программы, и даже один пользователь может захотеть запустить несколько экземпляров одной программы, поэтому ядру нужно как-то идентифицировать такие однотипные процессы. Для этого каждому процессу присваивается PID (Process Identifier).

Каждый из процессов может находиться в одном из таких состояний:

- **Запуск** - процесс либо уже работает, либо готов к работе и ждет, когда ему будет дано процессорное время;
- **Ожидание** - процессы в этом состоянии ожидают какого-либо события или освобождения системного ресурса. Ядро делит такие процессы на два типа - те, которые ожидают освобождения аппаратных средств и приостановление с помощью сигнала;
- **Остановлено** - обычно, в этом состоянии находятся процессы, которые были остановлены с помощью сигнала;
- **Зомби** - это мертвые процессы, они были остановлены и больше не выполняются, но для них есть запись в таблице процессов, возможно, из-за того, что у процесса остались дочерние процессы.

А теперь давайте перейдем ближе к практике.

Команда ps в Linux

Сначала рассмотрим общий синтаксис команды, здесь все очень просто:

\$ ps опции

\$ ps опции | grep параметр

Во втором варианте мы используем утилиту grep для того, чтобы отобрать нужные нам процессы по определенному критерию. Теперь рассмотрим опции утилиты. Они делятся на два типа - те, которые идут с дефисом Unix и те, которые используются без дефиса - BSD. Лучше пользоваться только опциями Unix, но мы рассмотрим и одни и другие. Заметьте, что при использовании опций BSD, вывод утилиты будет организован в BSD стиле.

- **-A, -e, (a)** - выбрать все процессы;
- **-a** - выбрать все процессы, кроме фоновых;
- **-d, (g)** - выбрать все процессы, даже фоновые, кроме процессов сессий;
- **-N** - выбрать все процессы кроме указанных;
- **-C** - выбирать процессы по имени команды;

Privacy

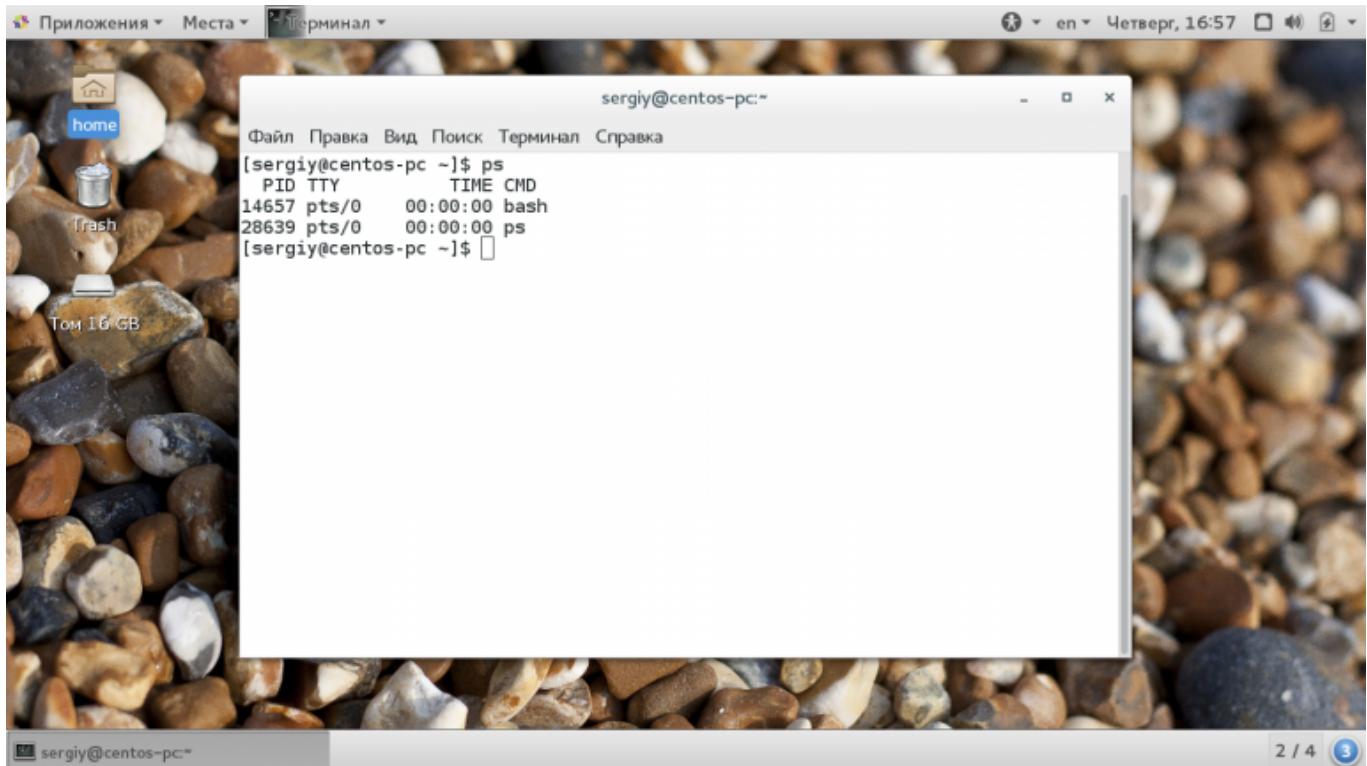
- **-G** - выбрать процессы по ID группы;
- **-p, (p)** - выбрать процессы PID;
- **--ppid** - выбрать процессы по PID родительского процесса;
- **-s** - выбрать процессы по ID сессии;
- **-t, (t)** - выбрать процессы по tty;
- **-u, (U)** - выбрать процессы пользователя.

Опции форматирования:

- **-c** - отображать информацию планировщика;
- **-f** - вывести максимум доступных данных, например, количество потоков;
- **-F** - аналогично -f, только выводит ещё больше данных;
- **-l** - длинный формат вывода;
- **-j, (j)** - вывести процессы в стиле Jobs, минимум информации;
- **-M, (Z)** - добавить информацию о безопасности;
- **-o, (o)** - позволяет определить свой формат вывода;
- **--sort, (k)** - выполнять сортировку по указанной колонке;
- **-L, (H)** - отображать потоки процессов в колонках LWP и NLWP;
- **-m, (m)** - вывести потоки после процесса;
- **-V, (V)** - вывести информацию о версии;
- **-H** - отображать дерево процессов;

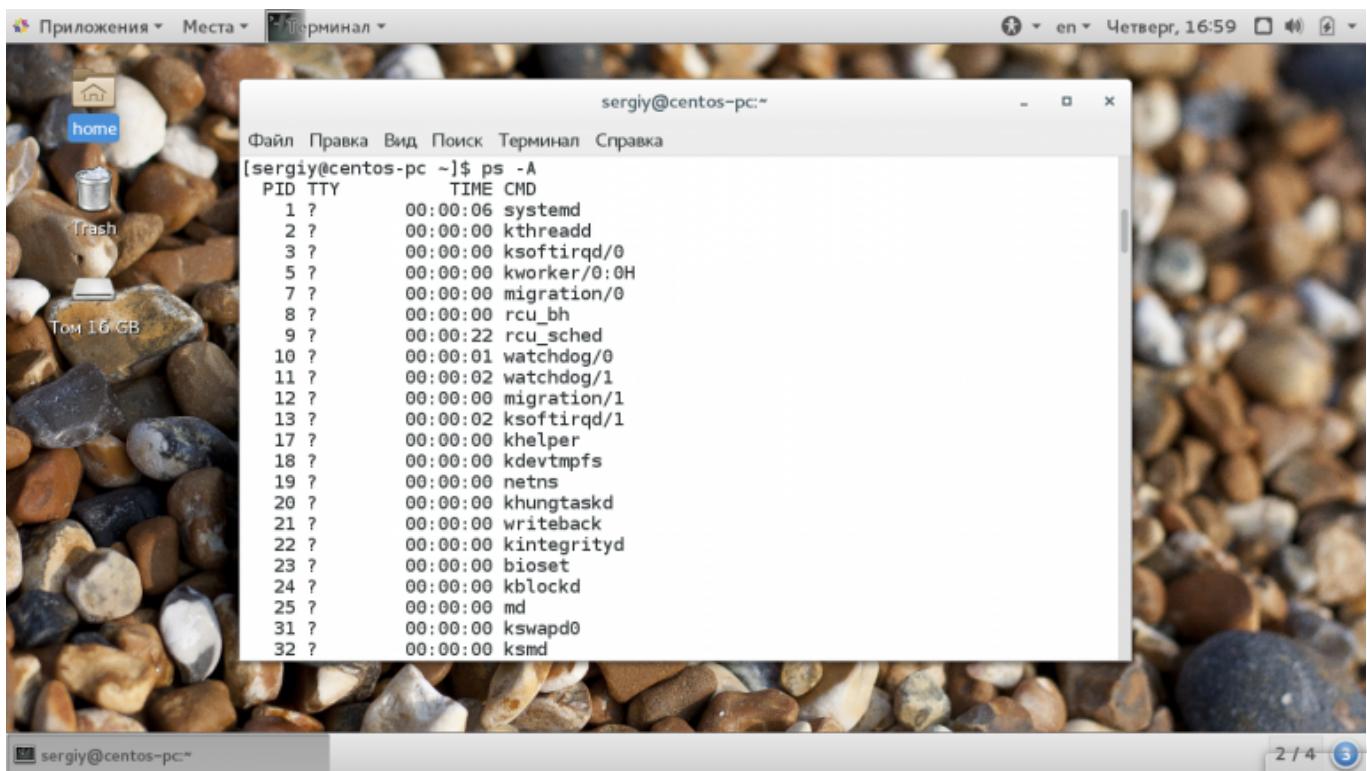
Теперь, когда вы знаете синтаксис и опции, можно перейти ближе к практике. Чтобы просто посмотреть процессы в текущей оболочке используется такая команда терминала ps:

```
$ ps
```



Все процессы, кроме лидеров групп, в том же режиме отображения:

\$ ps -A



Все процессы, включая фоновые и лидеры групп:

\$ ps -d

Privacy

```
sergiy@centos-pc:~$ ps -d
PID TTY      TIME CMD
 2 ?    00:00:00 kthreadd
 3 ?    00:00:00 ksoftirqd/0
 5 ?    00:00:00 kworker/0:0H
 7 ?    00:00:00 migration/0
 8 ?    00:00:00 rcu_bh
 9 ?    00:00:22 rcu_sched
10 ?    00:00:01 watchdog/0
11 ?    00:00:02 watchdog/1
12 ?    00:00:00 migration/1
13 ?    00:00:02 ksoftirqd/1
17 ?    00:00:00 khelper
18 ?    00:00:00 kdevtmpfs
19 ?    00:00:00 netns
20 ?    00:00:00 khungtaskd
21 ?    00:00:00 writeback
22 ?    00:00:00 kintegrityd
23 ?    00:00:00 bioset
24 ?    00:00:00 kblockd
25 ?    00:00:00 md
31 ?    00:00:00 kswapd0
32 ?    00:00:00 ksmd
33 ?    00:00:01 khugepaged
```

Чтобы вывести больше информации о процессах используйте опцию **-f**:

\$ ps -f

```
sergiy@centos-pc:~$ ps -f
UID      PID  PPID  C STIME TTY      TIME CMD
sergiy  32139 14652  0 17:54 pts/2    00:00:00 bash
sergiy  32303 32139  0 17:57 pts/2    00:00:00 ps -f
[sergiy@centos-pc ~]$
```

При использовании опции **-f** команда выдает такие колонки:

- **UID** - пользователь, от имени которого запущен процесс;
- **PID** - идентификатор процесса;

- **PPID** - идентификатор родительского процесса;
- **C** - процент времени CPU, используемого процессом;
- **STIME** - время запуска процесса;
- **TTY** - терминал, из которого запущен процесс;
- **TIME** - общее время процессора, затраченное на выполнение процессора;
- **CMD** - команда запуска процессора;
- **LWP** - показывает потоки процессора;
- **PRI** - приоритет процесса.

Например, также можно вывести подробную информацию обо всех процессах:

\$ ps -Af

```

Файл Правка Вид Поиск Терминал Справка
[sergiy@centos-pc ~]$ ps -Af
UID      PID  PPID  C STIME TTY          TIME CMD
root      1     0  0 07:24 ?        00:00:07 /usr/lib/systemd/systemd --switc
root      2     0  0 07:24 ?        00:00:00 [kthreadd]
root      3     2  0 07:24 ?        00:00:00 [ksoftirqd/0]
root      5     2  0 07:24 ?        00:00:00 [kworker/0:0H]
root      7     2  0 07:24 ?        00:00:00 [migration/0]
root      8     2  0 07:24 ?        00:00:00 [rcu_bh]
root      9     2  0 07:24 ?        00:00:26 [rcu_sched]
root     10    2  0 07:24 ?        00:00:02 [watchdog/0]
root     11    2  0 07:24 ?        00:00:02 [watchdog/1]
root     12    2  0 07:24 ?        00:00:00 [migration/1]
root     13    2  0 07:24 ?        00:00:03 [ksoftirqd/1]
root     17    2  0 07:24 ?        00:00:00 [khelper]
root     18    2  0 07:24 ?        00:00:00 [kdevtmpfs]
root     19    2  0 07:24 ?        00:00:00 [netns]
root     20    2  0 07:24 ?        00:00:00 [khungtaskd]
root     21    2  0 07:24 ?        00:00:00 [writeback]
root     22    2  0 07:24 ?        00:00:00 [kintegrityd]
root     23    2  0 07:24 ?        00:00:00 [bioset]
root     24    2  0 07:24 ?        00:00:00 [kblockd]
root     25    2  0 07:24 ?        00:00:00 [md]
root     31    2  0 07:24 ?        00:00:00 [kswapd0]
root     32    2  0 07:24 ?        00:00:00 [ksmd]

```

Больше информации можно получить, использовав опцию **-F**:



\$ ps -Fe

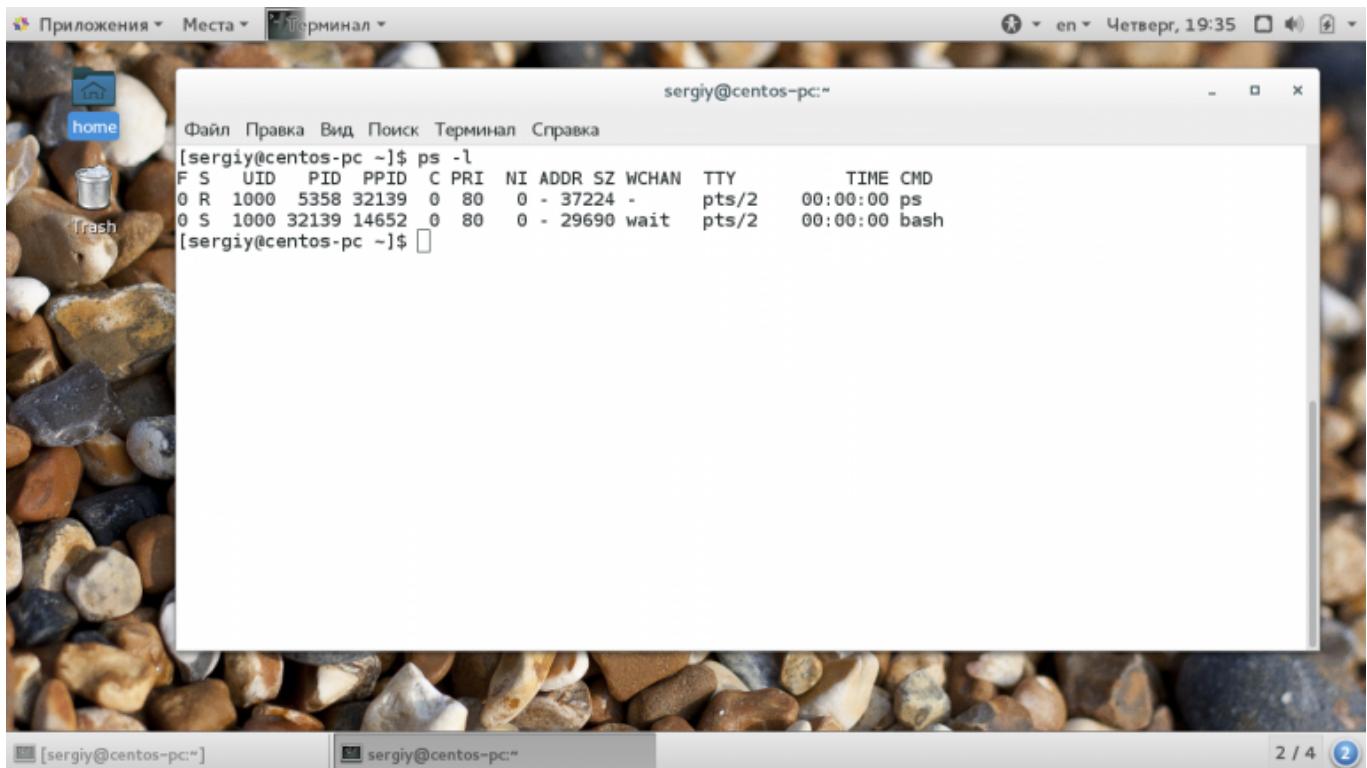
```
sergiy@centos-pc:~$ ps -Fe
UID      PID  PPID  C   SZ   RSS PSR STIME TTY          TIME CMD
root      1      0    0 32024  6732  1 07:24 ?          00:00:07 /usr/lib/systemd/systemd --switched-roo
root      2      0    0      0      0  1 07:24 ?          00:00:00 [kthreadd]
root      3      2    0      0      0  0 07:24 ?          00:00:00 [ksoftirqd/0]
root      5      2    0      0      0  0 07:24 ?          00:00:00 [kworker/0:0H]
root      7      2    0      0      0  0 07:24 ?          00:00:00 [migration/0]
root      8      2    0      0      0  0 07:24 ?          00:00:00 [rcu_bh]
root      9      2    0      0      0  0 07:24 ?          00:00:33 [rcu_sched]
root     10      2    0      0      0  0 07:24 ?          00:00:02 [watchdog/0]
root     11      2    0      0      0  1 07:24 ?          00:00:02 [watchdog/1]
root     12      2    0      0      0  1 07:24 ?          00:00:00 [migration/1]
root     13      2    0      0      0  1 07:24 ?          00:00:05 [ksoftirqd/1]
root     17      2    0      0      0  0 07:24 ?          00:00:00 [khelper]
root     18      2    0      0      0  1 07:24 ?          00:00:00 [kdevtmpfs]
root     19      2    0      0      0  1 07:24 ?          00:00:00 [netns]
root     20      2    0      0      0  1 07:24 ?          00:00:00 [khungtaskd]
root     21      2    0      0      0  0 07:24 ?          00:00:00 [writeback]
root     22      2    0      0      0  1 07:24 ?          00:00:00 [kintegrityd]
root     23      2    0      0      0  1 07:24 ?          00:00:00 [bioset]
root     24      2    0      0      0  1 07:24 ?          00:00:00 [kblockd]
root     25      2    0      0      0  0 07:24 ?          00:00:00 [md]
root     31      2    0      0      0  1 07:24 ?          00:00:00 [kswapd0]
root     32      2    0      0      0  0 07:24 ?          00:00:00 [ksmd]
```

Эта опция добавляет такие колонки:

- **SZ** - это размер процесса в памяти;
- **RSS** - реальный размер процесса в памяти;
- **PSR** - ядро процессора, на котором выполняется процесс.

Если вы хотите получить еще больше информации, используйте вместо **-f** опцию **-l**:

\$ ps -l



Эта опция добавляет отображение таких колонок:

- **F** - флаги, ассоциированные с этим процессом;
- **S** - состояние процесса;
- **PRI** - приоритет процесса в планировщике ядра Linux;
- **NI** - рекомендованный приоритет процесса, можно менять;
- **ADDR** - адрес процесса в памяти;
- **WCHAN** - название функции ядра, из-за которой процесс находится в режиме ожидания.

Дальше мы можем отобрать все процессы, запущенные от имени определенного пользователя:

```
$ ps -fu root
```

```
sergiy@centos-pc:~$ ps -fu root
UID      PID  PPID  C STIME TTY          TIME CMD
root      1      0  0 07:24 ?
root      2      0  0 07:24 ?
root      3      2  0 07:24 ?
root      5      2  0 07:24 ?
root      7      2  0 07:24 ?
root      8      2  0 07:24 ?
root      9      2  0 07:24 ?
root     10      2  0 07:24 ?
root     11      2  0 07:24 ?
root     12      2  0 07:24 ?
root     13      2  0 07:24 ?
root     17      2  0 07:24 ?
root     18      2  0 07:24 ?
root     19      2  0 07:24 ?
root     20      2  0 07:24 ?
root     21      2  0 07:24 ?
root     22      2  0 07:24 ?
root     23      2  0 07:24 ?
root     24      2  0 07:24 ?
root     25      2  0 07:24 ?
root     31      2  0 07:24 ?
root     32      2  0 07:24 ?

TIME CMD
00:00:07 /usr/lib/systemd/systemd --switched-root --system --des
00:00:00 [kthreadd]
00:00:00 [ksoftirqd/0]
00:00:00 [kworker/0:0H]
00:00:00 [migration/0]
00:00:00 [rcu_bh]
00:00:33 [rcu_sched]
00:00:02 [watchdog/0]
00:00:02 [watchdog/1]
00:00:00 [migration/1]
00:00:05 [ksoftirqd/1]
00:00:00 [khelper]
00:00:00 [kdevtmpfs]
00:00:00 [netns]
00:00:00 [khungtaskd]
00:00:00 [writelock]
00:00:00 [kintegrityd]
00:00:00 [bioset]
00:00:00 [kblockd]
00:00:00 [md]
00:00:00 [kswapd0]
00:00:00 [ksmd]
```

С помощью опции **-H** можно отобразить дерево процессов:

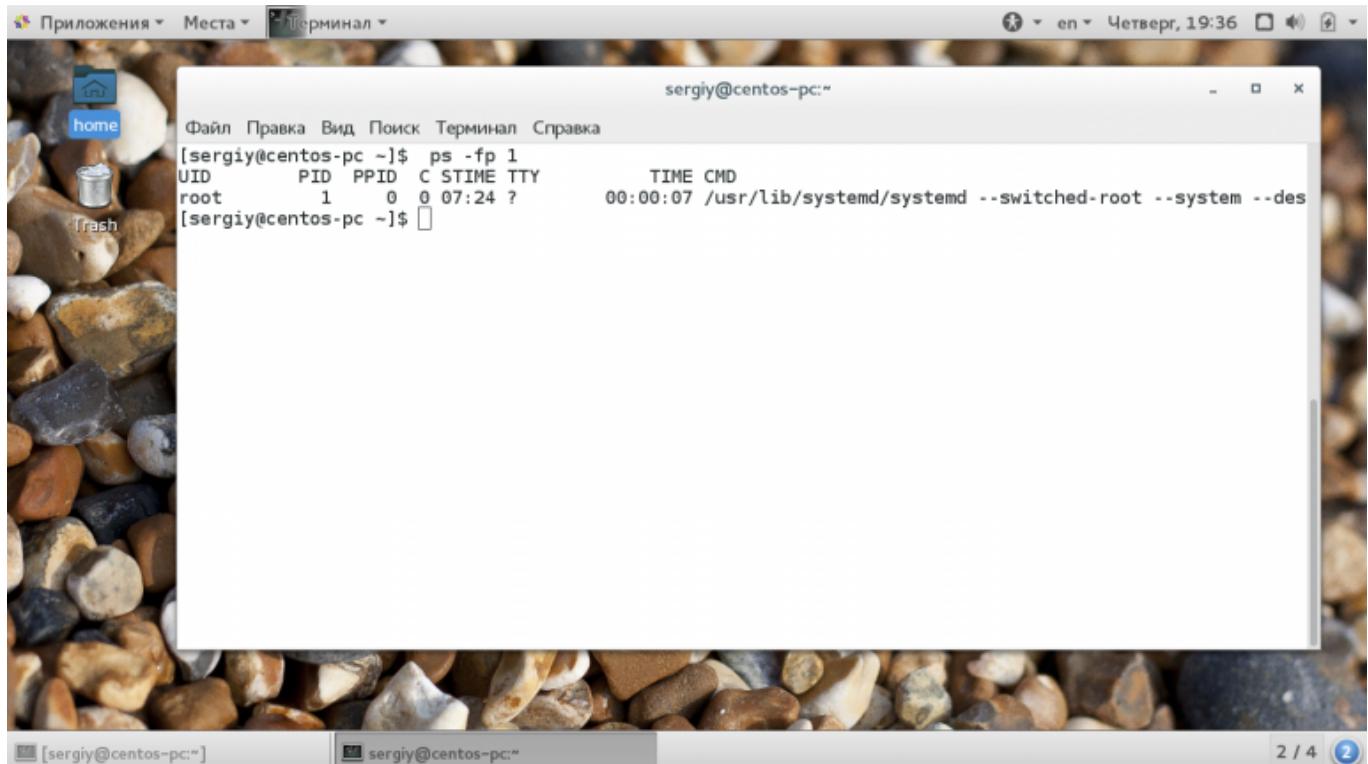
\$ ps -fHu root

```
sergiy@centos-pc:~$ ps -fHu root
UID      PID  PPID  C STIME TTY          TIME CMD
root    2306  2305  0 07:25 ?
root      2      0  0 07:24 ?
root      3      2  0 07:24 ?
root      5      2  0 07:24 ?
root      7      2  0 07:24 ?
root      8      2  0 07:24 ?
root      9      2  0 07:24 ?
root     10      2  0 07:24 ?
root     11      2  0 07:24 ?
root     12      2  0 07:24 ?
root     13      2  0 07:24 ?
root     17      2  0 07:24 ?
root     18      2  0 07:24 ?
root     19      2  0 07:24 ?
root     20      2  0 07:24 ?
root     21      2  0 07:24 ?
root     22      2  0 07:24 ?
root     23      2  0 07:24 ?
root     24      2  0 07:24 ?
root     25      2  0 07:24 ?
root     31      2  0 07:24 ?
root     32      2  0 07:24 ?

TIME CMD
00:00:00 /sbin/dnsmasq --conf-file=/var/lib/libvirt/dnsmasq/defa
00:00:00 [kthreadd]
00:00:00 [ksoftirqd/0]
00:00:00 [kworker/0:0H]
00:00:00 [migration/0]
00:00:00 [rcu_bh]
00:00:33 [rcu_sched]
00:00:02 [watchdog/0]
00:00:02 [watchdog/1]
00:00:00 [migration/1]
00:00:05 [ksoftirqd/1]
00:00:00 [khelper]
00:00:00 [kdevtmpfs]
00:00:00 [netns]
00:00:00 [khungtaskd]
00:00:00 [writelock]
00:00:00 [kintegrityd]
00:00:00 [bioset]
00:00:00 [kblockd]
00:00:00 [md]
00:00:00 [kswapd0]
00:00:00 [ksmd]
```

Если вас интересует информация только об определенном процессе, то вы можете использовать опцию **-p** и указать PID процесса:

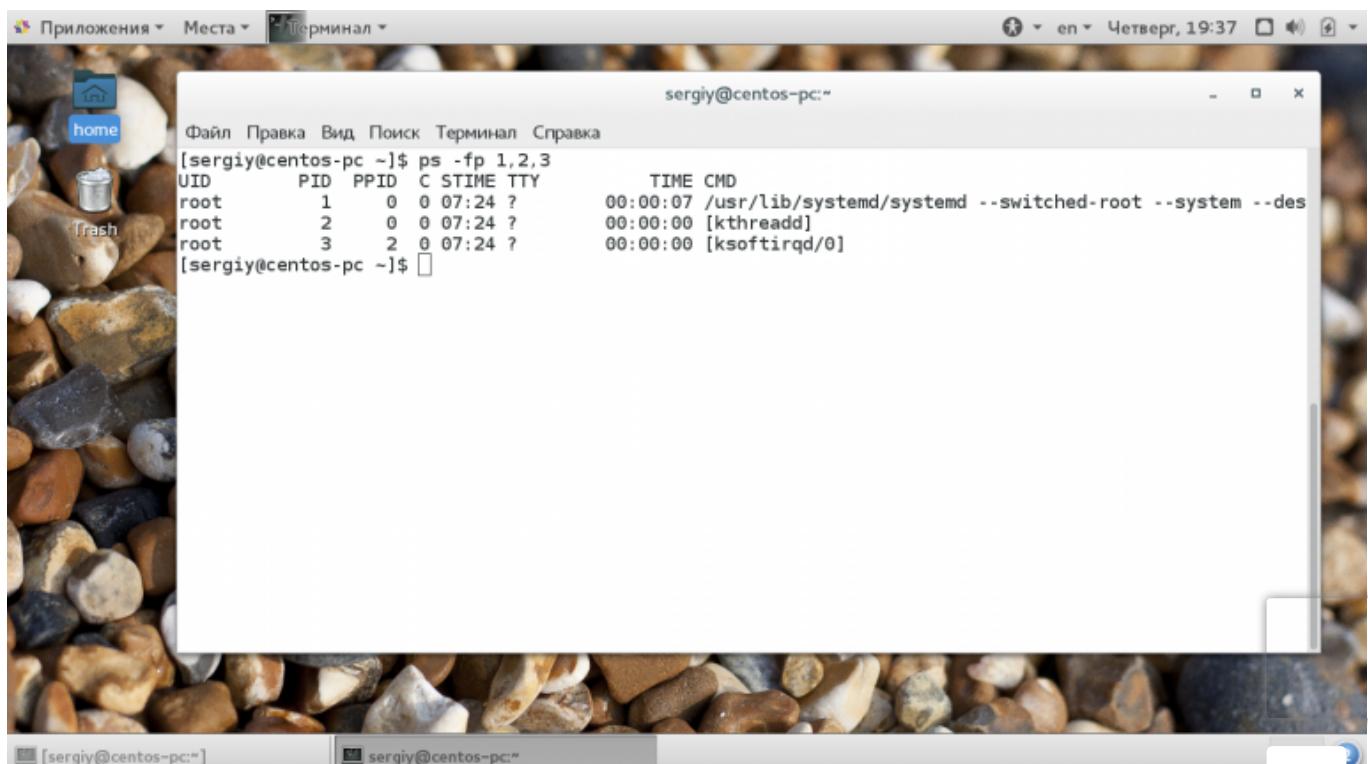
```
$ ps -fp 1
```



```
sergiy@centos-pc:~$ ps -fp 1
UID      PID  PPID  C STIME TTY          TIME CMD
root      1      0  0 07:24 ?        00:00:07 /usr/lib/systemd/systemd --switched-root --system --des
[sergiy@centos-pc ~]$
```

Через запятую можно указать несколько PID:

```
$ ps -fp 1,2,3
```



```
sergiy@centos-pc:~$ ps -fp 1,2,3
UID      PID  PPID  C STIME TTY          TIME CMD
root      1      0  0 07:24 ?
root      2      0  0 07:24 ?
root      3      2  0 07:24 ?
00:00:07 /usr/lib/systemd/systemd --switched-root --system --des
00:00:00 [kthreadd]
00:00:00 [ksoftirqd/0]
[sergiy@centos-pc ~]$
```

Опция **-C** позволяет фильтровать процессы по имени, например, выберем только процессы chrome:

```
$ ps -fC chrome
```

```
[sergiy@centos-pc ~]$ ps -fC chrome
UID      PID  PPID  C STIME TTY          TIME CMD
sergiy    1498  4744  0 18:19 ?        00:00:23 /opt/google/chrome/chrome --type=renderer --field-trial
sergiy    4725  4084  4 07:32 ?        00:35:01 /opt/google/chrome/chrome
sergiy    4735  4734  0 07:32 ?        00:00:00 /opt/google/chrome/chrome --type=zygote
sergiy    4744  4735  0 07:32 ?        00:00:01 /opt/google/chrome/chrome --type=zygote
sergiy    4826  4725  2 07:32 ?        00:15:28 /opt/google/chrome/chrome --type=gpu-process --field-trial
sergiy    4828  4826  0 07:32 ?        00:00:00 /opt/google/chrome/chrome --type=gpu-broker
sergiy    4885  4744  0 07:32 ?        00:00:00 /opt/google/chrome/chrome --type=renderer --field-trial
sergiy    4891  4744  0 07:32 ?        00:00:04 /opt/google/chrome/chrome --type=renderer --field-trial
sergiy    4904  4744  0 07:33 ?        00:02:50 /opt/google/chrome/chrome --type=renderer --field-trial
sergiy    4910  4744  0 07:33 ?        00:00:48 /opt/google/chrome/chrome --type=renderer --field-trial
sergiy    4914  4744  0 07:33 ?        00:00:00 /opt/google/chrome/chrome --type=renderer --field-trial
sergiy    4920  4744  0 07:33 ?        00:00:05 /opt/google/chrome/chrome --type=renderer --field-trial
sergiy    4929  4744  0 07:33 ?        00:00:11 /opt/google/chrome/chrome --type=renderer --field-trial
sergiy    4935  4744  0 07:33 ?        00:00:29 /opt/google/chrome/chrome --type=renderer --field-trial
sergiy    4938  4744  0 07:33 ?        00:00:00 /opt/google/chrome/chrome --type=renderer --field-trial
sergiy    4946  4744  0 07:33 ?        00:00:04 /opt/google/chrome/chrome --type=renderer --field-trial
sergiy    5102  4744  0 07:33 ?        00:00:00 /opt/google/chrome/chrome --type=renderer --field-trial
sergiy    5129  4744  2 07:33 ?        00:18:34 /opt/google/chrome/chrome --type=renderer --field-trial
sergiy    5207  4744  0 07:33 ?        00:00:15 /opt/google/chrome/chrome --type=renderer --field-trial
sergiy    5234  4744  0 07:33 ?        00:01:03 /opt/google/chrome/chrome --type=renderer --field-trial
sergiy    5261  4744  0 07:33 ?        00:00:26 /opt/google/chrome/chrome --type=renderer --field-trial
sergiy   10095  4744  0 09:43 ?        00:00:12 /opt/google/chrome/chrome --type=renderer --field-trial
```

Дальше можно использовать опцию **-L** чтобы отобразить информацию о процессах:

```
$ ps -fL
```

```
sergiy@centos-pc:~$ ps -fL
UID      PID  PPID  LWP   C NLWP STIME TTY          TIME CMD
sergiy    5489  32139  5489  0    1 19:37 pts/2    00:00:00 ps -fL
sergiy    32139 14652  32139  0    1 17:54 pts/2    00:00:00 bash
[sergiy@centos-pc ~]$
```

Очень интересно то, с помощью опции **-o** можно настроить форматирование вывода, например, вы можете вывести только pid процесса и команду:

```
$ ps -o pid,comm
```

```
sergiy@centos-pc:~$ ps -o pid,comm
PID COMMAND
5541 ps
32139 bash
[sergiy@centos-pc ~]$
```

Вы можете выбрать такие колонки для отображения: **pcpu**, **pmem**, **args**, **comm**, **cputime**, **pid**, **gid**, **lwp**, **rss**, **start**, **user**, **vsize**, **priority**. Для удобства просмотра можно

отсортировать вывод программы по нужной колонке, например, просмотр процессов, которые используют больше всего памяти:

```
$ ps -Fe --sort rss
```

```

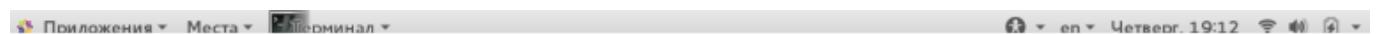
Файл Правка Вид Поиск Терминал Справка
sergiy 4929 4744 0 227954 60256 0 07:33 ?
sergiy 4946 4744 0 231947 68744 0 07:33 ?
sergiy 4634 1 0 473292 73500 0 07:32 ?
sergiy 11575 4744 0 242833 95592 1 10:01 ?
sergiy 1498 4744 0 240370 98788 1 18:19 ?
sergiy 11607 4744 0 242321 101352 1 10:01 ?
sergiy 5207 4744 0 239570 102688 1 07:33 ?
sergiy 5261 4744 0 244015 106552 1 07:33 ?
sergiy 4910 4744 0 245858 114216 1 07:33 ?
sergiy 10095 4744 0 253622 134068 1 09:43 ?
sergiy 11927 4744 1 257347 137756 0 10:04 ?
sergiy 32247 4744 0 262647 145224 1 17:56 ?
sergiy 4904 4744 0 268659 157032 0 07:33 ?
sergiy 4084 3793 2 477303 158092 0 07:32 ?
sergiy 14149 4744 0 277105 168604 1 10:45 ?
sergiy 21570 4744 1 344172 206556 1 14:43 ?
sergiy 4826 4725 2 203943 249848 1 07:32 ?
sergiy 4725 4084 4 466068 286388 1 07:32 ?
sergiy 5234 4744 0 382421 304408 0 07:33 ?
sergiy 12466 4744 0 382427 316752 1 10:13 ?
sergiy 12004 4744 7 312898 329020 1 10:05 ?
sergiy 30457 4744 5 394044 401840 0 17:31 ?
sergiy 5129 4744 2 449883 511272 1 07:33 ?
[sergiy@centos-pc ~]$
```

Или по проценту загрузки CPU:

```
$ ps -FA --sort pcru
```

Ещё одна опция - **-M**, которая позволяет вывести информацию про права безопасности и флаги SELinux для процессов:

```
$ ps -eM
```



Общее количество запущенных процессов Linux можно узнать командой:

```
$ ps -e | wc
```



Мы рассмотрели все основные возможности утилиты ps. Дальше вы можете поэкспериментировать с её параметрами и опциями чтобы найти нужные комбинации, также можно попытаться применить опции BSD.

Выводы

Команда ps Linux может быть очень полезной если система перегружена и вам необходимо срочно узнать запущенные процессы Linux чтобы освободить память или ресурсы процессора. Интерактивные средства не всегда могут помочь, потому что они потребляют слишком много ресурсов. С другой стороны ps дает большую гибкость поскольку утилита имеет множество опций и параметров. Во всяком случае теперь вы знаете как вывести список процессов. Если остались вопросы, спрашивайте в комментариях!

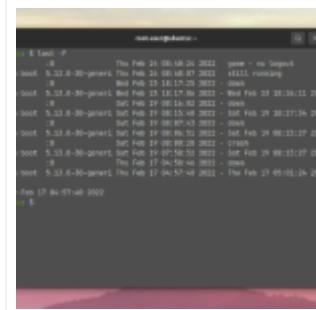
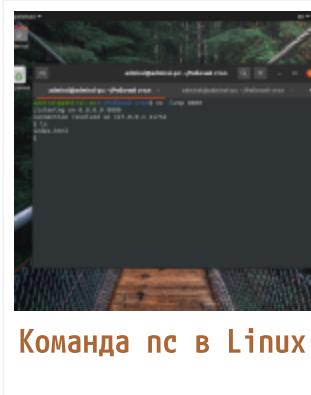
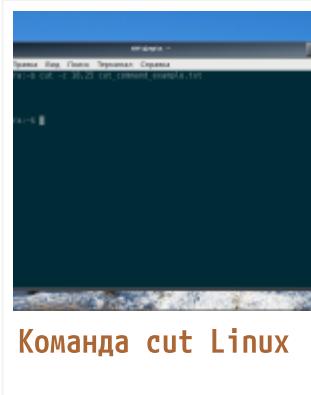
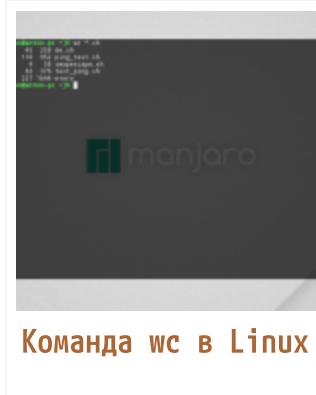
<https://www.youtube.com/watch?v=ftn71nXz5hc>

Была ли эта информация полезной для вас? Да Нет

X

Похожие записи

[Privacy](#)



Оцените статью

(10 оценок, среднее: 4,40 из 5)



Статья распространяется под лицензией Creative Commons ShareAlike 4.0 при копировании материала ссылка на источник обязательна .

[Команды](#)

Об авторе



ADMIN

Основатель и администратор сайта losst.ru, увлекаюсь открытым программным обеспечением и операционной системой Linux. В качестве основной ОС сейчас использую Ubuntu. Кроме Linux, интересуюсь всем, что связано с информационными технологиями и современной наукой.

7 комментариев к “Команда ps в Linux”



Сардор

16 июля, 2020 в 6:25 пп

круто

[Ответить](#)



Прохожий

3 августа, 2020 в 1:45 дп

Спасибо тебе за работу, добрый человек, admin.

Приятно читать.

[Ответить](#)



юрий

3 ноября, 2021 в 1:37 пп

Спасибо, огромное

[Ответить](#)



FiveFiveFive

2 февраля, 2022 в 10:00 пп

[Privacy](#)

За статью конечно спасибо! но ранее в своих статьях вы обычно давали краткую инструкцию как эту полезную команду можно установить. так вот, как её поставить?

[Ответить](#)



admin

2 февраля, 2022 в 10:25пп

Программа находится в пакете pprocps и в большинстве дистрибутивов поставляется из коробки.

[Ответить](#)



Den

5 февраля, 2022 в 9:12пп

stranno 4to net info po "ps -aux"

[Ответить](#)



Артём

26 сентября, 2022 в 12:15пп

Добрый день!

Можно пояснить один момент: в статье говорится, что зомби-процессы появляются из-за того, что у этого процесса есть дочерние процессы. Но, насколько я знаю, жизненный цикл процесса построен так, что родительский процесс всегда хоронит дочерний после того, как получит от него системный вызов exit().
Также читал, что зомби-процессы могут появляться в виду причинно-следственных связей или проблем с железом.

Прошу Вас, если можно, пояснить этот момент. Большое Вам спасибо!

Большое спасибо за сайт и информацию, представленную на нём!

[Privacy](#)

[Ответить](#)

Оставьте комментарий

 Имя * Email

Я прочитал и принимаю политику конфиденциальности. Подробнее [Политика конфиденциальности](#) *

[Комментировать](#)

[Русский](#)

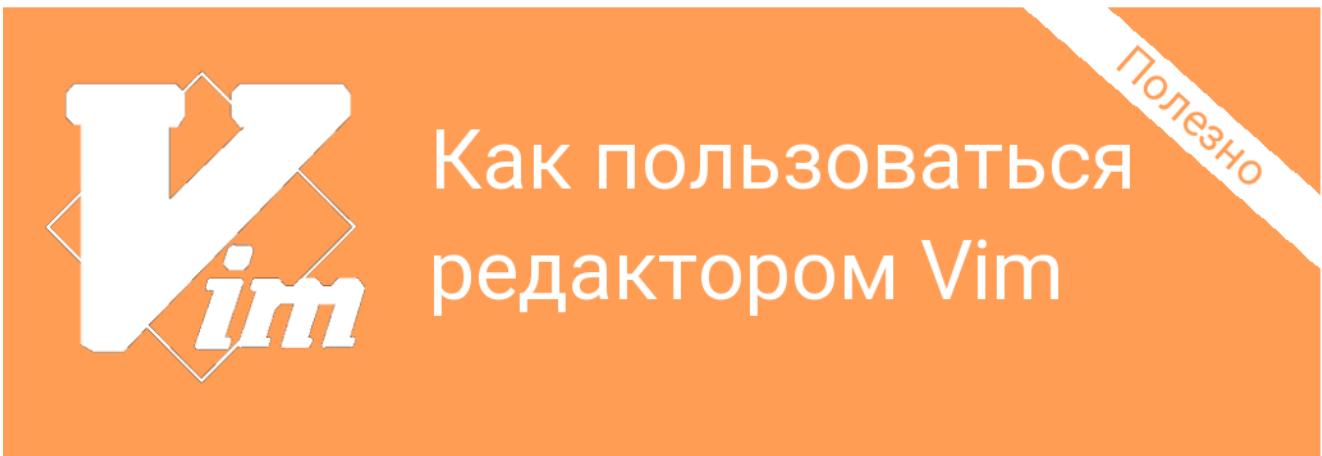
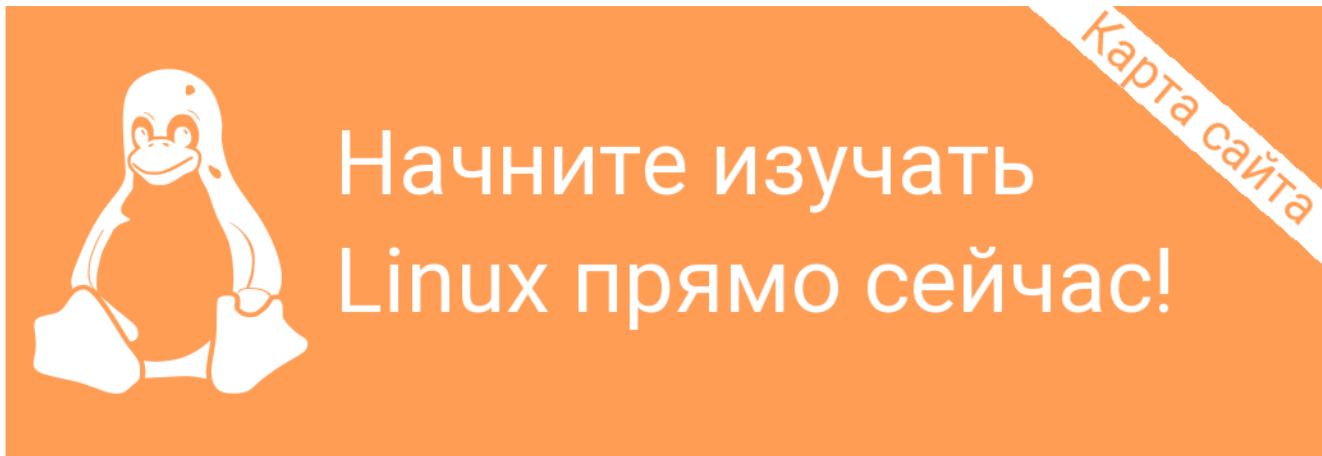
Поиск

[Privacy](#)

ПОИСК ПО КОМАНДАМ

Начните вводить команду

Поиск



Лучшие

Свежие

Теги

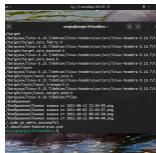


Команда chmod Linux

2020-04-13

Команда find в Linux

Privacy



2021-10-17

Как узнать IP-адрес Linux

2023-04-14



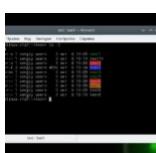
Настройка Старт

2021-10-01



Права доступа к файлам в Linux

2020-10-09



РАССЫЛКА

Ваш E-Mail адрес

Я прочитал(а) и принимаю политику конфиденциальности

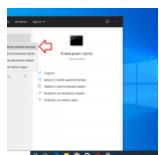
Sign up

Privacy



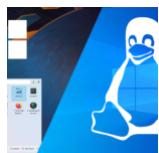
Windows

Списки



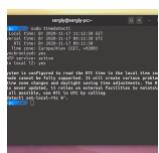
Восстановление Grub после установки Windows 10

2020-08-15



Установка Linux рядом с Windows 10 или 11

2023-02-08



Сбивается время в Ubuntu и Windows

2023-02-18



Ошибка Ubuntu не видит сеть Windows

2023-02-18

Смотреть ещё

META

Privacy

[Регистрация](#)[Войти](#)[Лента записей](#)[Лента комментариев](#)

СЛЕДИТЕ ЗА НАМИ В СОЦИАЛЬНЫХ СЕТЯХ

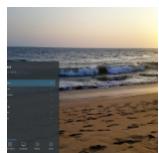


Интересное



Линус Торвальдс – человек, создавший Linux

2021-01-28



Самые маленькие дистрибутивы Linux

2020-12-15



Лучшие VPS сервера для VPN

2022-09-04



Лучшие панели управления для Linux

2020-12-08

[Privacy](#)

©Losst 2024 CC-BY-SA [Политика конфиденциальности](#)