

Sysadminium

База знаний системного администратора

Утилита top – наблюдаем за процессами

В этой статье будет показана **утилита top**, с помощью которой можно наблюдать за активностью процессов в реальном времени.



Оглавление [скрыть]

Утилита top

Общая информация о системе

Информация по каждому процессу

Опции команды top

Количество итераций

Процессы определённого пользователя

Горячие клавиши top

Режим цветности

Больше информации о команде (поле COMMAND)

Сортировка процессов

Завершение процессов

Добавление полей

Итог

Утилита top

Утилита top — это консольный диспетчер задач. Он показывает общую информацию о системе и информацию о каждом процессе. Работа этой утилиты выглядит следующим образом:

```
top - 12:36:48 up 7 min, 1 user, load average: 0,00, 0,03, 0,00
Tasks: 101 total, 1 running, 100 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0,0 us, 0,2 sy, 0,0 ni, 99,8 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st
MiB Mem : 969,2 total, 424,2 free, 186,8 used, 358,3 buff/cache
MiB Swap: 2048,0 total, 2048,0 free, 0,0 used. 647,3 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1	root	20	0	166240	11404	8204	S	0,0	1,1	0:01.16	systemd
2	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kthreadd
3	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	rcu_par_gp
5	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	netns
6	root	20	0	0	0	0	I	0,0	0,0	0:00.02	kworker/0:0-events
7	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker/0:0H-events_highpri
8	root	20	0	0	0	0	I	0,0	0,0	0:00.04	kworker/u4:0-events_unbound
9	root	0	-20	0	0	0	I	0,0	0,0	0:00.01	kworker/0:1H-events_highpri
10	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	mm_percpu_wq
11	root	20	0	0	0	0	S	0,0	0,0	0:00.00	rcu_tasks_rude_
12	root	20	0	0	0	0	S	0,0	0,0	0:00.00	rcu_tasks_trace
13	root	20	0	0	0	0	S	0,0	0,0	0:00.03	ksoftirqd/0
14	root	20	0	0	0	0	I	0,0	0,0	0:00.10	rcu_sched
15	root	rt	0	0	0	0	S	0,0	0,0	0:00.00	migration/0
16	root	-51	0	0	0	0	S	0,0	0,0	0:00.00	idle_inject/0
18	root	20	0	0	0	0	S	0,0	0,0	0:00.00	cpuhp/0
19	root	20	0	0	0	0	S	0,0	0,0	0:00.00	cpuhp/1
20	root	-51	0	0	0	0	S	0,0	0,0	0:00.00	idle_inject/1
21	root	rt	0	0	0	0	S	0,0	0,0	0:00.18	migration/1
22	root	20	0	0	0	0	S	0,0	0,0	0:00.02	ksoftirqd/1
24	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker/1:0H-events_highpri
25	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kdevtmpfs

Работа утилиты top



Вы можете почитать официальный мануал по использованию **top** [здесь](#), или выполните команду `man top`.

Это не привычная нам утилита командной строки. Это – интерактивная утилита, то есть она постоянно обновляет данные и с ней можно взаимодействовать с помощью горячих клавиш.

Вывод данной утилиты можно разбить на две части:

- сверху – шапка. Здесь вы можете посмотреть на общую информацию о системе;
- ниже – информация по каждому процессу.

Общая информация о системе

Как видно, шапка состоит из пяти строк.

На первой и второй строке показана общая информация о системе:

```
top - 14:02:55 up 1:33, 1 user, load average: 0,00, 0,00, 0,00
Tasks: 100 total, 1 running, 99 sleeping, 0 stopped, 0 zombie
```

Утилита top – общая информация о системе

На первой строке показаны следующие поля:

- текущее время;
- как долго работает система после последней загрузки;
- сколько в системе активных пользователей;
- средняя загрузка системы (**load average**) — три величины, усредненные за последние 1 и 5 и 15 минут. Чем ниже, тем лучше. Если превышает количество ядер на сервере, то значит сервер перегружен.

Ниже:

- количество процессов;
- количество работающих в данный момент процессов;
- количество спящих процессов;
- количество остановленных процессов;
- количество зомби процессов.

На третьей строке показана информация о потреблении ЦПУ:



```
%Cpu(s):  0,0 us,  0,2 sy,  0,0 ni, 99,8 id,  0,0 wa,  0,0 hi,  0,0 si,  0,0 st
```

Утилита top – информация о потреблении ЦПУ

- **us** — потребление сри пользовательскими процессами (top и apache2);
- **sy** — потребление сри системными процессами;
- **ni** — потребление сри пользовательскими процессами, с измененным приоритетом (командой **nice** или **renice**);
- **id** — простаивание сри;
- **wa** — процент времени, в течение которого процессор простаивал, ожидая завершения операции ввода-вывода. Если слишком высокое (более 10), значит за вашим процессором не поспевает диск, и нужно бы поменять его на более быстрый;
- **hi** — процент времени работы аппаратных прерываний. Периферийные устройства могут прерывать работу процессора, чтобы передать какие-то данные;
- **si** — процент времени работы программных прерываний. Некоторые приложения могут прерывать работу процессора, чтобы передать какие-то данные;
- **st** — процент времени, когда реальный процессор не был доступен виртуальной машине. Будет расти если виртуальной машине выделить больше ядер, чем имеет гипервизор. Этот показатель имеет значение только в системах виртуализации.

Ниже видим информацию по потреблению оперативной памяти (**Mem**) и подкачки (**Swap**):

```
MiB Mem :  969,2 total,  416,7 free,  186,7 used,  365,8 buff/cache
MiB Swap: 2048,0 total, 2048,0 free,   0,0 used.  647,0 avail Mem
```

Утилита top – информация о потреблении оперативной памяти

- **total** — общее количество памяти;
- **free** — количество свободной памяти;
- **used** — количество используемой памяти;
- **buff / cache** — память выделенная под буфер и дисковый кэш;
- **avail Mem** — память которая может быть выделена для запуска новых процессов.



Информация по каждому процессу

Ниже показана информация по каждому процессу. Информация разбита на следующие колонки:

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
929	alex	20	0	10476	4000	3296	R	0,3	0,4	0:01.99	top
1	root	20	0	166240	11404	8204	S	0,0	1,1	0:01.19	systemd
2	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kthreadd

top – информация по процессам

- **PID** — номер процесса;
- **USER** — имя пользователя под которым работает процесс;
- **PR** — приоритет процесса;
- **NI** — любезность процесса, чем она выше, тем ниже приоритет;
- **VIR** — общее количество памяти, которое способен адресовать процесс в данный момент времени. Включает в себя **RES**, **SHR**, прикрепленные файлы (например — файлы библиотек). Также включает в себя память, которая была выделена, но не использована;
- **RES** — количество физической памяти, которую использует процесс;
- **SHR** — количество разделяемой памяти, которую использует процесс. Разделяемая память потенциально может быть разделена с другими процессами;
- **S** — состояние процесса (я их уже описывал [здесь](#)):
 - **S (sleeping)** — спящий;
 - **R (running)** — в работе;
 - **Z (zombie)** — зомби;
 - **D (uninterruptible sleep)** — ожидает дискового ввода/вывода, и не принимает никакие сигналы;
 - **T (stopped by job control signal)** — остановлен специальным сигналом;
 - **t (stopped by debugger during trace)** — остановлен в процессе дебага;
 - **I (Idle)** — бездействующий поток ядра.
- **%CPU** — использование сри в процентах;
- **%MEM** — RES в процентах от общего количества памяти;
- **TIME+** — сколько времени с этим процессом работал ЦПУ, с точностью до сотых долей секунды;



- **COMMAND** — с помощью какой команды запустили процесс.

Опции команды top

Теперь пробежимся по основным опциям, с которыми можно запускать **top**.

Количество итераций

Опция **-n** укажет через сколько повторений нужно завершить работу **top**. По умолчанию чтобы выйти из **top** нужно нажать клавишу **q**.

Следующая команда выйдет из **top** через 10 повторений:

```
$ top -n 10
```

Процессы определённого пользователя

Можно заставить **top** отображать процессы определенного пользователя с помощью опции **-u**:

```
$ top -u www-data
```

```
top - 14:10:33 up 1:40, 1 user, load average: 0,00, 0,00, 0,00
Tasks: 99 total, 1 running, 98 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0,0 us, 0,2 sy, 0,0 ni, 99,8 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st
MiB Mem : 969,2 total, 416,4 free, 187,0 used, 365,8 buff/cache
MiB Swap: 2048,0 total, 2048,0 free, 0,0 used, 646,8 avail Mem
```

PID	PPID	UID	USER	RUSER	TTY	TIME+	%CPU	%MEM	S	COMMAND
516	514	33	www-data	www-data	?	0:00.00	0,0	1,1	S	nginx
517	514	33	www-data	www-data	?	0:00.00	0,0	1,1	S	nginx

top – процессы пользователя www-data

Горячие клавиши top

Режим цветности

Если во время работы **top** нажать клавишу **z**, это переключит режим цветности (белое на чёрном или черное на белом):

```
top - 14:11:14 up 1:41, 1 user, load average: 0,00, 0,00, 0,00
Tasks: 99 total, 1 running, 98 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0,2 us, 0,2 sy, 0,0 ni, 99,7 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st
MiB Mem : 969,2 total, 416,4 free, 187,0 used, 365,8 buff/cache
MiB Swap: 2048,0 total, 2048,0 free, 0,0 used, 646,8 avail Mem
```

PID	PPID	UID	USER	RUSER	TTY	TIME+	%CPU	%MEM	S	COMMAND
516	514	33	www-data	www-data	?	0:00.00	0,0	1,1	S	nginx
517	514	33	www-data	www-data	?	0:00.00	0,0	1,1	S	nginx



top – изменение режима цветности

Больше информации о команде (поле COMMAND)

Если во время работы **top** нажать клавишу **c**, это заставит выводить больше информации о команде:

```
top - 14:12:02 up 1:42, 1 user, load average: 0,00, 0,00, 0,00
Tasks: 99 total, 1 running, 98 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0,0 us, 0,0 sy, 0,0 ni,100,0 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st
MiB Mem : 969,2 total, 416,4 free, 187,0 used, 365,8 buff/cache
MiB Swap: 2048,0 total, 2048,0 free, 0,0 used. 646,8 avail Mem
```

PID	PPID	UID	USER	RUSER	TTY	TIME+	%CPU	%MEM	S	COMMAND
516	514	33	www-data	www-data	?	0:00.00	0,0	1,1	S	nginx: worker process
517	514	33	www-data	www-data	?	0:00.00	0,0	1,1	S	nginx: worker process

top – больше информации о команде процесса

Сортировка процессов

По умолчанию процессы отсортированы по колонке **%CPU**. Те что потребляют больше всего находятся сверху.

Чтобы выделить столбец, по которому сейчас идет сортировка, нажмите клавишу **x**.

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
789	alex	20	0	17068	9628	8048	S	0,0	1,0	0:00.07	systemd
790	alex	20	0	169164	3712	16	S	0,0	0,4	0:00.00	(sd-pam)
873	alex	20	0	17296	7944	5508	S	0,0	0,8	0:00.75	sshd
874	alex	20	0	8916	5648	3748	S	0,0	0,6	0:00.08	bash
1020	alex	20	0	10512	3948	3232	R	0,0	0,4	0:00.54	top
451	message+	20	0	8780	4616	4056	S	0,0	0,5	0:00.06	dbus-daemon
1	root	20	0	166240	11404	8204	S	0,0	1,1	0:01.22	systemd
2	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kthreadd
3	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	rcu_par_gp
5	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	netns
7	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker/0:0H-events_highpri
9	root	0	-20	0	0	0	I	0,0	0,0	0:00.11	kworker/0:1H-events_highpri

top – сортировка по колонке USER

Клавишами “<” или “>”, можно выбрать столбец, по которому нужно сортировать процессы.

А если нажать клавишу **R**, то сортировка пойдет в обратном порядке.

Клавиша **M** отсортирует процессы по потреблению памяти (**%MEM**).

Завершение процессов



Из **top** можно завершить процесс, для этого нужно нажать клавишу **k**, далее ввести **pid** процесса, который хотим завершить и нажать **Enter**.

По умолчанию процессу посылается сигнал **15 (sigterm)**. Но завершать чужие процессы можно только запустив **top** под пользователем **root**, или с помощью команды **sudo**.

Сигналы, которые можно посылать процессам, я разберу чуть позже в этом курсе.

Добавление полей

По умолчанию отображаются далеко не все поля. Чтобы добавить поля нажмите клавишу **F**. Затем, с помощью клавиш вверх и вниз, вы можете перемещаться по полям. Чтобы добавить выбранное поле нажмите клавишу **d** или **пробел**. А чтобы выбрать поле для сортировки выделите его и нажмите **s**. Ну и чтобы вернуться в режим просмотра, нажмите **q**.

```

s-ubu:~ - alex@172.28.90.34:22 - Bitvise xterm-256color - alex@ubu-22: ~
Fields Management for window 1:Def, whose current sort field is %CPU
Navigate with Up/Dn, Right selects for move then <Enter> or Left commits,
'd' or <Space> toggles display, 's' sets sort. Use 'q' or <Esc> to end!

* PID      = Process Id          TIME    = CPU Time          RSan     = RES Anonymous (KiB)
* USER    = Effective User Name SWAP    = Swapped Size (KiB)  RSfd     = RES File-based (KiB)
* PR      = Priority            CODE    = Code Size (KiB)    RSlk     = RES Locked (KiB)
* NI      = Nice Value         DATA   = Data+Stack (KiB)  RSSh     = RES Shared (KiB)
* VIRT    = Virtual Image (KiB) nMaj    = Major Page Faults CGNAME    = Control Group name
* RES     = Resident Size (KiB) nMin    = Minor Page Faults NU        = Last Used NUMA node
* SHR     = Shared Memory (KiB) nDRT    = Dirty Pages Count
* S       = Process Status     WCHAN   = Sleeping in Function
* %CPU    = CPU Usage          Flags   = Task Flags <sched.h>
* %MEM    = Memory Usage (RES) CGROUPS = Control Groups
* TIME+   = CPU Time, hundredths SUPGIDS = Supp Groups IDs
* COMMAND = Command Name/Line SUPGRPS = Supp Groups Names
PPID      = Parent Process pid  TGID    = Thread Group Id
UID       = Effective User Id   OOMa    = OOMEM Adjustment
RUID      = Real User Id       OOMS    = OOMEM Score current
RUSER     = Real User Name     ENVIRON  = Environment vars
SUID      = Saved User Id      vMj     = Major Faults delta
SUSER     = Saved User Name    vMn     = Minor Faults delta
GID       = Group Id          USED    = Res+Swap Size (KiB)
GROUP     = Group Name       nsIPC   = IPC namespace Inode
PGRP      = Process Group Id  nsMNT   = MNT namespace Inode
TTY       = Controlling Tty   nsNET   = NET namespace Inode
TPGID     = Tty Process Grp Id nsPID   = PID namespace Inode
SID       = Session Id       nsUSER  = USER namespace Inode
nTH       = Number of Threads nsUTS   = UTS namespace Inode
P         = Last Used Cpu (SMP) LXC      = LXC container name
  
```

top – добавление полей

Итог

Мы разобрали утилиту **top** для мониторинга за потреблением ресурсов **Linux**.

Узнали следующие опции **top**:



- **-n <количество>** — завершить top после нескольких итераций;
- **-u <username>** — показывать процессы определенного пользователя.

Узнали горячие клавиши top:

- **z** — изменить цветность;
- **c** — более полная команда запуска;
- **<** — сортировать по предыдущему столбцу;
- **>** — сортировать по следующему столбцу;
- **R** — обратить сортировку;
- **M** — сортировать по rss;
- **x** — выделить колонку по которой ведётся сортировка;
- **k** — завершить процесс;
- **F** — настроить поля и сортировку.

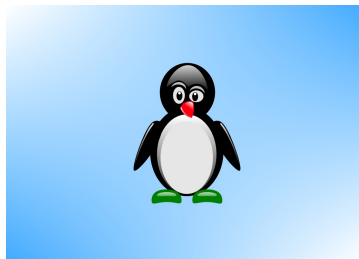
• • •

[Предыдущая статья](#)

[Вернуться к оглавлению](#)

[Следующая статья](#)


Сводка



Имя статьи Утилита top – наблюдаем за процессами

Описание В этой статье будет показана утилита top, с помощью которой можно наблюдать за активностью процессов в реальном времени



[debian](#)[linux](#)[ubuntu](#) admin 19.09.2022 Linux [Leave a Comment](#)[◀ Утилита ps – изучаем процессы](#)[Утилита htop – наблюдаем за процессами ▶](#)

Добавить комментарий

Ваш адрес email не будет опубликован. Обязательные поля помечены *

Комментарий *

Имя *

Email *



Сайт

- ☐ Сохранить моё имя, email и адрес сайта в этом браузере для последующих моих комментариев.

Отправить комментарий

Theme Daily Blog by Creativ Themes

