Как разбирать логи в Linux: journalctl, grep, awk и sed

Каждый админ хотя бы раз сталкивался с ситуацией, когда сервер внезапно начинает тупить: подвисают процессы, появляются странные задержки, что-то перестаёт работать. И первая остановка в таком случае — это журналы событий. В Linux системные логи хранятся в **systemd-journald**, а его главный инструмент для работы — `journalctl`. Но просто читать логи — это скучно. Разберем, **как фильтровать, искать ошибки, анализировать данные и автоматизировать разбор логов** с помощью `grep`, `awk`, `sed` и других утилит.

Работа с journalctl

Самый простой способ заглянуть в системные журналы — просто вызвать `journalctl`:

```
journalctl
```

Но если сервер работает долго, логов там будет много. Выведем их без постраничного режима:

```
journalctl --no-pager
```

Если команда зависает, скорее всего, журнал слишком большой. Ограничим его по времени, например, **выведем логи за последний час**:

```
journalctl --since "1 hour ago"
```

Теперь можно быстро отследить недавние события.

Как понять, что произошло после перезагрузки?

Если сервер неожиданно перезагрузился, полезно посмотреть, что записалось в логи **после старта**:

```
journalctl -b
```

Если система несколько раз перезагружалась, можно посмотреть события после конкретного запуска, например, **минус первый перезапуск**:

```
journalctl -b -1
```

Это поможет понять, что произошло перед падением системы.

Фильтрация по сервису

Чтобы разобраться в проблемах конкретного сервиса, например, `nginx`, используем флаг `-u`:

```
journalctl -u nginx.service --no-pager
```

Пример вывода:

```
Feb 02 10:15:00 myserver nginx[1234]: Starting nginx: [OK]
Feb 02 10:16:00 myserver nginx[1234]: 502 Bad Gateway
```

Можно дополнительно **сортировать логи от новых к старым**:

```
journalctl -u nginx.service --reverse
```

grep для поиска ошибок

Журналы бывают огромными, но нам нужны только ошибки. Отфильтруем их с помощью `grep`:

```
journalctl -u nginx.service | grep "error"
```

Пример вывода:

```
Feb 02 11:20:00 myserver nginx[5678]: error: failed to bind socket
Feb 02 11:22:00 myserver nginx[5678]: error: no permission to access /var/www/html
```

Чтобы **не зависеть от регистра**, используем `-i`:

```
journalctl -u nginx.service | grep -i "error"
```

Теперь `ERROR`, `Error` и `error` будут найдены.

Иногда важно видеть не только саму ошибку, но и контекст вокруг. Например, **две строки до и после найденного слова**:

```
journalctl -u nginx.service | grep -A 2 -B 2 "error"
```

Так можно увидеть, что происходило **до и после ошибки**.

Выбираем нужные данные с awk

Журналы содержат много лишней информации. Например, оставим **только дату, время и сообщение**:

```
journalctl -u nginx.service | awk '{print $1, $2, $3, $6}'
```

Вывод:

```
Feb 02 12:00:00 Starting
Feb 02 12:05:00 error:
Feb 02 12:10:00 Listening
```

Если сообщения длинные, можно **обрезать текст**:

```
journalctl -u nginx.service | awk '{print substr($0, index($0,$6))}'
```

Это оставит только само сообщение без метаданных.

Очистка логов с sed

Некоторые логи содержат ненужные теги или мусорные символы. Например, уберём всё, что находится в квадратных скобках `[INFO]`:

```
journalctl -u nginx.service | sed 's/\[.*\]//g'
```

Исходный лог:

```
Feb 02 13:30:00 myserver nginx[1234]: [INFO] Server started
```

После обработки:

```
Feb 02 13:30:00 myserver nginx[1234]: Server started
```

Также можно **заменить текст**, например, заменить `error` на `!!!ERROR!!!` для выделения:

```
journalctl -u nginx.service | sed 's/error/!!!ERROR!!!/g'
```

Автоматический сбор ошибок

Допустим, мы нужно **каждый день автоматически собирать ошибки Nginx** и записывать их в файл. Используем **crontab** и скрипт:

```
journalctl -u nginx.service --since "1 day ago" | grep -i "error" | awk '{print $1, $2, $3, $6, $7, $8}' > /var/log/nginx_errors.log
```

Этот код:

1. Берёт логи Nginx за последние 24 часа.
2. Ищет строки с `"error"`.
3. Оставляет только дату, время и сообщение.
4. Записывает результат в `/var/log/nginx_errors.log`.

Вывод логов в реальном времени

А если хочется **смотреть логи в реальном времени**, то спользуйте флаг `-f`:

```
journalctl -u nginx.service -f=
```

Это как `tail -f`, но для journalctl — вы сразу увидите, если сервис упадёт.

Revision #1
Created 2026-03-18 07:54:30 UTC by Эд Кукса
Updated 2026-03-18 07:58:49 UTC by Эд Кукса