

# Dstat: Универсальный инструмент мониторинга производительности Linux

Dstat — это мощная утилита с открытым исходным кодом для мониторинга производительности систем под управлением Linux в реальном времени. Он агрегирует и отображает исчерпывающую информацию, объединяя функционал множества классических инструментов в единое, удобное для восприятия представление.



# Комплексные возможности Dstat

1

## Сбор данных

Dstat собирает данные о производительности из различных источников, включая ядро Linux через виртуальные файловые системы, системные счетчики и специализированные интерфейсы.



## Анализ в реальном времени

Утилита предоставляет комплексный анализ в реальном времени, одновременно отслеживая метрики из разных областей: CPU, диск, память, сеть, система. Это позволяет оперативно выявлять корреляции.

3

## Модульная архитектура

Благодаря расширяемой архитектуре, Dstat поддерживает использование встроенных и пользовательских плагинов для мониторинга специфичных сервисов и показателей.

4

## Исторический анализ

Dstat может выводить данные в формате, пригодном для последующей обработки и построения графиков, что критически важно для ретроспективного анализа и отчетности.

# Ключевые преимущества Dstat

```
[root@freepbx ~]# dstat -s
--swap--
used free
2320k 818M
2320k 818M
2320k 818M
2320k 818M
2320k 818M
2320k 818M
3864k 816M
```

## → Низкие накладные расходы

Эффективная реализация Dstat обеспечивает минимальное влияние на производительность мониторируемой системы, что критически важно для производственных серверов.

## → Гибкость и расширяемость

Плагинная архитектура позволяет адаптировать утилиту под конкретные задачи и добавлять новые метрики без изменения основного кода.

## → Исторические данные

Поддержка экспорта и работы с сохраненными результатами облегчает долгосрочный анализ тенденций и планирование ресурсов.

# Как Dstat собирает информацию?

Dstat использует простую и эффективную архитектуру, основанную на централизованном сборе метрик через прямой опрос системных источников. Его работа строится на нескольких уровнях.

## Ядро утилиты

Центральный координатор,  
управляющий циклом  
мониторинга и  
координирующий работу  
плагинов.



## Плагины

Специализированные сборщики  
данных, отвечающие за  
конкретные типы метрик.

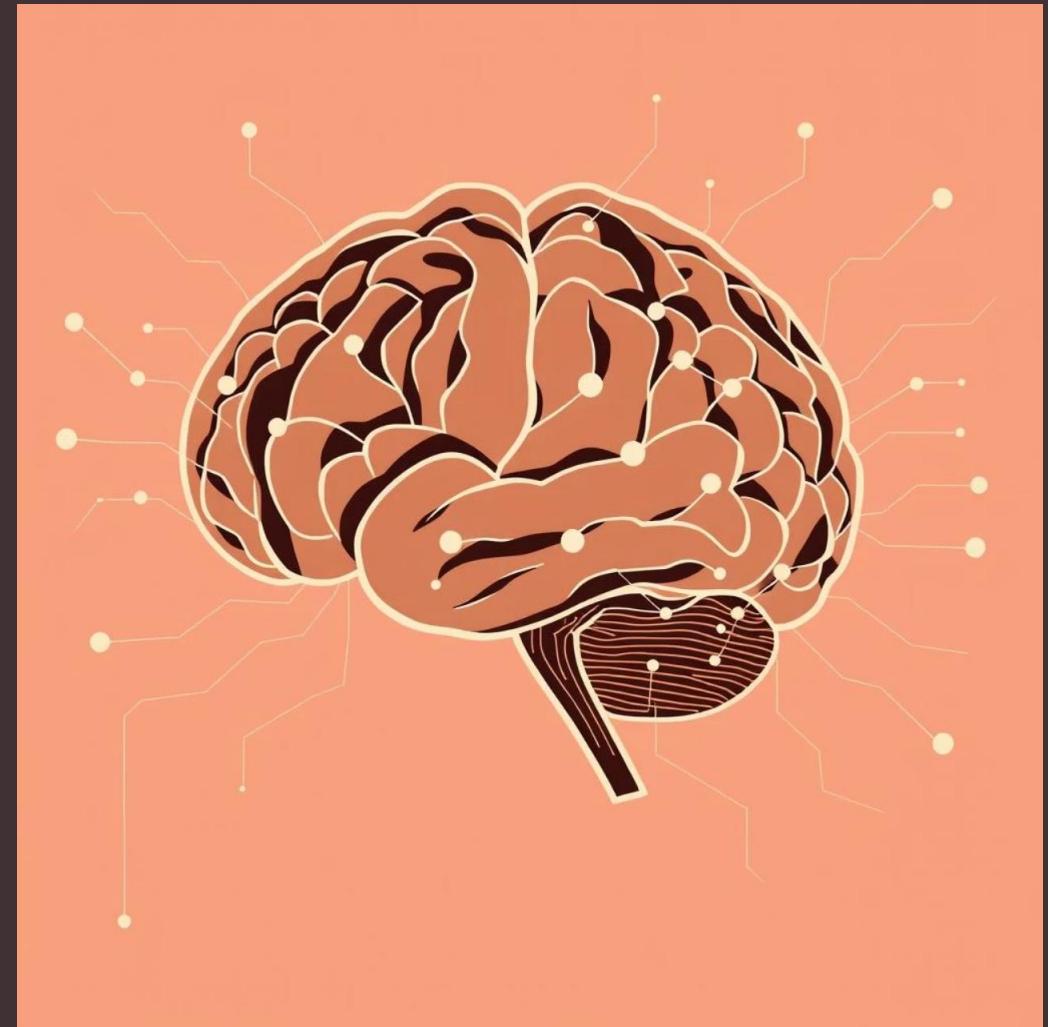
## Источники данных

Виртуальные файловые системы и  
системные вызовы ядра Linux.

# Ядро Dstat: Центральный координатор

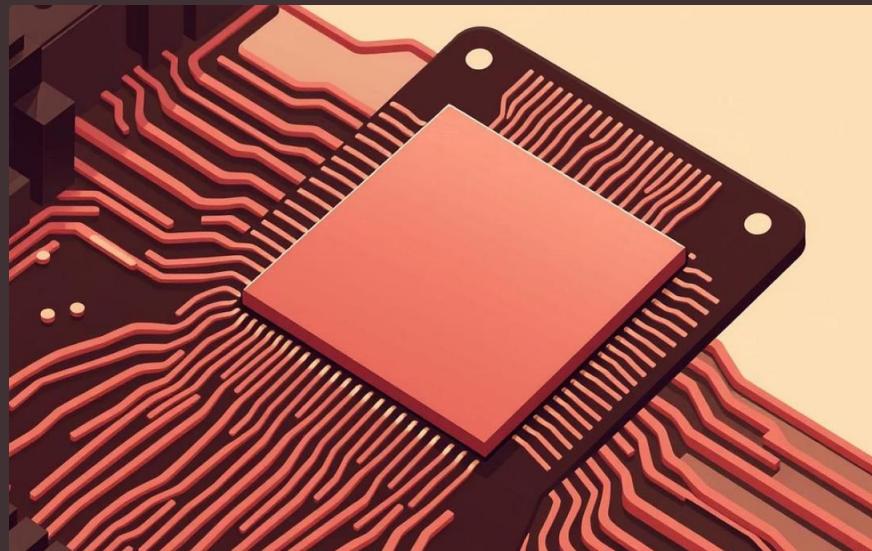
Ядро утилиты (Core Engine) — это центральный элемент Dstat, который управляет всем процессом мониторинга. Оно определяет интервал обновления данных и количество итераций.

- Координирует работу всех активных плагинов, запрашивая у них данные в каждом цикле.
- Форматирует полученную информацию в единую таблицу для удобного вывода на экран или в файл.
- Обеспечивает синхронизацию и целостность данных от различных источников.



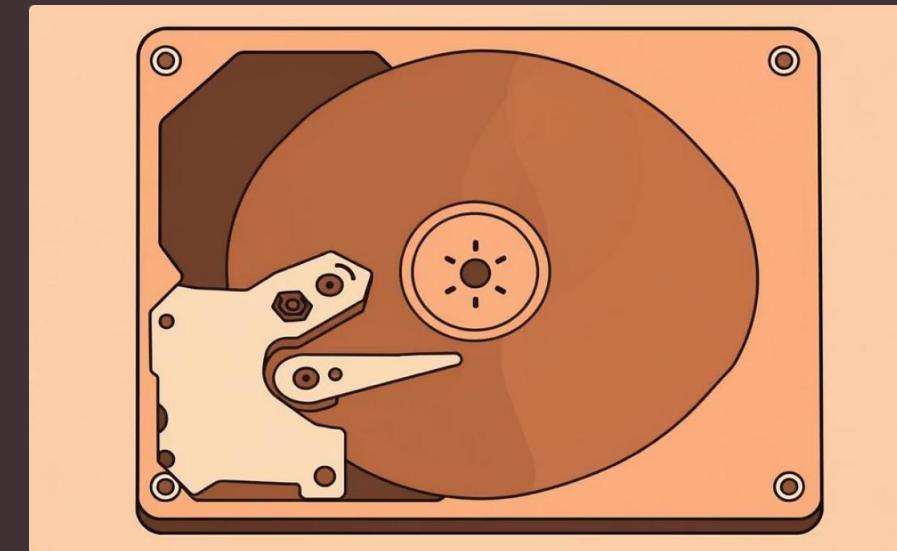
# Плагины Dstat: Специализированные сборщики

Каждый плагин в Dstat отвечает за сбор данных для определенного типа метрик и знает, где их получить. Это обеспечивает модульность и гибкость инструмента.



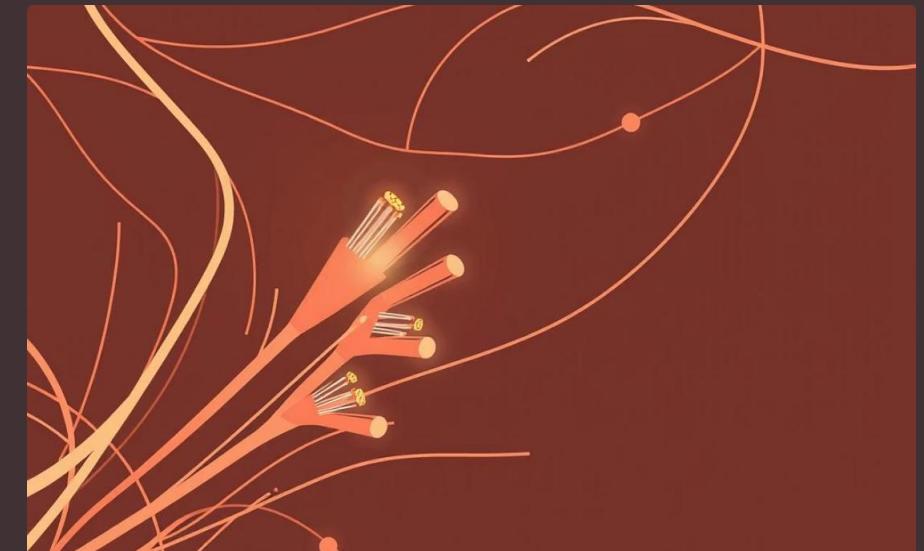
## CPU-плагин

Собирает данные об использовании процессорного времени, включая пользовательское, системное, время ожидания ввода-вывода и простоя.



## Disk-плагин

Предоставляет статистику операций ввода-вывода дисковых подсистем: чтение, запись, количество операций в секунду.



## Net-плагин

Отвечает за сетевую статистику, отображая объем отправленных и полученных данных по каждому сетевому интерфейсу.

# Дополнительные плагины



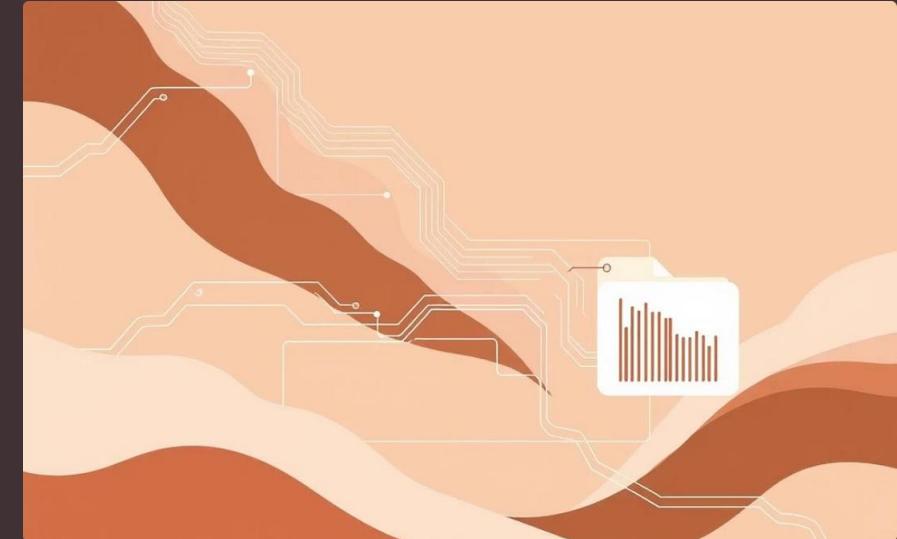
## Memory-плагин

Собирает информацию об использовании оперативной памяти: свободная, использованная, буферы, кэш.



## Proc-плагин

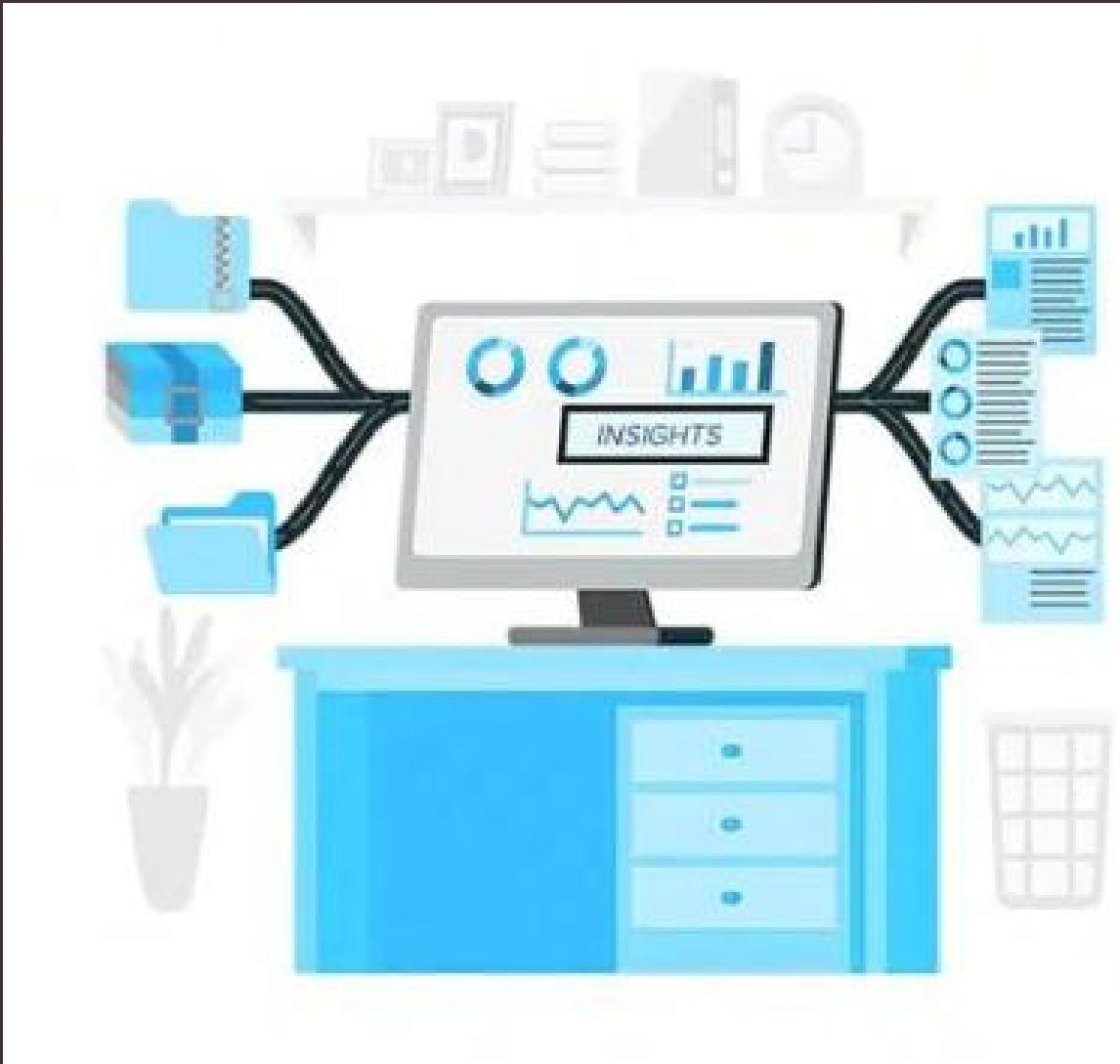
Предоставляет данные о процессах и системной нагрузке: количество новых процессов, загрузка системы.



## Virt-плагин

Отображает метрики для виртуализированных сред, такие как использование процессора и памяти гостевыми системами (для KVM, Xen).

# Источники данных Dstat



Плагины Dstat получают информацию из ключевых компонентов операционной системы Linux.

1

## Виртуальная файловая система /proc

Основной источник данных, содержащий актуальные счетчики ядра. Файлы типа `/proc/stat`, `/proc/meminfo`, `/proc/diskstats` и `/proc/net/dev` предоставляют жизненно важную информацию о состоянии системы.

2

## Файловая система /sys

Используется для получения структурированной информации об устройствах и их конфигурации, что позволяет плагинам получать более детальные аппаратные метрики.

3

## Системные вызовы и библиотеки

Применяются для получения более детальных данных, например, о процессах, их состояниях и ресурсах, используя стандартные интерфейсы ядра Linux.

# Алгоритм работы Dstat

Dstat следует четкому итеративному алгоритму для сбора и представления данных мониторинга.



## Инициализация

Пользователь запускает Dstat, указывая необходимые плагины и параметры. Ядро утилиты инициализирует выбранные плагины.



## Цикл мониторинга

Начинается непрерывный цикл. В каждом интервале ядро опрашивает каждый активный плагин на предмет текущих данных.



## Сбор данных плагинами

Каждый плагин обращается к своему источнику данных (`/proc`, `/sys`, системные вызовы) и считывает текущие значения счетчиков.



## Вычисление метрик

Для накопительных метрик плагины вычисляют разницу между текущим и предыдущим значением, определяя активность за прошедший интервал.



## Форматирование и вывод

Ядро получает обработанные данные от всех плагинов, форматирует их в четкую, согласованную таблицу и выводит на экран или в файл.

# Dstat: Ваш пульт управления системой

Таким образом, Dstat выступает в роли универсального и настраиваемого консольного пульта, предоставляя системному администратору целостное и динамичное представление о состоянии всей системы в режиме реального времени.



Примите Dstat на вооружение для эффективного мониторинга и диагностики вашей Linux-инфраструктуры!