# Мониторинг состояния ФС

### 1. Введение

В операционных системах Linux файловая система играет ключевую роль в стабильности, производительности и долговечности системы. Её износ, переполнение, фрагментация или ошибки могут не только снизить скорость работы, но и привести к потере данных.

**№** Если вы не мониторите состояние ФС — вы не знаете, когда она выйдет из строя.

### 🔾 Почему важно мониторить файловую систему

- **Ш Износ SSD** важно контролировать количество циклов перезаписи и своевременно применять TRIM.
- $\oslash$  **Переполнение разделов** приводит к сбоям сервисов, невозможности логгирования, падению баз данных.
- **Фрагментация** снижает производительность, особенно на HDD и при больших файлах на ext4.

### **П** Что можно и нужно контролировать

- Свободное место и inode (через df, du);
- Нагрузку на диск (через iostat, iotop, dstat);
- Износ и состояние SSD (через smartctl, Isblk --discard);
- Фрагментацию и дефрагментацию (через e4defrag, filefrag);
- Paбoty TRIM (через fstrim и systemd-timer);
- **Файлы, которые нельзя увидеть напрямую** (удалённые, но занятые через lsof);

### **Я Когда мониторинг особенно важен**

• 📳 На серверах — где переполнение раздела может остановить критически

важные процессы;

- 🗂 **В контейнерах** ограниченные ресурсы требуют жёсткого контроля использования места;
- ¼ **Ha SSD и NVMe** где важен TRIM, износ и равномерное распределение нагрузки;
- **В виртуальных машинах** особенно c thin provisioning, где видимое пространство не отражает реальность;
- На десктопах для избежания фрагментации и потери данных.

<sup>☼</sup> Вывод: грамотный мониторинг ФС — это не только диагностика, но и профилактика. Чем раньше вы замечаете проблему, тем дешевле и проще её устранить.

### 2. Использование df и du: сколько занято и где

Одни из самых базовых и полезных инструментов в Linux — это df и du. Они позволяют быстро понять, сколько места занято и какими каталогами. Однако они показывают информацию по-разному и предназначены для разных задач.

### 🖬 df — информация о файловых системах

Показывает использование места на уровне точек монтирования и устройств:

df -h

Показывает размеры в человекочитаемом виде (GB, MB)

df-i

Отображает использование **inode** — критично, если файлов очень много, даже при наличии свободного места.

🕅 du — анализ содержимого каталогов

Подсчитывает **фактический размер каталогов и подкаталогов** на уровне файловой структуры:

du -sh \*

Показывает, какие директории в текущем каталоге занимают больше всего места.

sudo du -xh / --max-depth=2 | sort -h

Рекурсивный просмотр с сортировкой по размеру.

🌣 Совет: используйте ncdu (если установлен) для интерактивного анализа:

sudo ncdu /

### 🍳 Разница между df и du

- df читает информацию из /proc/mounts и показывает размеры томов;
- du обходит реальные файлы и папки, рассчитывая их размеры по факту;
- df включает скрытые и удалённые, но ещё открытые файлы (например, журналы, занятые процессами);
- du не увидит эти файлы, т.к. они уже не видны в файловом дереве.

<u>↑</u> **Если df показывает занятое место, а du — нет**, скорее всего, виноваты удалённые, но не освобождённые файлы. Найти их можно командой:

Isof +L1

### 3. Мониторинг ввода-вывода: iostat, iotop, dstat

Мониторинг I/O (ввода-вывода) — важная часть наблюдения за состоянием файловой системы. Он помогает выявить медленные диски, «тяжёлые» процессы и узкие места в производительности.

### 📰 iostat — статистика по устройствам

Утилита из пакета sysstat показывает производительность дисковых устройств, включая загрузку, задержки и throughput:

iostat -xz 1

• %util — насколько устройство занято (если 100% — диск перегружен);

- await средняя задержка всех операций (чем выше, тем хуже);
- r/s, w/s количество чтений и записей в секунду;
- svctm среднее время выполнения операций устройством.

<u>↑</u> **Если %util 100% при больших await** — диск является «бутылочным горлышком».

**●**□ iotop — кто грузит диск прямо сейчас

Интерактивная утилита, показывающая процессы, активно работающие с диском:

sudo iotop

- Отображает **PID, процесс, скорость чтения/записи** в реальном времени;
- Удобно для обнаружения утечек логов, неэффективных баз данных и бэкапов;
- Необходимо иметь права root.

### **a** dstat — много метрик в одной строке

dstat — универсальный инструмент для мониторинга I/O, CPU, памяти, сети в реальном времени:

dstat -cdngytl

Каждая колонка — отдельный ресурс: диск, сеть, память, процессы, задержки.

**☼ Альтернатива:** если dstat не установлен, попробуйте atop или glances для более детального real-time обзора.

### Интерпретация показателей

- **Высокий %util** + **высокий await** □ диск перегружен (HDD, плохой SSD, или много IOPS);
- **Низкий %util, но медленный доступ** □ проблемы на уровне ФС, контроллера или блокировок;
- Большие объёмы записи возможно, неочищаемые логи или тяжёлые базы.

**Комбинируйте:** iostat для диагностики устройства, iotop — для поиска виновников, dstat — для общей картины.

## 4. Мониторинг файловой активности: inotify, auditd, Isof

Помимо мониторинга места и I/O, в Linux важно следить за **активностью файлов**: кто и когда их создаёт, удаляет, изменяет или использует. Это полезно для отладки, безопасности и устранения проблем с «невидимыми» файлами.

### **Q** lsof +L1 — поиск удалённых, но занятых файлов

Иногда файлы уже удалены, но всё ещё занимают место, потому что открыты какимто процессом. Isof позволяет их найти:

sudo Isof +L1

- Показывает открытые файлы с нулевыми линками (уже удалённые);
- Часто виноваты журналы, временные файлы, бэкапы и базы данных;
- Можно освободить место, перезапустив процесс или закрыв дескриптор.

<u>М</u>□ **df может показывать, что диск забит, но du** — **нет.** Это классический признак удалённого, но занятого файла.

### **▲** Inotifywait — отслеживание изменений в каталогах

Инструмент из пакета inotify-tools позволяет «подслушивать» файловые события в реальном времени:

inotifywait -m -r /var/log

- -m режим постоянного наблюдения;
- -r рекурсивно по подкаталогам.

Вы увидите события вроде CREATE, DELETE, MODIFY, OPEN, CLOSE.

Пример: слежение за появлением новых файлов в /tmp:

inotifywait -m /tmp -e create

### ♡□ auditd — мощный аудит действий на ФС

Демон auditd позволяет логировать доступ к файлам, изменения и нарушения политик безопасности. Используется в защищённых системах (в т.ч. с SELinux).

sudo auditctl -w /etc/passwd -p war -k passwd-watch

- -w путь к файлу;
- -p типы доступа: write, attribute change, read, xexecute;
- -k метка (ключ) для фильтрации.

Просмотр логов:

sudo ausearch -k passwd-watch

**¼ auditd** подходит не только для безопасности, но и для диагностики: кто, когда и как менял конфигурационные файлы или запускал удаление.

### 5. TRIM и износ SSD: fstrim, smartctl, Isblk

В отличие от жёстких дисков, **SSD имеют ограниченное число циклов записи** на ячейку памяти. Для поддержания скорости и срока службы важно регулярно использовать команду **TRIM**, которая сообщает контроллеру, какие блоки больше не используются.

### **≍**□ Что такое TRIM и зачем он нужен

- Позволяет контроллеру SSD «освобождать» блоки после удаления файлов;
- Увеличивает производительность и срок службы накопителя;
- Особенно важен для систем с высокой скоростью записи (журналы, БД, временные файлы).

#### 

Можно вручную инициировать TRIM на смонтированном разделе:

sudo fstrim -v /

Вывод покажет, сколько байт было освобождено.

### Ф Проверка таймера TRIM (если используется systemd)

Большинство дистрибутивов используют systemd-таймер для периодического запуска TRIM:

systemctl status fstrim.timer

Чтобы вручную запустить его немедленно:

sudo systemctl start fstrim.service

### ■ Проверка износа SSD через smartctl

Инструмент smartctl из пакета smartmontools позволяет узнать состояние SSD, включая wear-leveling и число перезаписей:

sudo smartctl -a /dev/sdX | grep -i wear

Другие важные метрики, которые стоит искать:

- Wear\_Leveling\_Count
- Media\_Wearout\_Indicator
- Percent\_Lifetime\_Remain особенно на Intel, Samsung, Kingston

⚠□ **Если износ превышает 90%** или Wear\_Leveling\_Count падает к нулю — стоит задуматься о замене SSD.

### 🗂 Проверка поддержки TRIM устройством

Не все устройства и файловые системы поддерживают TRIM. Проверить можно так:

Isblk --discard

Интересуют столбцы:

- DISC-GRAN минимальный размер TRIM-операции;
- DISC-MAX максимальный размер;
- Если значения равны 0 TRIM **не поддерживается**.

### Поддержка TRIM зависит от:

- Типа накопителя (HDD 🛮 SSD);
- Файловой системы (ext4, XFS, Btrfs поддерживают);
- Флагов монтирования (discard или периодический fstrim).

### 6. Фрагментация: e4defrag, filefrag, xfs\_db

Несмотря на распространённое мнение, **фрагментация файлов** существует и в Linux, особенно на **HDD** и при частом создании/удалении больших файлов. Это может привести к снижению производительности чтения и записи, особенно при доступе к большим логам, базам данных и образам VM.

### 🔑 Проверка фрагментации с помощью filefrag

Утилита filefrag показывает, из скольких физических фрагментов состоит файл:

filefrag -v /path/to/file

- Ищите строку extent чем меньше, тем лучше (идеально 1);
- Подходит для ext4, XFS, Btrfs и других ФС с поддержкой extent;
- Работает с правами пользователя, но лучше запускать от root.

### Дефрагментация ext4 c e4defrag

Стандартный инструмент дефрагментации в ext4:

sudo e4defrag /

Можно дефрагментировать не весь раздел, а конкретную директорию или файл.

Пример:

sudo e4defrag /var/log

### ↑ Для XFS: xfs\_db и xfs\_fsr

- xfs db интерактивный отладочный инструмент (не для новичков);
- xfs\_fsr автоматизированная утилита дефрагментации XFS:

sudo xfs fsr -v /dev/sdX1

Работает в фоновом режиме, дефрагментируя файлы на лету.

и HDD это может привести к падению производительности.

### **∜ Почему фрагментация** — это не миф

- На HDD считывание данных с фрагментами □ частые перемещения головки □ медленная работа;
- На ext4 без lazy\_itable\_init и при постоянном перезаписывании больших файлов (бэкапы, базы) фрагментация растёт быстро;
- SSD менее чувствительны, но фрагментация влияет на wear-leveling и TRIM.

№ Рекомендация: Проверяйте и дефрагментируйте хотя бы периодически, особенно на лог-серверах, базах данных, видеонаблюдении и архивных HDD.

### 7. КОНТРОЛЬ КВОТ: quota, repquota, xfs\_quota, btrfs qgroup

Контроль дисковых квот позволяет ограничивать использование файловой системы по пользователям, группам и подтомам. Это особенно важно в мультипользовательских и серверных системах, где необходимо избежать переполнения ресурсов отдельными пользователями или сервисами.

### **및** Включение квот в ext4 и XFS

Для **ext4** нужно включить квоты в /etc/fstab:

/dev/sdX1 /home ext4 defaults,usrquota,grpquota 0 2

Затем инициализировать файлы квот:

sudo mount -o remount /home sudo quotacheck -cum /home sudo quotaon /home

Для **XFS** поддержка встроенная, но также нужно указать флаги в fstab:

/dev/sdX1 /data xfs defaults,uquota,gquota 0 2

※ XFS поддерживает квоты из коробки и может управлять ими на лету через xfs\_quota.

#### 📊 Утилиты для работы с квотами

- quota -u username ПОКАЗАТЬ КВОТУ ПОЛЬЗОВАТЕЛЯ;
- repquota /mountpoint ОТЧЁТ ПО ВСЕМ ПОЛЬЗОВАТЕЛЯМ;
- edquota username установить лимиты вручную;
- xfs\_quota -x -c 'report -h' /mountpoint для XFS.

### 🗂 Квоты в Btrfs: подтомы и qgroup

Btrfs использует механизм qgroup (quota groups) для ограничения подтомов.

Включение квот:

sudo btrfs quota enable /mnt

Посмотреть текущие квоты:

sudo btrfs qgroup show -pcre /mnt

Установка лимита на подтом:

sudo btrfs qgroup limit 5G /mnt/@home

**Btrfs qgroup** работает на уровне подтомов и даёт гибкий контроль, включая учёт снапшотов.

#### 🔊 Выводы

- ext4: простые user/group квоты, но не поддерживает project-квоты;
- XFS: поддерживает user/group/project-квоты, включая гибкое управление;
- Btrfs: гибкая квотируемая модель с поддержкой подтомов и снапшотов;
- Для **ReiserFS**, **exFAT** и других квоты, как правило, отсутствуют.

# 8. Наблюдение за файловыми системами в реальном времени

Регулярный мониторинг — хорошо, но бывают ситуации, когда нужно **отслеживать состояние дисков прямо сейчас**. Это особенно полезно при устранении проблем, подозрении на утечку места или отладке активности приложений.

### 🛱 watch df -h — базовый real-time монитор

Отображает занятое место с обновлением каждые 2 секунды по умолчанию:

watch df -h

Можно задать частоту обновления:

watch -n 5 df -i

Хорошо подходит для отслеживания резкого роста логов, временных файлов и inode.

### 🖬 ncdu, gdu — интерактивный анализ места

Эти утилиты позволяют в интерактивном режиме просматривать, какие каталоги занимают больше всего места.

- ncdu / наглядная древовидная структура, управление стрелками;
- gdu / более быстрая альтернатива на Go (особенно для больших томов);
- Работают даже на сервере по SSH.

sudo ncdu /var/log

Полезно: сразу видно каталоги, занимающие десятки или сотни ГБ, даже если они
"глубоко спрятаны".

### glances и htop с колонками I/O

glances — комплексная панель мониторинга ресурсов:

- Показывает загрузку CPU, RAM, дисков, сети, температуру и т.д.;
- Поддерживает web-интерфейс и API;
- Запуск: glances (требует установки пакета).

**htop** можно настроить для отображения активности I/O по процессам:

- Нажмите F2 🛘 **Columns** 🖺 добавьте IO read rate, IO write rate;
- Удобно отслеживать, кто нагружает диск в данный момент.

**Совет:** Используйте ncdu для анализа, watch df — для наблюдения, htop/glances — для общего контроля состояния.

## 9. Интеграция с мониторингом и алертами

Важно не только наблюдать за файловой системой вручную, но и **автоматизировать мониторинг** с оповещениями. Это позволяет предупреждать аварии до того, как они приведут к простоям или потере данных.

### node\_exporter + Prometheus

Один из самых популярных инструментов — <u>node exporter</u> от Prometheus. Он собирает множество системных метрик, включая:

- node\_filesystem\_avail\_bytes доступное место;
- node\_filesystem\_size\_bytes общий размер ФС;
- node\_filesystem\_files\_free свободные inode;
- node\_filesystem\_readonly признак Read-Only монтирования.

### Пример выражения Prometheus:

100 \* (node\_filesystem\_avail\_bytes{mountpoint="/"} / node\_filesystem\_size\_bytes{mountpoint="/"}) < 10

Алерт при остатке меньше 10% на /

### 🔏 Примеры алертов

• Диск почти заполнен:

node\_filesystem\_avail\_bytes / node\_filesystem\_size\_bytes < 0.1

• Inode почти закончились:

node filesystem files free / node filesystem files < 0.05

• Файловая система в RO:

node\_filesystem\_readonly == 1

**☼ Совет:** интегрируйте с Alertmanager для отправки уведомлений в Telegram, Slack, email и т.д.

#### **☆**□ Альтернативы и готовые решения

- Netdata визуальный real-time мониторинг с auto-оповещениями;
- Zabbix классическая система с агентами и встроенными шаблонами;
- **Grafana + Telegraf + InfluxDB** мощная система визуализации с лёгким сборщиком данных;
- Checkmk, Icinga, Nagios для корпоративных или гибридных решений.

**Вывод:** если у вас есть более одного сервера или важные сервисы — настройте алерты. Лучше узнать о проблеме заранее, чем от пользователей.

### 10. Заключение

Эффективное использование дискового пространства и предотвращение сбоев напрямую зависит от **регулярного мониторинга файловой системы**. Даже простые инструменты дают ценные данные, если применять их системно.

- **Регулярный мониторинг = стабильность**: вовремя замеченные проблемы позволяют избежать аварий и простоев.
- Комбинируйте инструменты:
  - ∘ df + du анализ места;
  - ∘ iostat + smartctl диагностика I/O и износа;
  - ∘ fstrim + e4defrag оптимизация SSD и ext4.
- 💾 Учитывайте носитель: для SSD критичен TRIM и контроль записи, для HDD фрагментация и seek time.
- **Автоматизируйте:** используйте systemd-таймеры, cron, алерты в Prometheus или Zabbix.

 ✓ Помните: поддерживать файловую систему в хорошем состоянии — это не разовая задача, а постоянный процесс.