

# Диагностика проблем

- [Диагностика проблем производительности \(ОС\) Linux](#)
- [Диагностика сетевых проблем](#)

# Диагностика проблем производительности (ОС) Linux

Диагностика в ОС Linux является важной задачей для обеспечения эффективной работы системы. Такие проблемы могут возникать по разным причинам, включая неправильную настройку, недостаток ресурсов, проблемы с программным обеспечением или нагрузкой на систему.

Почему возникают проблемы производительности:

- Неправильная настройка параметров системы, таких как ядра, сети или памяти.
- Недостаточные ресурсы, такие как процессорное время, память или пропускная способность дисков.
- Наличие вредоносных программ или других проблемных процессов.
- Высокая нагрузка на систему из-за интенсивной работы или слишком большого количества активных процессов.

Утилиты для диагностики проблем производительности:

- `top`: отображает текущие активные процессы, их использование CPU и памяти, а также общую нагрузку на систему.
- `htop`: альтернатива `top` с расширенными функциями и интерактивным интерфейсом.
- `vmstat`: предоставляет информацию о системных ресурсах, включая использование CPU, памяти, дисков и сети.
- `iostat`: отображает статистику ввода/вывода дисковых устройств, помогая выявить проблемы с производительностью дисков.
- `sar`: собирает и анализирует системные данные, включая загрузку процессора, использование памяти, активность сети и другие параметры.
- `strace`: позволяет отслеживать системные вызовы и сигналы, что может помочь выявить проблемные процессы.

Влияние проблем производительности:

- Снижение отзывчивости системы и приложений.
- Увеличение времени отклика пользовательских запросов.
- Задержки и прерывания в работе процессов и сервисов.
- Потеря данных или некорректная обработка информации.

Например, в гипотетической системе Linux наблюдаются проблемы с недостатком ресурсов, и один из процессов расходует все ресурсы ОС.

1. Нужно использовать утилиту `top` для определения процесса, потребляющего большую часть ресурсов.
2. В окне `top` будет список активных процессов, их использование CPU, памяти и другие параметры. Отсортируйте процессы по использованию ресурсов, нажав клавишу `Shift + P`.
3. Обратите внимание на процесс, который потребляет большую часть ресурсов, таких как CPU или память. Возможно, это будет видно в столбцах `%CPU` и `%MEM`.
4. Определите идентификатор процесса (PID) проблемного процесса. Он обычно указывается в левой части окна `top`. Запишите этот PID для дальнейшего использования.
5. После определения проблемного процесса, введите команду `kill <PID>`, чтобы остановить его. Например, если PID проблемного процесса равен 1234, выполните следующую команду:

```
kill 1234
```

6. Дождитесь некоторого времени, чтобы убедиться, что процесс завершился. Вы можете повторно запустить команду `top`, чтобы проверить использование ресурсов системы.
7. Если проблема все еще продолжается или процесс не останавливается, вы можете использовать команду `kill -9 <PID>`, которая принудительно прекратит выполнение процесса. Это следует использовать только в крайних случаях, когда обычная команда `kill` не срабатывает.

Вывод `top` гипотетической ОС:

```
top - 13:15:28 up 10 days,  2:30,  2 users,  load average: 0.78, 1.12, 1.21
tasks: 193 total,   2 running, 191 sleeping,   0 stopped,   0 zombie
Cpu(s): 12.3 us,  4.5 sy,  0.0 ni, 82.7 id,  0.5 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :  7862.4 total,   170.3 free,  5285.9 used,   2406.2 buff/cache
MiB Swap:  2048.0 total,   896.0 free,  1152.0 used.  1445.9 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR S  %CPU  %MEM    TIME+  COMMAND
 1234 username  20   0  23456   6789  4321 R   50.0   0.1   0:10.00 problematic_process
 5678 username  20   0  12345   1234   567 S    5.0   0.0   0:02.00 another_process
```

В этом примере показаны заголовки столбцов, а также несколько строк, представляющих различные процессы. Вы можете видеть идентификатор процесса (PID), пользователя (USER), использование CPU (%CPU), использование памяти (%MEM), а также другую информацию о процессах. Проблемный процесс обычно отличается по высокому использованию CPU или памяти.

# Диагностика сетевых проблем

Диагностика сетевых проблем в ОС Linux включает исследование и анализ различных аспектов сети для определения и устранения проблем.

## Возможные причины проблем:

- Неправильная конфигурация сетевых настроек.
- Проблемы с подключением к сети или устройством сетевого интерфейса.
- Конфликты IP-адресов или другие проблемы сетевого стека.
- Недоступность сетевых ресурсов, таких как серверы или маршрутизаторы.

## Базовые утилиты для диагностики сетевых проблем:

- `ping`: Позволяет проверить доступность узла в сети и оценить время отклика.
- `traceroute` или `tracert`: Определяют маршрут пакетов к заданному узлу и показывают промежуточные хопы.
- `ifconfig` или `ip`: Позволяют просмотреть и настроить сетевые интерфейсы, IP-адреса и другие параметры.
- `netstat` или `ss`: Предоставляют информацию о сетевых соединениях, портах и маршрутах.
- `tcpdump`: Позволяет захватывать и анализировать сетевой трафик для обнаружения проблем.
- `nslookup` или `dig`: Предоставляют информацию о DNS-запросах и разрешении имён.
- `iptables` или `ufw`: Позволяют управлять правилами брандмауэра и фильтровать сетевой трафик.
- `mtr`: `mtr` (My Traceroute) - это комбинированный инструмент, который объединяет функции `ping` и `traceroute`. Он предоставляет непрерывный мониторинг сети и позволяет определить проблемные участки на маршруте к заданному узлу.

## Влияние сетевых проблем:

- Потеря или задержка сетевых пакетов, что может привести к снижению производительности приложений и служб.
- Недоступность сетевых ресурсов, таких как серверы, базы данных или веб-сайты, что может привести к простоем приложений работающих через сеть.
- Неправильное маршрутизирование или нарушение безопасности, что может повлиять на интеграцию и связность сети.

## MTR

`mtr` выводит информацию о промежуточных хопах, аналогично `traceroute`, но отличается тем, что продолжает слать пакеты на каждый хоп в режиме реального времени. Это позволяет получать актуальные данные о времени отклика и потерях пакетов.

Преимущества использования `mtr`:

- Он помогает определить точное место возникновения проблемы на маршруте к узлу.
- Предоставляет информацию о времени отклика и потерях пакетов на каждом хопе, что полезно для анализа и диагностики.
- Непрерывный режим мониторинга помогает отслеживать изменения в сети и выявлять временные проблемы.

Информация о потере пакетов (`Loss%`) позволяет определить наличие проблем на определенных хопах, а столбцы `Avg`, `Best`, `Worst` и `StDev` предоставляют информацию о времени отклика и его статистике.

Пример использования `mtr`:

`mtr google.com`

команда запустит `mtr` для мониторинга маршрута к `google.com` и будет выводить статистику времени отклика и потерь пакетов для каждого хопа в режиме реального времени.

Пример вывода команды:

| OST: имя_хоста     | Loss% | Snt | Last | Avg | Best | Wrst | StDev |
|--------------------|-------|-----|------|-----|------|------|-------|
| 1.   -- прыжок1    | 0.0%  | 5   | 0.5  | 1.2 | 0.5  | 2.3  | 0.8   |
| 2.   -- прыжок2    | 0.0%  | 5   | 1.5  | 2.0 | 1.5  | 2.8  | 0.4   |
| 3.   -- прыжок3    | 0.0%  | 5   | 2.8  | 3.1 | 2.8  | 3.7  | 0.3   |
| 4.   -- прыжок4    | 0.0%  | 5   | 5.1  | 5.2 | 4.9  | 5.6  | 0.2   |
| 5.   -- прыжок5    | 0.0%  | 5   | 4.3  | 4.5 | 4.3  | 4.9  | 0.2   |
| 6.   -- google.com | 0.0%  | 5   | 6.1  | 5.8 | 5.4  | 6.6  | 0.4   |

Пример показывает что проблем сетевого характера нет. Значение `Loss%` равно 0.0%, что означает отсутствие потери пакетов на всех хопах. Столбец `Avg` показывает среднее время отклика (в миллисекундах) на каждом хопе. Время отклика на каждом хопе является стабильным и достаточно низкими.

Пример, когда на маршруте наблюдаются проблемы:

| OST: имя_хоста  | Loss% | Snt | Last | Avg | Best | Wrst | StDev |
|-----------------|-------|-----|------|-----|------|------|-------|
| 1.   -- прыжок1 | 0.0%  | 5   | 0.5  | 1.2 | 0.5  | 2.3  | 0.8   |
| 2.   -- прыжок2 | 5.0%  | 5   | 1.5  | 2.0 | 1.5  | 2.8  | 0.4   |

|                     |         |   |      |      |      |      |      |
|---------------------|---------|---|------|------|------|------|------|
| 3.   -- прыжок3     | 10. 0%  | 5 | 2. 8 | 3. 1 | 2. 8 | 3. 7 | 0. 3 |
| 4.   -- прыжок4     | 50. 0%  | 5 | 5. 1 | 5. 2 | 4. 9 | 5. 6 | 0. 2 |
| 5.   -- прыжок5     | 80. 0%  | 5 | 4. 3 | 4. 5 | 4. 3 | 4. 9 | 0. 2 |
| 6.   -- google. com | 100. 0% | 5 | 6. 1 | 5. 8 | 5. 4 | 6. 6 | 0. 4 |

В этом примере наблюдается потеря пакетов на нескольких хопах. Значение **Loss%** показывает процент потери пакетов на каждом хопе. Как видно, потери пакетов возникают на хопах 2, 3 и 4, а также на хопе 5 уже наблюдается потеря 80% пакетов. На последнем хопе (целевом сервере) потеря пакетов достигает 100%. Это указывает на проблемы в сети на пути к целевому серверу.

Пример, когда проблемы возникают у пользователя на его устройстве:

|                        |         |     |      |      |      |      |       |
|------------------------|---------|-----|------|------|------|------|-------|
| OST: имя_хоста         | Loss%   | Snt | Last | Avg  | Best | Wrst | StDev |
| 1.   -- прыжок1        | 0. 0%   | 5   | 0. 5 | 1. 2 | 0. 5 | 2. 3 | 0. 8  |
| 2.   -- прыжок2        | 0. 0%   | 5   | 5. 1 | 5. 2 | 4. 9 | 5. 6 | 0. 2  |
| 3.   -- имя_устройства | 100. 0% | 5   | 0. 0 | 0. 0 | 0. 0 | 0. 0 | 0. 0  |
| 4.   -- google. com    | 100. 0% | 5   | 6. 1 | 5. 8 | 5. 4 | 6. 6 | 0. 4  |

В данном примере видно, что на хопе 5 (**имя\_устройства**) наблюдается потеря всех пакетов (100% потерь). Это указывает на проблему, возникающую на устройстве пользователя, причем в первую очередь на его компьютере или домашнем роутере. Проблема может быть связана с конфигурацией сетевых настроек, проблемами с подключением или неисправностью устройства.

Пример, когда у целевого сервера есть проблемы:

|                     |        |     |      |      |      |      |       |
|---------------------|--------|-----|------|------|------|------|-------|
| OST: имя_хоста      | Loss%  | Snt | Last | Avg  | Best | Wrst | StDev |
| 1.   -- прыжок1     | 0. 0%  | 5   | 0. 5 | 1. 2 | 0. 5 | 2. 3 | 0. 8  |
| 2.   -- прыжок2     | 0. 0%  | 5   | 1. 5 | 2. 0 | 1. 5 | 2. 8 | 0. 4  |
| 3.   -- прыжок3     | 0. 0%  | 5   | 2. 8 | 3. 1 | 2. 8 | 3. 7 | 0. 3  |
| 4.   -- прыжок4     | 0. 0%  | 5   | 5. 1 | 5. 2 | 4. 9 | 5. 6 | 0. 2  |
| 5.   -- прыжок5     | 0. 0%  | 5   | 4. 3 | 4. 5 | 4. 3 | 4. 9 | 0. 2  |
| 6.   -- google. com | 50. 0% | 5   | 6. 1 | 5. 8 | 5. 4 | 6. 6 | 0. 4  |

В данном примере на хопе 6 (**google. com**) наблюдается 50% потери пакетов. Это может указывать на проблемы, возникающие непосредственно на целевом сервере. Возможные причины могут включать перегрузку сервера, неполадки в сети у хостинг-провайдера или проблемы с самим сервером.