

# Администрирование Linux:

# Производительность системы

## Часть 2



Александр  
Зубарев



## **Александр Зубарев**

Председатель цикловой комиссии “Информационной  
безопасности инфокоммуникационных систем”

АКТ (ф) СПбГУТ



[Александр Зубарев](#)

---

# Предисловие

**На этом занятии мы поговорим о:**


- о ресурсах системы;
- об устройствах;
- виртуальной памяти;
- различных работ накопителей.

**По итогу занятия** вы узнаете как устройства взаимосвязи с ядром операционной системы и как это влияет на производительность.

---

# План занятия

1. [Предисловие](#)
2. [Производительность системы](#)
3. [Методы для ускорения системы](#)
4. [Утилиты мониторинга производительности](#)
5. [Утилита vmstat](#)
6. [Итоги](#)
7. [Домашнее задание](#)



# **Производительность системы**

# Производительность системы

Операционная система при своей работе сохраняет всю информацию работы устройств в директории [/proc](#).

Устройство в системе рассматривается как блочное устройство с системой ввода и вывода.

Устройства, как правило, рассматриваются файлом через который можно считывать или куда можно записывать информацию.

→ Находится данные файлы в директории [/dev](#).

Как видно из примера, некоторые файлы ссылаются на директорию [/proc](#).

# Производительность системы

```
crw-rw---- 1 root cdrom    11,   0 Jan 25 13:28 sr0
lrwxrwxrwx  1 root root      15 Jan 25 13:28 stderr -> /proc/self/fd/2
lrwxrwxrwx  1 root root      15 Jan 25 13:28 stdin  -> /proc/self/fd/0
lrwxrwxrwx  1 root root      15 Jan 25 13:28 stdout -> /proc/self/fd/1
crw-rw-rw-  1 root tty       5,    0 Jan 25 13:28 tty
crw-rw-rw-  1 root tty       4,    0 Jan 25 13:28 tty0
```

# Пример работы с /proc

## Пример:

Мы хотим просмотреть точки монтирования, режим работы, устройство и т.д.

```
root@debian:~# mount
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
udev on /dev type devtmpfs (rw,nosuid,relatime,size=490712k,nr_inodes=122678,mode=755)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,noexec,relatime,size=101108k,mode=755)
/dev/sda1 on / type ext4 (rw,relatime,errors=remount-ro)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)
tmpfs on /run/lock type tmpfs (rw,nosuid,nodev,noexec,relatime,size=5120k)
tmpfs on /sys/fs/cgroup type tmpfs (ro,nosuid,nodev,noexec,mode=755)
cgroup2 on /sys/fs/cgroup/unified type cgroup2 (rw,nosuid,nodev,noexec,relatime,nsdelegate)
cgroup on /sys/fs/cgroup/systemd type cgroup (rw,nosuid,nodev,noexec,relatime,xattr,name=systemd)
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime)
bpf on /sys/fs/bpf type bpf (rw,nosuid,nodev,noexec,relatime,mode=700)
cgroup on /sys/fs/cgroup/rdma type cgroup (rw,nosuid,nodev,noexec,relatime,rdma)
cgroup on /sys/fs/cgroup/cpu,cpuacct type cgroup (rw,nosuid,nodev,noexec,relatime,cpu,cpuacct)
cgroup on /sys/fs/cgroup/blkio type cgroup (rw,nosuid,nodev,noexec,relatime,blkio)
cgroup on /sys/fs/cgroup/freezer type cgroup (rw,nosuid,nodev,noexec,relatime,freezer)
cgroup on /sys/fs/cgroup/devices type cgroup (rw,nosuid,nodev,noexec,relatime,devices)
cgroup on /sys/fs/cgroup/memory type cgroup (rw,nosuid,nodev,noexec,relatime,memory)
cgroup on /sys/fs/cgroup/perf_event type cgroup (rw,nosuid,nodev,noexec,relatime,perf_event)
cgroup on /sys/fs/cgroup/net_cls,net_prio type cgroup (rw,nosuid,nodev,noexec,relatime,net_cls,net_prio)
cgroup on /sys/fs/cgroup/pids type cgroup (rw,nosuid,nodev,noexec,relatime,pids)
cgroup on /sys/fs/cgroup/cpuset type cgroup (rw,nosuid,nodev,noexec,relatime,cpuset)
systemd-1 on /proc/sys/fs/binfmt_misc type autofs (rw,relatime,fd=36,pgrp=1,timeout=0,minproto=5,direct,pipe_ino=10169)
mqueue on /dev/mqueue type mqueue (rw,relatime)
hugetlbfs on /dev/hugepages type hugetlbfs (rw,relatime,pagesize=2M)
debugfs on /sys/kernel/debug type debugfs (rw,relatime)
tmpfs on /run/user/0 type tmpfs (rw,nosuid,nodev,relatime,size=101104k,mode=700)
```



## Пример работы с /proc

Перейдя в папку `/proc` и просмотрев папку можно увидеть следующий вывод

```
root@debian:/proc# cat mounts
sysfs /sys sysfs rw,nosuid,nodev,noexec,relatime 0 0
proc /proc proc rw,nosuid,nodev,noexec,relatime 0 0
udev /dev devtmpfs rw,nosuid,relatime,size=490712k,nr_inodes=122678,mode=755 0 0
devpts /dev/pts devpts rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000 0 0
tmpfs /run tmpfs rw,nosuid,noexec,relatime,size=101108k,mode=755 0 0
```

Формально все настройки и состояния системы находятся в данной папке.

# Накопители информации

Компонент системы нужный для хранения информации.

Все программы: как и операционная система, так и ядро, – загружается с накопителя.

Чем быстрее шина, чем быстрее чтения и запись данных тем производительней система.

Кроме этого на производительность может влиять использование технологии RAID.

---

# Виды массивов

- RAID 0 — дисковый массив повышенной производительности с чередованием, без отказоустойчивости;
- RAID 1 — зеркальный дисковый массив;
- RAID 2 — зарезервирован для массивов, которые применяют код Хемминга;
- RAID 3 и RAID 4 — дисковые массивы с чередованием и выделенным диском чётности;
- RAID 5 — дисковый массив с чередованием и отсутствием выделенного диска чётности.
- ....

# Виды массивов


- ...
- RAID 6 — дисковый массив с чередованием, использующий две контрольные суммы, вычисляемые двумя независимыми способами;
- RAID 10 — массив RAID 0, построенный из массивов RAID 1;
- RAID 1E — зеркало из трёх устройств:
  - RAID 50 — массив RAID 0 из массивов RAID 5;
  - RAID 05 — RAID 5 из RAID 0;
  - RAID 60 — RAID 0 из RAID 6 и различные другие.
- RAID-Z — один избыточный диск.

# Утилита Free

Без параметров возвращает результаты для «total», «used», and «free» объема памяти в вашей системе благодаря доступу к информации из ядра Linux.

Она также отображает категории для «shared», «buff / cache», и «available».

	total	used	free	shared	buff/cache	available
Mem:	4041532	888044	2400568	24064	752920	2895796
Swap:	998396	0	998396			



# Методы для ускорения системы

# Планировщик заданий

**Cron** – планировщик заданий, используемый для планирования выполнения команд на определённое время.

Примерный шаблон задания для Cron:

Минуты (0-59) Часы (0-24) День (1-31) Месяц (1-12)

День недели (0-6) Команда

*Для отображения содержимого **crontab-файла** текущего пользователя используйте команду:*

**crontab -l**

*Для редактирования заданий пользователя:*

**crontab -e**

# Работа с памятью компьютера

Обычная ситуация:

все тормозит, что-то не работает, не загружается Xorg...

→ Консоль запустится в любом случае.

*Выполняем:*

`df`

→ Видим отсутствия памяти на диске.

*Выполняем:*

`sudo apt-get autoremove`

→ Поможет освободить немного места.

*Операция полезна и для профилактики.*



## Работа кэшем пакетов

Другой метод это выполнить команду:

```
user@NetaLog:~$ du -sh /var/log/apt
148K    /var/log/apt
user@NetaLog:~$
```

В процессе эксплуатации данное значение будет увеличиваться.

*Выполняем:*

```
rm -r /var/log/apt/*
```

# cpufreq

В сборки различных систем входит пакет **cpufreq**, который отвечает за управлением питания процессора.

*Просмотреть значения частоты процессора:*

**cat /proc/cpuinfo**

```
root@Netalogy:/home/user# cat /proc/cpuinfo |grep MHz
cpu MHz          : 2294.770
cpu MHz          : 2294.770
root@Netalogy:/home/user# cat /proc/cpuinfo
processor         : 0
vendor_id        : GenuineIntel
cpu family       : 6
model            : 58
model name       : Intel(R) Core(TM) i7-3615QM CPU @ 2.30GHz
stepping         : 9
microcode        : 0x19
cpu MHz          : 2294.770
cache size       : 6144 KB
physical id      : 0
siblings         : 2
core id          : 0
cpu cores        : 2
```

# cpufreq

Данный пакет позволяет управлять скоростью процессора, настройки находятся по адресу:

```
cat /sys/devices/system/cpu/cpu1/cpufreq/scaling_governor
```

Установочные ключи для работы демона:

- Ключ `powersave` – говорит энергосбережение;
- Ключ `performance` – выставляет максимальное значение.

# Ускорение swap

Бывает так что оперативной памяти достаточно много, но по какой-то причине задействуется swap.

→ Данные манипуляции ведут к уменьшению производительности.

```
cat /proc/sys/vm/swappiness
```

Позволяет увидеть распределение между оперативной памятью и swap.

# Ускорение swar

Исправить данную историю можно изменив параметры ядра:


```
sudo sysctl vm.swappiness= x,
```

Где:

- x – значение, которое будет в процентом отношении выполняться:
  - 0 – swar не будет использоваться;
  - 100 – оперативная память практически не будет использоваться.

*Применим данные настройки в ядро системы:*

```
sudo sysctl -p
```



# **Утилиты мониторинга производительности**

# Утилиты мониторинга производительности

Утилиты (службы) мониторинга производительности, которые будут рассмотрены сегодня:

- **iostat** – мониторит использования дисковых разделов;
- **vmstat** – выводит статистику процессора, памяти и о процессах;

# Утилита iostat

Утилита, предназначенная для мониторинга использования дисковых разделов.

```
root@Netalogy:/home/user# iostat
Linux 4.19.0-13-amd64 (Netalogy)          02/02/2021          _x86_64_          (2 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           2.33    0.01    1.08    0.42    0.00   96.16

Device            tps    kB_read/s    kB_wrtn/s    kB_read    kB_wrtn
sda                30.10        643.14        246.08     1284503     491488
```



# Утилита iostat

Рассмотрим наиболее полезные ключи:

- **-d** — отображать только использование дисков;
- **-c** — отобразить только использование CPU;
- **-j** — отобразить имя раздела;

ID | LABEL | PATH | UUID

- **-k** — отобразить данные в килобайтах;
- ...

# Утилита iostat

Рассмотрим наиболее полезные ключи:

- **-m** — отобразить данные в мегабайтах;
- **-p** — отобразить статистику по указанному блочному устройству;
- **-t** — отобразить время выполнения теста;
- **-x** — отобразить расширенную статистику.

# Пример использования

## Ситуация:

падает производительность, а именно все жутко тормозит.

## Решение:

- Запускаем `iostat`;
- Смотрим колонку `wait`:
  - если значение 100, то значит:
    - проблема с диском;
    - `idle` может быть близка к 0.
- Запускает утилиту `iostat` и смотрим, в чем причина.

# Пример использования

Current DISK READ:		0.00 B/s		Current DISK WRITE:		63.39 K/s	
TID	PRI	IO USER	DISK READ	DISK WRITE	SWAPIN	IO>	COMMAND
177	be/3	root	0.00 B/s	7.04 K/s	0.00 %	14.70 %	[jbd2/sda1-8]
729	be/4	mysql	0.00 B/s	0.00 B/s	0.00 %	0.01 %	mysqld
922	be/4	mysql	0.00 B/s	7.04 K/s	0.00 %	0.01 %	mysqld
1	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	init
2	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[kthreadd]
3	be/0	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[rcu_gp]
4	be/0	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[rcu_par_gp]
6	be/0	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[kworker/~H-kblockd]



# Утилита vmstat

# Утилита vmstat

Позволяет вывести информацию об использовании памяти, дисков, процессора.

```
root@Netalogy:/home/user# 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  134400  162000  22200 212180     7   39   232   89   107  144   2   1  97   0   0
```

# Поля вывода программы

## Procs:

- **r** — количество процессов в очереди на выполнение процессором;

Если значение  $> 0$  — налицо нагрузка на процессор.

- **b** — количество процессов, ожидающих операций I/O.

Если значение  $> 0$  — налицо нагрузка на диски и/или файловую систему.

---

# Поля вывода программы

## Memory

- **swpd** — количество блоков, перемещённых в swar;
- **free** — свободная память;
- **buff** — буферы памяти;
- **cache** — кеш.

## Swap

- **si** (swar in) — количество блоков в секунду, которое система считывает из раздела или файла swar в память;
- **so** (swar out) — и наоборот, количество блоков в секунду, которое система перемещает из памяти в swar.



---

# Поля вывода программы

## IO

- **bi** (blocks in) — количество блоков в секунду, считанных с диска;
- **bo** (blocks out) — количество блоков в секунду, записанных на диск.

## System

- **in** (interrupts) — количество прерываний в секунду;
- **cs** (context switches) — количество переключений между задачами.

---

# Поля вывода программы

## CPU

- **us** (user time) — % времени CPU, занятый на выполнение «пользовательских» задач;
- **sy** (system time) — % времени CPU, занятый на выполнение задач ядра;
- **id** (idle) — % времени в бездействии;
- **wa** — % времени CPU, занятый на ожидание операций I/O.

# Опции vmstat

- **-a** — переключение колонки memory в **active / inactive** значение вместо **buff / cache**:
  - **active** — память, реально используемая в данный момент определённым процессом,
  - **inactive** — память, выделенная ранее какому-то процессу, которого уже нет.
- **-f** — информация о количестве системных вызовов **fork(2)**, **vfork(2)** и **rfork(2)**, выполненных с момента старта системы, и количество страниц виртуальной памяти, задействованных каждым из них;
- **-m** — использование динамической памяти ядра, выделенной с использованием **malloc(9)**;
- ...

## Опции vmstat

- **-n** — задаёт интервал выполнения в секундах;
- **-s** — изменение вида отображение результатов (в две колонки);
- **-d** — статистика использования дисков;
- **-w** — увеличивает визуальный размер поля для больших объемов;
- **-p [partition name]** — с указанием имени раздела, информация по конкретном разделу (read / writes);
- **-S [k | K | m | M]** — с указанием в чем выводить информацию (килобиты, килобайты, мегабиты, мегабайты соответственно);
- **-V** — версия **vmstat**.



# Итоги



## Итоги

Рассмотренные утилиты и службы не только позволяют просмотреть значения параметров и характеристик, но и увеличить производительность.

---

# Итоги

## Важно запомнить:

- процессор;
  - виртуальная память;
  - физическая память;
  - приоритеты выполнения процессов;
  - процессы;
  - компьютерная сеть;
- все эти параметры нужно контролировать для оценки производительности.

---

# Домашнее задание

Давайте посмотрим ваше [домашнее задание](#).

- Вопросы по домашней работе задавайте **в чате** мессенджера Slack.
- Задачи можно сдавать **по частям**.
- Зачёт по домашней работе проставляется после того, как **приняты все задачи**.



**Задавайте вопросы и  
пишите отзыв о лекции!**

**Александр Зубарев**