

Администрирование Linux: Производительность системы

Часть 1



Александр
Зубарев



Александр Зубарев

Председатель цикловой комиссии “Информационной
безопасности инфокоммуникационных систем”

АКТ (ф) СПбГУТ



[Александр Зубарев](#)

Предисловие

На этом занятии мы поговорим о:


- о производительности систем
- о процессах;
- о ресурсах компонентов системы;
- утилитах позволяющих выполнить сбор данных.

По итогу занятия вы узнаете как работает ядро операционной системы с процессами и как это влияет на производительность.



План занятия

1. [Предисловие](#)
2. [Производительность системы](#)
3. [Утилиты мониторинга производительности](#)
4. [Утилита top](#)
5. [Параметры, отвечающие за производительность](#)
6. [Утилита atop](#)
7. [Пакет sysstat](#)
8. [Итоги](#)
9. [Домашнее задание](#)



Производительность системы

Производительность системы

Производительность системы – количественные характеристики скорости выполнения определенных операций устройств и программ таких как:

- ядро операционной системы;
- процессы и программы;
- оперативная память;
- центральные процессоры;
- накопители информации.

Сегодня наша задача – выполнить общий обзор утилит для мониторинга производительности и рассмотреть утилиты для работы с процессами.

Ядро операционной системы

- обрабатывает прерывания от устройств;
- выполняет запросы системных процессов и пользовательских приложений;
- распределяет виртуальную память;
- создает и уничтожает процессы;
- обеспечивает многозадачность посредством переключения между ними;
- содержит драйверы устройств;
- обслуживает файловую систему.

Ядро операционной системы

- Чем эффективней настроено ядро;
 - Чем эффективней работает ядро;
- ➔ Тем производительней система

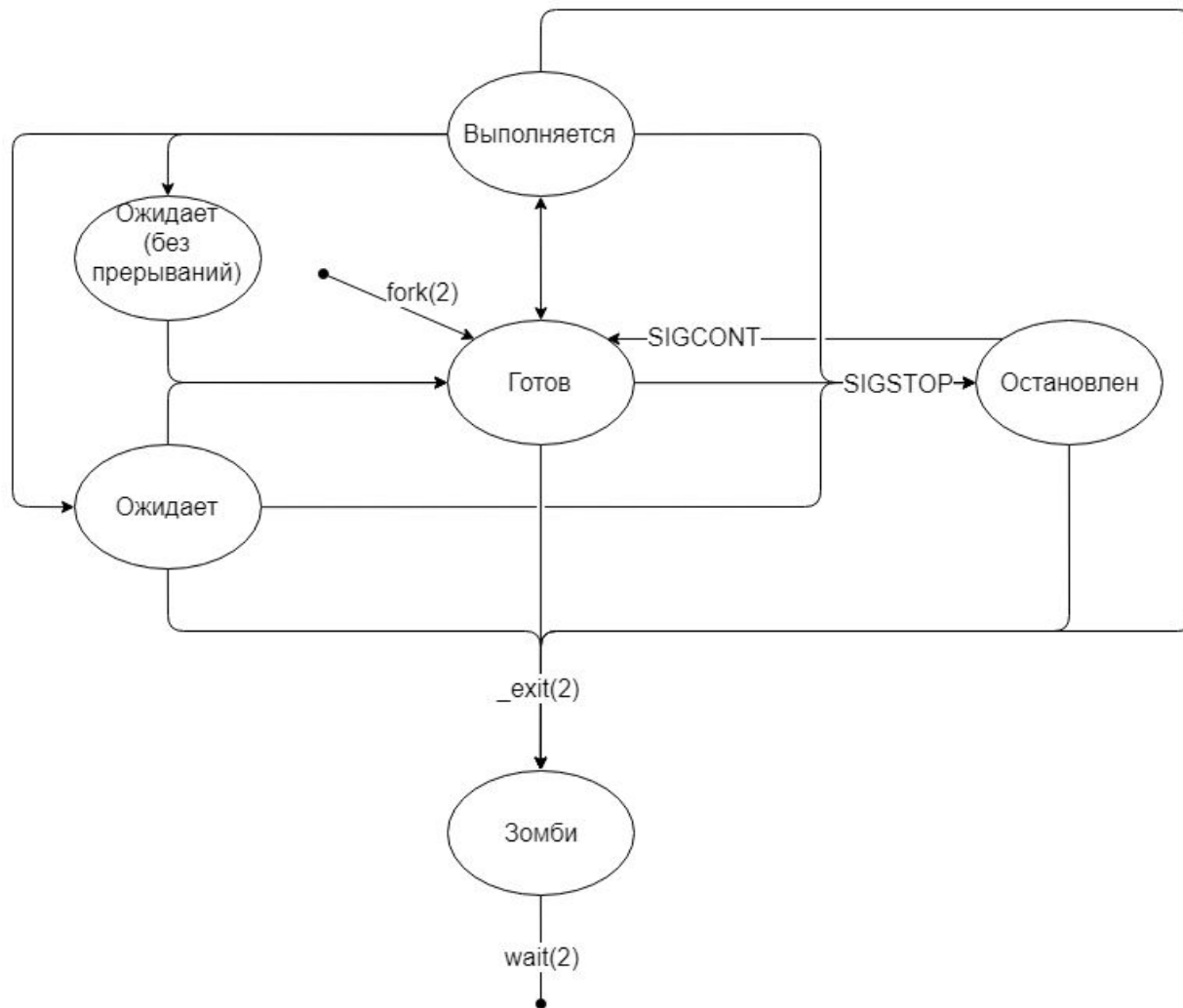


Процесс программы

Процесс программы в ядре представляется просто как структура с множеством полей:

- идентификатор процесса (`pid`);
- открытые файловые дескрипторы (`fd`);
- обработчики сигналов (`signal handler`);
- текущий рабочий каталог (`cwd`);
- переменные окружения (`environ`);
- код возврата.

Жизненный цикл процесса



Оперативная память

Влияет на производительность системы в целом, так как программы при запуске сохраняются в ней.

Кроме этого если не присутствует нехватка оперативной памяти, то это приводит к **постоянному кэшированию памяти на накопитель в раздел SWAP.**

Центральные процессоры

Элемент системы который отвечает за обработку операций.


Характеристики влияющие на производительность:

- архитектура;
- кэш память процессора;
- скорость шины передачи данных;
- множитель частот;
- число потоков;
- число ядер процессора.

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;
- на производительность также может влиять использование технологии RAID.



Утилиты мониторинга производительности

Утилиты мониторинга производительности

- **top** – вывод список работающих в системе процессов и информацию о них;
- **atop** – продвинутый интерактивный полноэкранный монитор производительности;
- **mpstat** – выводит статистику по процессору;
- **iostat** – мониторит использования дисковых разделов;
- **pidstat** – выводит статистику по процессам;
- **vmstat** – выводит статистику процессора, памяти и о процессах;
- **lvm** – инструмент для работы LVM;
- **mdraid** – программа для работы с массивами дисков.



Утилита top

Утилита top

Утилита top **интерактивная**. При запуске работает с окном терминала как **динамический процесс**.

```
pol@Pol-PC: ~  
top - 10:35:45 up 1:04, 2 users, load average: 0,33, 0,59, 0,56  
Tasks: 222 total, 1 running, 221 sleeping, 0 stopped, 0 zombie  
%Cpu(s): 4,5 us, 1,3 sy, 0,0 ni, 91,8 id, 2,4 wa, 0,0 hi, 0,0 si, 0,0 st  
Киб Mem: 3932556 total, 3402700 used, 529856 free, 135568 buffers  
Киб Swap: 3998716 total, 5644 used, 3993072 free. 1375552 cached Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2704	pol	20	0	1021692	136364	37280	S	7,0	3,5	3:41.79	chrome
2284	pol	20	0	1433008	484648	88628	S	4,0	12,3	7:06.78	firefox
2246	pol	20	0	5256132	209576	96108	S	2,7	5,3	1:55.16	viber
2675	pol	20	0	966484	172524	75132	S	2,7	4,4	1:28.57	chrome
2536	pol	20	0	1219468	220180	96296	S	2,3	5,6	2:18.88	chrome
1575	pol	9	-11	439184	10992	8308	S	2,0	0,3	1:10.34	pulseaudio
1260	root	20	0	785432	82964	66176	S	1,7	2,1	1:21.23	Xorg
1752	pol	20	0	1265176	87548	53188	S	1,7	2,2	0:57.34	compiz
1422	pol	20	0	363940	7992	5892	S	0,7	0,2	0:03.60	ibus-daemon
2405	pol	20	0	522924	83936	49932	S	0,7	2,1	0:23.00	plugin-con+
3816	pol	20	0	622984	30652	23804	S	0,7	0,8	0:02.74	gnome-term+
7	root	20	0	0	0	0	S	0,3	0,0	0:05.78	rcu_sched
18	root	20	0	0	0	0	S	0,3	0,0	0:03.01	rcuos/1

Утилита top

Первая строка (top) дает нам общее представление о загрузке системы.

Здесь указано:

- ** — текущее время;
- ** up — продолжительность работы системы с момента запуска;
- ** user — текущее количество пользователей системы;
- ** load average — средняя загрузка системы одну минуту назад, пять и 15 соответственно.

Утилита top

Вторая строка данных дает информацию об общем количестве процессов в системе, а также об их статусе:

- выполняемые (running);
- в ожидании (sleeping);
- приостановленные (stopped);
- zombie.


Далее идет информация о:

- состоянии процессора (%Cpu);
- ОЗУ (Mem);
- SWAP.

Утилита top

Первые четыре столбика:

- номер процесса (**PID**);
- имя пользователя его запустившего (**USER**);
- какой приоритет имеет процесс в данный момент (**PR**);
- приоритет, присвоенный ему командой NICE (**NI**).



```
top - 10:35:45 up 1:04
Tasks: 222 total, 1 r
%Cpu(s): 4,5 us, 1,3
КиБ Mem: 3932556 tota
КиБ Swap: 3998716 tota
```

PID	USER	PR	NI
2704	pol	20	0
2284	pol	20	0
2246	pol	20	0
2675	pol	20	0
2536	pol	20	0
1575	pol	9	-11
1260	root	20	0
1752	pol	20	0
1422	pol	20	0
2405	pol	20	0
3816	pol	20	0
7	root	20	0
18	root	20	0

Утилита top

Остальные столбики расшифровываются следующим образом:

- ****VIRT** — виртуальная память, которую использует процесс;
- ****RES** — физическая память, занятая данным процессом;
- ****SHR** — общий объем памяти, которую данный процесс делит с другими;
- ****S** — текущий статус процесса:
 - R — running;
 - S — sleeping,
 - Z — zombie;
- ...

Утилита top

Остальные столбики расшифровываются следующим образом:

- ...
- ****%CPU** — процент используемого времени центрального процессора;
- ****%MEM** — процент ОЗУ, используемой процессом;
- ****TIME+** — продолжительность работы процесса с момента запуска;
- ****COMMAND** — название команды (программы), которая инициировала процесс.

Утилита top

Режимы работы утилиты, горячие клавиши:

- сортировка по памяти (%MEM) **Shift+M**;
- сортировка процессов который дольше всего работает **Shift+T**;
- сортировка процессы по их номеру (PID) **Shift+N**;
- сортировка по уровню потребления ресурсов процессора **Shift+P**;
- сортировка процессов по потреблению SWAP **Shift+F**.

Утилита top


Режимы работы утилиты, команды:

- **u** — отфильтровать по имени пользователя;
- **o** — сортировка по столбцам **pri**, **size**, **res**, **cpu**, **time**, **thr**;
- **S** — показать системные процессы (по умолчанию они скрыты);
- **m** — переключение между режимами отображения нагрузки (процессор, память) и загрузки системы ввода-вывода;
- **a** — показать абсолютные пути запущенных процессов.

Утилита top

Режимы работы утилиты, команды:

- **i** — скрыть отображение простаивающих процессов;
- **P** — показывать статистику загрузки по каждому процессору отдельно (для SMP систем);
- **s** — установить время обновления вывода информации (в секундах);
- **k** — уничтожить процесс (запрашивает у PID процесса);
- **H** — включает отображение потоков.



Параметры, отвечающие за производительность

%CPU

Чем выше данный показатель, тем больше он «нагружает» процессор. Данный параметр влияет на всю систему в целом.

Решение:

- читаем логи процесса;
- увеличиваем приоритет;
- убиваем его и перезапускаем.

%MEM

Данный показатель показывает сколько оперативной памяти в данный момент времени «забирает процесс».

Как правило, с уменьшением оперативной памяти, информация начинает копироваться на жесткий диск в раздел подкачки и тем самым уменьшает время работы.

Утечка памяти может также быть связана с ошибками в страницах памяти или неправильной работе приложение.

Решение:

- логи,
- kill pid
- restart.

Load Average

Значения:

- **меньше 1** – самое хорошее значение, нагрузки практически нет;
- **1-3** – нагрузка в пределах нормы, сервер достаточно нагружен работой – но эта нагрузка в пределах нормы;
- **3-10** – это уже серьёзная нагрузка, следует обратить внимание на процессы, HDD и прочие параметры;
- **10-20** – серьёзные проблемы, заметно «торможение» сервера при обработке запросов;
- **20 и более** – очень серьёзные проблемы, доступ к серверу серьёзно затруднён.

Для серверов, у которых количество ядер в процессоре более 1-го – эти значения необходимо увеличить на кол-во ядер в системе.



Утилита atop

Утилита atop

Для установки программы atop для систем Debian:

- `apt install atop.`

Для установки программы atop для систем CentOS:

- `yum install atop.`

Для установки программы atop для систем BSD:

- `cd /usr/ports/sysutils/atop;`
- `make && make install && make install clean.`

Утилита atop

Утилита работает в **интерактивном режиме**, как утилита top.

Кроме этого atop позволяет показывать **загрузку по процессору, памяти, дискам и сети**.

ATOP - debian				2021/01/16 11:30:29				-----				10s elapsed				
PRC	sys	0.06s	user	0.00s	#proc	74	#tslpu	0	#zombie	0	#exit	0				
CPU	sys	0%	user	0%	irq	0%	idle	99%	wait	1%	ipc	notavail				
CPL	avg1	0.05	avg5	0.01	avg15	0.00	cs	1003	intr	2772	numcpu	1				
MEM	tot	987.4M	free	505.3M	cache	266.5M	buff	18.0M	slab	54.2M	hptot	0.0M				
SWP	tot	1.0G	free	1.0G				vmcom	1.2G	vmlim	1.5G					
DSK	sda		busy	1%	read	0	write	239	MBw/s	2.6	avio	0.44 ms				
NET	enp0s3		pcki	1	pcko	0	sp	1000 Mbps	si	0 Kbps	so	0 Kbps				
	PID	SYSCPU	USRCPU	VGROW	RGROW	RDDSK	WRDSK	RUID	ST	EXC	THR	S	CPUNR	CPU	CMD	1/1
	1351	0.02s	0.00s	-4K	-112K	OK	OK	root	--	-	1	R	0	0%	atop	
	555	0.01s	0.00s	OK	OK	OK	OK	root	--	-	1	S	0	0%	apache2	
	111	0.01s	0.00s	OK	OK	OK	OK	root	--	-	1	I	0	0%	kworker/u2:2-f	
	153	0.01s	0.00s	OK	OK	OK	OK	root	--	-	1	I	0	0%	kworker/0:1H-k	
	1090	0.01s	0.00s	OK	OK	OK	OK	root	--	-	1	I	0	0%	kworker/0:2-ev	

Утилита atop

Режимы работы утилиты shift + команда:

- **m** — выстроит процессы по занятой памяти;
- **d** — построит процессы по использованию диска;
- **n** — выстроит процессы по использованию сети (доступно только с установленным патчем);
- **v** — покажет более подробную информацию о процессах (пользователя, дату и время начала процесса);
- **u** — выстроит таблицу по самым прожорливым пользователям.

Утилита atop

Утилита по дефолту снимает показания каждые 10 минут, и сбрасывает их в лог [/var/log/atop-YYYYMMDD](#)

Файл конфигураций находится [/etc/default/atop](#)

```
root@debian:/var/log/atop# ls
atop_20210116  daily.log  dummy_after  dummy_before
root@debian:/var/log/atop# _
```

Команда atopsar

Команда atopsar — это встроенный анализатор логов atop, который позволит быстро найти проблему и посмотреть лог в нужное вам время.

Основные ключи:

- **b** — с какого времени надо вывести лог;
- **e** — до какого времени надо вывести лог.

Поиск отдельных процессов:

- **O** — top-3 процессов CPU;
- **G** — top-3 процессов RAM;
- **D** — top-3 процессов диск;
- **N** — top-3 процессов сеть.

Команда atopsar

Общая нагрузка:

- **d** — нагрузка на каждый диск отдельно;
- **m** — нагрузка на оперативную память и swap;
- **c** — нагрузка на процессор, отдельно по каждому ядру;
- **p** — количество процессов и тредов;
- **w** — нагрузка на ipv4;
- **W** — ошибки на ipv4;
- **i** — нагрузка на интерфейс.

Команда atopsar

Примеры команд:

нагрузка на RAM с 09:35:01 до 10:55:01 28 января

```
# atopsar -r /var/log/atop/atop_20120116 -b 09:35:01 -e 10:55:01 -m
```

Нагрузка по сетевым интерфейсам:

```
atopsar -r /var/log/atop/atop_20120116 -i -b 00:40:00 -e 01:30:00
```

Нагрузка на диски:

```
atopsar -r /var/log/atop/atop_20120116 -d -b 01:10:00 -e 01:30:00
```



Пакет sysstat

Пакет **sysstat**

sysstat – это набор инструментов мониторинга производительности для Linux

В состав пакета входят следующие утилиты:

- mpstat;
- pidstat;
- vmstat;
- iostat.

➔ Сегодня рассмотрим **mpstat** и **pidstat**.

mpstat

mpstat – отчет о использовании процессоров.

В состав программы входят следующие ключи:

- **A** – выводит всю возможную информацию о процессорах и всю возможную информацию команды **mpstat**;
- **P** – покажет индивидуальные процессоры или ядра вместе со статистикой.

Дополнительные параметры ALL {0..n}, где «n» – число процессоров.

pidstat

pidstat – используется для мониторинга процессов в режиме реального время.

В состав программы входят следующие ключи:

- **p xxx** – выводит список процессов.
 - **xxx- ALL** – выводит все процессы;
 - **xxx- номер процесса** – выводит конкретный процесс и описание.
- **r** – ключ получить статистику об использовании оперативной памяти и ошибок страниц;
- **C** – отображение статистики по эффективности на основе имени процесса.

pidstat

В состав **pidstat** входят следующие ключи:

- **l** – позволяет просмотреть полный путь до к команде;
Например: `pidstat -C mysql -l`.
- **t** – выводит на экран информацию в виде дерева;
Например: `pidstat -t -C "mysql"`.
- **rud** – посмотреть всю статистику в горизонтальном виде и на одной строке.



Итоги

Итоги

Утилиты для мониторинга производительности позволяют проверять производительность **в реальном времени**, что в свою очередь дает информацию для **принятия дальнейших решений**.

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