

# 20 команд мониторинга Linux, которые вы должны знать

September 27, 2023

В этой статье мы рассмотрим 20 команд мониторинга Linux, которые вы должны знать как системный администратор / системный администратор Linux / DevOps / SRE.

## Вступление

Если вы работаете системным администратором / администратором Linux / DevOps / SRE, то, скорее всего, вам придется решать проблемы, связанные с производительностью, в среде Linux.

Давайте рассмотрим некоторые из наиболее часто используемых утилит командной строки Linux для диагностики проблем, связанных с сетью и производительностью.

## 1. Top

Когда мы запустим эту команду, откроется интерактивный командный режим.

Где верхняя половина будет содержать статистику процессов и использования ресурсов.

А нижняя половина содержит список запущенных в данный момент процессов.

top

Нажатие q просто выведет вас из данного режима.

Вывод:

```
top - 09:51:35 up 649 days, 20:12, 2 users, load average: 0.18, 0.26, 0.28
Tasks: 224 total, 1 running, 221 sleeping, 2 stopped, 0 zombie
%Cpu(s): 2.4 us, 0.9 sy, 0.0 ni, 96.7 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 65974528 total, 17793004 free, 5400312 used, 42781212 buff/cache
KiB Swap: 8388604 total, 8353424 free, 35180 used, 50460600 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
18075	root	20	0	7310120	2.3g	1108	S	42.2	3.6	158593.46	dockerd
3697	root	20	0	146044	78012	77724	S	2.3	0.1	8768.58	systemd-journal
6934	root	20	0	1358944	750072	4248	S	2.3	1.1	7275.35	glusterfs
26182	postgres	20	0	409024	20240	7188	S	1.7	0.0	166:22.42	postmaster
4243	root	20	0	162120	2396	1600	R	1.0	0.0	0:00.07	top
19062	root	20	0	727784	18024	5276	S	1.0	0.0	5178.50	glusterd
19004	root	20	0	3331988	2.6g	45476	S	0.7	4.1	1129:10	rsyslogd
7059	root	20	0	227092	5376	4956	S	0.3	0.0	1946:16	vmtoolsd
20576	postgres	20	0	251872	2292	732	S	0.3	0.0	39:15.97	postmaster
1	root	20	0	192084	4044	2612	S	0.0	0.0	500:51.92	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:56.82	kthreadd
3	root	20	0	0	0	0	S	0.0	0.0	3:32.71	ksoftirqd/0
5	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworker/0:0H
6	root	20	0	0	0	0	S	0.0	0.0	496:19.38	kworker/u32:0
8	root	rt	0	0	0	0	S	0.0	0.0	50:06.98	migration/0
9	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_bh
10	root	20	0	0	0	0	S	0.0	0.0	1542:09	rcu_sched
11	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	lru-add-drain
12	root	rt	0	0	0	0	S	0.0	0.0	14:58.73	watchdog/0
13	root	rt	0	0	0	0	S	0.0	0.0	47:24.96	watchdog/1
14	root	rt	0	0	0	0	S	0.0	0.0	454:46.73	migration/1
15	root	20	0	0	0	0	S	0.0	0.0	4:19.25	ksoftirqd/1
17	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworker/1:0H
18	root	rt	0	0	0	0	S	0.0	0.0	31:31.04	watchdog/2
19	root	rt	0	0	0	0	S	0.0	0.0	229:56.02	migration/2
20	root	20	0	0	0	0	S	0.0	0.0	3:21.03	ksoftirqd/2
22	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworker/2:0H
23	root	rt	0	0	0	0	S	0.0	0.0	17:58.48	watchdog/3
24	root	rt	0	0	0	0	S	0.0	0.0	90:18.59	migration/3
25	root	20	0	0	0	0	S	0.0	0.0	3:30.98	ksoftirqd/3
27	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworker/3:0H
28	root	rt	0	0	0	0	S	0.0	0.0	13:10.23	watchdog/4
29	root	rt	0	0	0	0	S	0.0	0.0	92:36.00	migration/4

## 2. vmstat

Команда vmstat – это утилита Linux для отображения статистики, связанной с потреблением памяти, использования диска и другой системной информацией.

vmstat не должен присутствовать в вашей системе Linux, но не беспокойтесь.

Мы можем легко установить пакет «sysstat»

vmstat

Вывод:

```
[root@DevSexOps ~]# vmstat
procs -----memory----- --swap-- -----io----- -system-- -----cpu-----
r b swpd free buff cache si so bi bo in cs us sy id wa st
1 0 1288 675828 4172 2525444 0 0 0 2 1 4 0 0 100 0
0
[root@DevSexOps ~]#
```

### 3. iostat

Команда `iostat` на Linux позволяет отслеживать статистику использования ЦП и ввода-вывода (I/O) для всех дисков и файловых систем.

Команды `iostat` полезны для изменения конфигурации системы, чтобы лучше сбалансировать нагрузку ввода-вывода между физическими дисками.

Эта команда в основном используется системными администраторами Linux.

`iostat`

Вывод:

```
[root@ ~] # iostat
Linux 3.10.0-957.12.2.el7.x86_64 (s00lnd-elkib01) 03/29/2021 _x86_64_ (16 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           3.59    0.00    1.07    0.01    0.00   95.33

Device:            tps    kB_read/s    kB_wrtn/s    kB_read    kB_wrtn
fd0                 0.00         0.00         0.00         24         0
sda                 3.17         0.04         32.66    2193150    1833724732
dm-0                6.35         0.04         32.66    2148397    1833565172
dm-1                0.00         0.00         0.00      26680     159376
sdb                 0.22         0.60         3.53    33721876    198217670
dm-2                0.00         0.00         0.00      2460         4
dm-3                0.00         0.00         0.15     6068     8211744
```

### 4. iostat -d

Команда `iostat -d` используется для мониторинга загрузки устройств ввода / вывода системы, отслеживая время активности устройств в зависимости от их средней скорости передачи данных.

Команда `iostat -d` используется для отображения отчетов об использовании устройств.

Вывод:

```
[root@s00lnd-elkib01 ~] # iostat -d
Linux 3.10.0-957.12.2.el7.x86_64 03/29/2021 _x86_64_ (16 CPU)

Device:            tps    kB_read/s    kB_wrtn/s    kB_read    kB_wrtn
fd0                 0.00         0.00         0.00         24         0
sda                 3.17         0.04         32.66    2193158    1833730600
dm-0                6.35         0.04         32.66    2148405    1833571040
dm-1                0.00         0.00         0.00      26680     159376
sdb                 0.22         0.60         3.53    33721876    198217670
dm-2                0.00         0.00         0.00      2460         4
dm-3                0.00         0.00         0.15     6068     8211744
```

## 5. lsof

Задача команды lsof – «перечислить открытые файлы» в системе.

Открытый файл не означает pdf или текстовый файл, он включает файлы на диске или каналы, используемые процессами в фоновом режиме.

Эта команда – удобный инструмент для отладчиков операционной системы и системных администраторов.

lsof

Вывод:3

```
ss:main 19004 19007 root 33c REG 0,19 134217728 1351197255 /run/log/journal/28b71d5e86db477eade93490fb2dbc14/system8e54dd01a615474695ec5
salfdr29-000000009e1664d-0008b7f4cc07db6.journal
glusterd 19062 root cwd DIR 253,0 4096 2 /
glusterd 19062 root rwd DIR 253,0 4096 2 /
glusterd 19062 root txt REG 253,0 35372 1882249 /usr/sbin/glusterfsd
glusterd 19062 root mem REG 253,0 88776 1576293 /usr/lib64/libgcc_s-4.8.5-20150702.so.1
glusterd 19062 root mem REG 253,0 61624 1582842 /usr/lib64/libnss_files-2.17.so
glusterd 19062 root mem REG 253,0 402384 1577036 /usr/lib64/libpcrt.so.1.2.0
glusterd 19062 root DEL REG 253,0 1582842 /usr/lib64/libswellinux.so.1.605098c5
glusterd 19062 root mem REG 253,0 105824 1582850 /usr/lib64/libresolv-2.17.so
glusterd 19062 root mem REG 253,0 15688 1577145 /usr/lib64/libkeyutils.so.1.6
glusterd 19062 root DEL REG 253,0 1582850 /usr/lib64/libkbsupport.so.0.1.605098c5
glusterd 19062 root DEL REG 253,0 1577389 /usr/lib64/libkbscrypto.so.3.1.605098c5
glusterd 19062 root DEL REG 253,0 1576963 /usr/lib64/libbcom_err.so.2.1.605098c5
glusterd 19062 root DEL REG 253,0 1577385 /usr/lib64/libkrt5.so.3.3.605098c5
glusterd 19062 root DEL REG 253,0 1577173 /usr/lib64/libkrtapi.krt5.so.2.2.605098c5
glusterd 19062 root DEL REG 253,0 1582899 /usr/lib64/libsal.so.1.0.2k:605098c5
glusterd 19062 root mem REG 253,0 129368 1586140 /usr/lib64/glusterfs/6.1/rpc-transport/socket.so
glusterd 19062 root mem REG 253,0 19240 1586139 /usr/lib64/librpc-common.so.6.0.0
glusterd 19062 root mem REG 253,0 157424 1577406 /usr/lib64/liblma.so.5.2.2
glusterd 19062 root mem REG 253,0 41088 1586156 /usr/lib64/librpc-ods.so.6.0.0
glusterd 19062 root mem REG 253,0 32528 1586154 /usr/lib64/librpc-tp.so.6.0.0
glusterd 19062 root mem REG 253,0 1509376 1577414 /usr/lib64/libxml2.so.2.9.1
glusterd 19062 root mem REG 253,0 2964208 1707262 /usr/lib64/glusterfs/6.1/xlator/mgmt/glusterd.so
glusterd 19062 root mem REG 253,0 106075056 1576605 /usr/lib/locale/locale-archive
glusterd 19062 root mem REG 253,0 2151672 1576577 /usr/lib64/libc-2.17.so
glusterd 19062 root DEL REG 253,0 1577145 /usr/lib64/libcrypto.so.1.0.2k:605098c5
glusterd 19062 root mem REG 253,0 141968 1576603 /usr/lib64/libpthread-2.17.so
glusterd 19062 root mem REG 253,0 19288 1580239 /usr/lib64/libdl-2.17.so
glusterd 19062 root mem REG 253,0 43774 1582852 /usr/lib64/librt-2.17.so
glusterd 19062 root mem REG 253,0 1137024 1580243 /usr/lib64/libm-2.17.so
glusterd 19062 root mem REG 253,0 20112 1577466 /usr/lib64/libuuid.so.1.3.0
glusterd 19062 root DEL REG 253,0 1576872 /usr/lib64/libz.so.1.2.7:605098c5
glusterd 19062 root mem REG 253,0 126680 1582097 /usr/lib64/libgfs.so.0.0.1
glusterd 19062 root mem REG 253,0 257088 1582095 /usr/lib64/libgfrpc.so.0.0.1
glusterd 19062 root mem REG 253,0 1128400 1582099 /usr/lib64/libglusterfs.so.0.0.1
glusterd 19062 root mem REG 253,0 163400 1576286 /usr/lib64/lib-2.17.so
glusterd 19062 root mem REG 253,0 24254 1576193 /usr/lib64/gconv/gconv-modules.cache
glusterd 19062 root 0r CHR 1,3 0t0 1047 /dev/null
glusterd 19062 root 1w CHR 1,3 0t0 1047 /dev/null
```

## 6. tcpdump

Tcpdump – это инструмент, который используется для анализа пакетов TCP / IP.

Эта команда обычно используется для анализа трафика в Linux, а также во многих других операционных системах.

tcpdump позволяет прослушивать весь входящий и исходящий трафик со всех интерфейсов.

Что еще более важно, он может фильтровать трафик по интерфейсу, хосту, месту назначения или хосту-источнику, типу трафика и многим другим критериям.

Мы также можем сохранить захваченные пакеты в файл для последующего анализа.

tcpdump

## 8. netstat -s

Команда netstat -s выводит сетевую статистику, такую как общее количество полученных и переданных пакетов по типу протокола и так далее.

Чтобы вывести статистику только избранных протоколов, таких как TCP или UDP, используйте соответствующие параметры, такие как t и u, вместе с параметром s.

netstat -s

```
[root@ # netstat -s
Ip:
 6334695268 total packets received
 8287896 forwarded
 0 incoming packets discarded
 6326392731 incoming packets delivered
 5971461766 requests sent out
 1 dropped because of missing route
Icmp:
 2748455 ICMP messages received
 24207 input ICMP message failed.
 ICMP input histogram:
   destination unreachable: 2743770
   echo requests: 4639
   echo replies: 46
 8074011 ICMP messages sent
 0 ICMP messages failed
 ICMP output histogram:
   destination unreachable: 8069341
   echo request: 48
   echo replies: 4622
IcmpMsg:
 InType0: 46
 InType3: 2743770
 InType8: 4639
 OutType0: 4622
 OutType3: 8069341
 OutType8: 48
Tcp:
 420555353 active connections openings
 120076140 passive connection openings
 260497958 failed connection attempts
 29830 connection resets received
 19 connections established
 5313029191 segments received
 5794424164 segments send out
 4253727 segments retransmitted
 224607 bad segments received.
 1462445392 resets sent
```

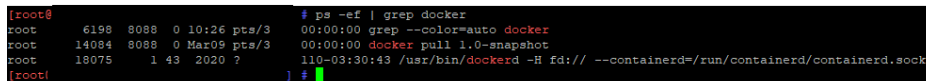
## 9. ps -ef | grep PID

Команда ps выводит четыре столбца информации для двух минимальных процессов, запущенных в текущей оболочке.

Параметр -e указывает ps отображать все процессы.

Параметр `-f` означает полный список, который предоставляет подробную информацию о процессах.

```
ps -ef | grep docker
```



```
root@ 6198 8088 0 10:26 pts/3 $ ps -ef | grep docker
root 6198 8088 0 10:26 pts/3 00:00:00 grep --color=auto docker
root 14084 8088 0 Mar09 pts/3 00:00:00 docker pull 1.0-snapshot
root 18075 1 43 2020 ? 110-03:30:43 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
[root@ 1 #
```

## 10. nethogs

NetHogs – это инструмент для небольших сетей.

Вместо того, чтобы разбивать трафик по протоколам или подсети, как это делают большинство таких инструментов, он группирует полосу пропускания по процессам и не зависит от загрузки специального модуля ядра.

Если внезапно возникает большой сетевой трафик, вы можете запустить NetHogs и сразу увидеть, какой PID вызывает это, а если это какой-то странный процесс, убить его.

```
nethogs
```

## 11. mpstat

Использование команды `mpstat` отобразит среднюю глобальную активность всех процессоров.

Мы можем отображать общую статистику ЦП по системе или по процессору с помощью команды `mpstat`.

```
mpstat
```

Вывод:

```
[root@ ~]# ps -ef | grep docker
root      6198   8088   0 10:26 pts/3    00:00:00 grep --color=auto docker
root     14084   8088   0 Mar09 pts/3    00:00:00 docker pull 1.0-snapshot
root     18075     1 43 2020 ?        110-03:30:43 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
[root@ ~]#
```

## 12. free -m

Free -m – это команда, которая отображает используемую память в нашей системе.

Команда Free -m также отображает доступную память!

free -m

Вывод:

```
[root@ ~]# free -m
              total        used         free       shared    buff/cache   available
Mem:          64428         5292        17347          3260         41788         49274
Swap:          8191           34          8157
```

## 13. uptime

Команда uptime показывает, как долго работает система.

Она также показывает, сколько пользователей вошли в систему.

Вывод:

uptime

11:23:38 up 649 days, 21:44, 2 users, load average: 0.15, 0.15, 0.20

## 14. ps -e

Команда ps помогает просматривать сведения о запущенных в данный момент процессах.

Мы также можем убить или завершить процессы, которые не работают нормально.

В выводе будут перечислены все запущенные процессы, а также те, которые запущены другими пользователями.

ps -e

PID	TTY	TIME	CMD
1	?	08:20:55	systemd
2	?	00:00:56	kthreadd
3	?	00:03:32	ksoftirqd/0
5	?	00:00:00	kworker/0:0H
6	?	08:16:19	kworker/u32:0
8	?	00:50:06	migration/0
9	?	00:00:00	rcu_bh
10	?	1-01:42:18	rcu_sched
11	?	00:00:00	lru-add-drain
12	?	00:14:59	watchdog/0
13	?	00:47:25	watchdog/1
14	?	07:34:47	migration/1
15	?	00:04:19	ksoftirqd/1
17	?	00:00:00	kworker/1:0H
18	?	00:31:31	watchdog/2
19	?	03:49:56	migration/2
20	?	00:03:21	ksoftirqd/2
22	?	00:00:00	kworker/2:0H
23	?	00:17:58	watchdog/3
24	?	01:30:18	migration/3
25	?	00:03:30	ksoftirqd/3
27	?	00:00:00	kworker/3:0H
28	?	00:13:10	watchdog/4
29	?	01:32:36	migration/4
30	?	00:23:33	ksoftirqd/4
32	?	00:00:00	kworker/4:0H
33	?	00:13:13	watchdog/5
34	?	01:19:46	migration/5
35	?	00:23:30	ksoftirqd/5

## 15. ac

Команда ac отображает отчет о времени соединения (часах) этой системы на основе времени входа и выхода.

Хранится в файле wtmp.

ac

## 16. ac -d

Команда ac -d выводит общее количество за каждый день, а не только одно общее количество.

ac -d



## 17. ac -p

Команда `ac -p` выводит общее время для каждого пользователя в дополнение ко всему.

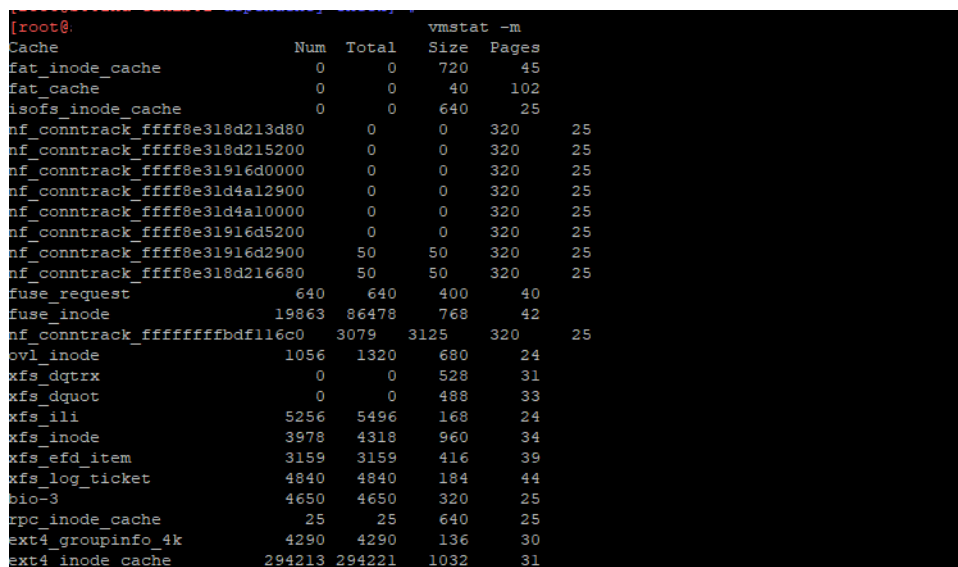
`ac -p`

## 18. vmstat -m

Инструмент `vmstat` используется для мониторинга использования виртуальной памяти системы.

Команда `vmstat -m` отображает информацию о вашей виртуальной машине.

`vmstat -m`



```
[root@ ~]# vmstat -m
```

	Num	Total	Size	Pages
Cache				
fat_inode_cache	0	0	720	45
fat_cache	0	0	40	102
isoofs_inode_cache	0	0	640	25
nf_conntrack_ffff8e318d213d80	0	0	320	25
nf_conntrack_ffff8e318d215200	0	0	320	25
nf_conntrack_ffff8e31916d0000	0	0	320	25
nf_conntrack_ffff8e31d4a12900	0	0	320	25
nf_conntrack_ffff8e31d4a10000	0	0	320	25
nf_conntrack_ffff8e31916d5200	0	0	320	25
nf_conntrack_ffff8e31916d2900	50	50	320	25
nf_conntrack_ffff8e318d216680	50	50	320	25
fuse_request	640	640	400	40
fuse_inode	19863	86478	768	42
nf_conntrack_ffffffffffbdf116c0	3079	3125	320	25
ovl_inode	1056	1320	680	24
xfs_dqtrx	0	0	528	31
xfs_dquot	0	0	488	33
xfs_ili	5256	5496	168	24
xfs_inode	3978	4318	960	34
xfs_efd_item	3159	3159	416	39
xfs_log_ticket	4840	4840	184	44
bio-3	4650	4650	320	25
rpc_inode_cache	25	25	640	25
ext4_groupinfo_4k	4290	4290	136	30
ext4_inode_cache	294213	294221	1032	31

## 19. vmstat -d

`Vmstat -d` отображает статистику диска виртуальной памяти нашей системы.

Инструмент `vmstat -d` используется для мониторинга использования виртуальной памяти системой.

Вывод:

```
[root@ ~]# vmstat -d
disk- -----reads----- --vmstat -d-- -----writes----- ----IO-----
      total merged sectors      ms total merged sectors      ms      cur      sec
fd0          6         0        48      205         0         0         0         0         0
sr0          0         0         0         0         0         0         0         0         0
sda 154282      4719 4390476 625347 178088731 178488832 3668137912 246032583         0 90257
dm-0 153071         0 4300970 617235 356537685         0 3667818792 513798119         0 90839
dm-1   5649         0   53360  46509  39844         0 318752 2798904         0    10
sdb 1763555        513 67443753 1738030 10592836 1197426 396435373 32338075         0 5168
dm-2    90         0   4920    640         2         0         9         3         0
dm-3   401         0   12137   1262  57126         0 16423520 37039956         0   125
```

## 20. pstree

Команда pstree показывает структуру наследования процессов.

Команда pstree отображает ту команду, которая является дочерней по отношению к другой команде.