
Logs

Índex

1. Per què existeixen?	1
2. On es guarden?	1
3. Qui genera els logs?	1
4. Com es llegeixen?	1
5. Què és journalctl?	2
6. Ús bàsic	2
6.1. Llegir el journal	2
6.2. Filtrar per servei	2
6.3. Filtrar per temps	2
7. Filtres avançats	3
7.1. Per prioritat (nivell)	3
7.2. Per arrencada (boot)	3
7.3. Altres filtres útils	3
8. Format de sortida	4
9. Manteniment i espai en disc	4
10. Combinacions pràctiques	4
11. Conceptes clau	5
12. Comparativa: /var/log/syslog vs. journalctl	5
12.1. Operacions habituals	5
12.2. Filtratge: potència real de la diferència	6
12.3. Estructura dels fitxers	6
12.4. Configuració	6
12.5. Convivència: els dos sistemes junts	7
12.6. Quan usar cada eina	7
Índex de figures	8

Els **logs** (registres) són fitxers de text on el sistema operatiu i les aplicacions **anoten automàticament tot el que passa**: arrencades, errors, connexions, accions d'usuaris, canvis de configuració, etc.

Són una eina fonamental per a qualsevol administrador de sistemes: sense ells, diagnosticar problemes seria com buscar una agulla en un paller sense llum.

1. Per què existeixen?

- **Diagnòstic**: saber per què ha fallat un servei o el sistema
- **Seguretat**: detectar intents d'accés no autoritzats
- **Auditoria**: saber qui ha fet què i quan
- **Monitoratge**: veure l'estat dels serveis en temps real

2. On es guarden?

Tradicionalment a `/var/log/`, en fitxers de text pla.

En sistemes moderns amb **systemd**, els logs guarden en format binari a `/var/log/journal/` i s'accedeix amb `journalctl`.

3. Qui genera els logs?

- El **kernel** (maquinari, drivers)
- Els **serveis del sistema** (nginx, ssh, cron...)
- El **gestor de sessions** (GNOME, login)
- Les **aplicacions** d'usuari

4. Com es llegeixen?

```
# Mètode clàssic (text pla)
tail -f /var/log/syslog
grep "error" /var/log/auth.log

# Mètode modern (systemd)
journalctl -f
journalctl -u ssh -p err
```

5. Què és journalctl?

`journalctl` és la interfície principal per consultar el **journal** de [systemd](#) --- el sistema de registre centralitzat que substitueix (i complementa) `/var/log/syslog`. Tot el que generen serveis, el kernel, el bootloader i les sessions s'emmagatzema en format binari a `/var/log/journal/`.

A diferència dels fitxers `.log` de text pla, el journal emmagatzema metadades (PID, UID, servei, prioritat) que permeten filtrar molt precisament.

6. Ús bàsic

6.1. Llegir el journal

Ordre	Descripció
<code>journalctl</code>	Tots els registres des de l'inici (més antics primer). Usa Space, G, q per navegar (és less).
<code>journalctl -e</code>	Salta directament al final (<i>end</i>), els registres més recents.
<code>journalctl -f</code>	Seguiment en temps real (<i>follow</i>), com <code>tail -f</code> . Ctrl+C per sortir.
<code>journalctl -n 50</code>	Mostra les últimes 50 línies (per defecte 10 amb <code>-n</code> sense valor).
<code>journalctl --no-pager</code>	Ho bolca tot a stdout sense obrir less. Útil per a scripts.

6.2. Filtrar per servei

Ordre	Descripció
<code>journalctl -u nginx</code>	Tot el que ha generat la unitat <code>nginx.service</code> .
<code>journalctl -u nginx -f</code>	Seguiment en temps real del servei <code>nginx</code> .
<code>journalctl -u nginx -u php-fpm</code>	Múltiples serveis combinats (s'afegeix <code>-u</code> per a cada un).
<code>journalctl -u 'ssh*'</code>	Comodins: totes les unitats que comencen per <code>ssh</code> .

Per saber el nom exacte d'una unitat: `systemctl list-units --type=service`

6.3. Filtrar per temps

Ordre	Descripció
<code>journalctl --since today</code>	Registres d'avui a partir de les 00:00.
<code>journalctl --since yesterday</code>	Des d'ahir. Altres mots clau: <code>now</code> , <code>tomorrow</code> .
<code>journalctl --since "2025-01-15 08:00"</code>	Format <code>YYYY-MM-DD HH:MM:SS</code> --- les hores i segons són opcionals.
<code>journalctl --since "1 hour ago"</code>	Expressions relatives. Funciona <code>2 hours ago</code> , <code>30 min ago</code> , etc.
<code>journalctl --since "09:00" --until "10:30"</code>	Interval tancat. <code>--until</code> és l'equivalent final del rang.
<code>journalctl -u nginx --since today -f</code>	Es pot combinar amb qualsevol altre filtre.

7. Filtres avançats

7.1. Per prioritat (nivell)

Ordre	Descripció
<code>journalctl -p err</code>	Mostra registres de nivell <i>error</i> i superiors (err, crit, alert, emerg).
<code>journalctl -p warning..err</code>	Rang de prioritats: de <i>warning</i> a <i>err</i> .

Escala de prioritats (0 = màxima):

Nivell	Número
emerg	0
alert	1
crit	2
err	3
warning	4
notice	5
info	6
debug	7

7.2. Per arrencada (boot)

Ordre	Descripció
<code>journalctl -b</code>	Registres de l'arrencada actual (-b 0 és equivalent).
<code>journalctl -b -1</code>	Arrencada anterior. -2 seria la del penúltim boot, etc.
<code>journalctl --list-boots</code>	Llista totes les arrencades amb data, hora i ID.
<code>journalctl -b -p err</code>	Error del boot actual. Combinació molt útil per diagnosticar.

7.3. Altres filtres útils

Ordre	Descripció
<code>journalctl -k</code>	Missatges del kernel (dmesg equivalent però del journal).
<code>journalctl _PID=1234</code>	Registres d'un PID concret. Podeu usar <code>_UID=</code> , <code>_GID=</code> , <code>_COMM=</code> .
<code>journalctl _SYSTEMD_USER_UNIT=foo</code>	Unitats de l'usuari (no del sistema). Alternativa: <code>journalctl --user -u foo</code> .
<code>journalctl -g "error fail"</code>	Filtre per text amb expressió regular (-g o --grep).
<code>journalctl -u nginx grep 404</code>	La sortida es pot passar per pipe com qualsevol altra ordre.

8. Format de sortida

Ordre	Descripció
<code>journalctl -o short</code>	Per defecte. Una línia per entrada amb timestamp.
<code>journalctl -o json-pretty</code>	JSON formatat amb tots els camps. Útil per processar amb jq.
<code>journalctl -o cat</code>	Només el missatge, sense timestamp ni servei. El més net per a scripts.
<code>journalctl -o verbose</code>	Tots els camps de metadades per entrada. Útil per descobrir camps disponibles.
<code>journalctl -o short-iso</code>	Com short però amb timestamps ISO 8601 --- millor per logs exportats.

9. Manteniment i espai en disc

Ordre	Descripció
<code>journalctl --disk-usage</code>	Quant espai ocupa el journal en disc.
<code>journalctl --vacuum-size=200M</code>	Elimina entrades antigues fins que el journal ocupi menys de 200 MB.
<code>journalctl --vacuum-time=2weeks</code>	Elimina entrades de fa més de 2 setmanes. Unitats: s, m, h, days, weeks, months, years.
<code>journalctl --rotate</code>	Força la rotació dels fitxers actius del journal.

La configuració permanent del journal és a `/etc/systemd/journald.conf` --- paràmetres com `SystemMaxUse=`, `MaxRetentionSec=`, `Storage=`.

10. Combinacions pràctiques

```
# Seguir nginx en temps real des de l'inici del dia
journalctl -u nginx --since today -f

# Errors del boot actual sense paginació --- per copiar o guardar
journalctl -b -p err --no-pager

# Missatges del kernel de l'última mitja hora
journalctl -k --since "30 min ago"

# Avisos i errors de SSH d'avui --- útil per detectar intents d'accés
journalctl -u sshd -p warning --since today

# Extreure només els missatges en JSON per processar amb jq
journalctl -o json | jq '.MESSAGE'
```

11. Conceptes clau

El journal és **binari i persistent** (si es configura), cosa que permet filtrar per camps estructurats --- no pas per regexp sobre text pla. Cada entrada té metadades com `_SYSTEMD_UNIT`, `_PID`, `PRIORITY`, `_HOSTNAME`, etc.

El flux de diagnòstic típic:

1. Primer `journalctl -b -p err` per veure si hi ha errors al boot actual.
2. Després `journalctl -u <servei> --since today` per aprofundir en el servei concret.
3. Si cal seguiment, afegir `-f` al final.

Permisos: cal ser root o pertànyer al grup `systemd-journal` per veure els registres d'altres usuaris o del sistema sencer. Amb l'usuari normal, `journalctl --user` mostra els registres de la sessió pròpia.

12. Comparativa: `/var/log/syslog` vs. `journalctl`

	<code>/var/log/syslog</code> (sistema clàssic)	<code>journalctl</code> (systemd journal)
Format	Text pla, llegible directament	Binari estructurat, cal <code>journalctl</code> per llegir
Eina de lectura	<code>cat</code> , <code>less</code> , <code>tail</code> , <code>grep</code>	<code>journalctl</code> (amb opcions de filtratge avançat)
Metadades	Mínim: timestamp, hostname, procés, missatge	Extens: PID, UID, GID, unitat systemd, prioritat, boot ID, etc.
Persistent per defecte	Sí (fitxers al disc)	Depèn de la distro --- cal <code>/var/log/journal/</code> creat manualment en algunes
Rotació	<code>logrotate</code> (configuració a <code>/etc/logrotate.d/</code>)	Automàtica per mida/temps (<code>journal.d.conf</code>)
Daemon responsable	<code>rsyslog</code> / <code>syslog-ng</code>	<code>systemd-journald</code>

12.1. Operacions habituals

Tasca	Sistema clàssic (syslog)	Sistema modern (journalctl)
Veure registres en temps real	<code>tail -f /var/log/syslog</code>	<code>journalctl -f</code>
Veure les últimes N línies	<code>tail -n 50 /var/log/syslog</code>	<code>journalctl -n 50</code>
Buscar un text	<code>grep "error" /var/log/syslog</code>	<code>journalctl -g "error"</code>
Filtrar per servei	<code>grep "nginx" /var/log/syslog</code>	<code>journalctl -u nginx</code>
Filtrar per data	<code>grep "Jan 15" /var/log/syslog</code>	<code>journalctl --since "2025-01-15"</code>
Filtrar per gravetat	<code>grep -E "error crit" /var/log/syslog</code>	<code>journalctl -p err</code>
Veure missatges del kernel	<code>dmesg</code> o <code>grep "kernel" /var/log/syslog</code>	<code>journalctl -k</code>
Veure l'arrencada anterior	Impossible (sobreescriu)	<code>journalctl -b -1</code>

Exportar en JSON	No possible nativament	<code>journalctl -o json</code>
Veure espai utilitzat	<code>du -sh /var/log/syslog</code>	<code>journalctl --disk-usage</code>
Netejar registres antics	<code>logrotate o rm manual</code>	<code>journalctl --vacuum-time=2weeks</code>

12.2. Filtratge: potència real de la diferència

Cas	Sistema clàssic	Sistema modern
Errors d'un servei avui	<code>grep "nginx" /var/log/syslog grep "\$(date +%b %e)"</code>	<code>journalctl -u nginx --since today -p err</code>
Missatges d'un PID concret	<code>grep "\[1234\]" /var/log/syslog</code>	<code>journalctl _PID=1234</code>
Missatges d'un usuari	No disponible directament	<code>journalctl _UID=1000</code>
Rang de dates exacte	<code>awk complex sobre el timestamp</code>	<code>journalctl --since "09:00" --until "10:30"</code>
Múltiples serveis	<code>grep -E "nginx php-fpm" /var/log/syslog</code>	<code>journalctl -u nginx -u php-fpm</code>

12.3. Estructura dels fitxers

	Sistema clàssic	Sistema modern
Ubicació	<code>/var/log/syslog, /var/log/auth.log, /var/log/kern.log...</code>	<code>/var/log/journal/<machine-id>/</code>
Fragmentació	Un fitxer per tipus de missatge	Tot centralitzat, filtrat per consulta
Fitxers rotats	<code>syslog.1, syslog.2.gz...</code>	Fitxers <code>.journal</code> gestionats automàticament
Lectura de rotats	<code>zcat syslog.2.gz grep ...</code>	<code>journalctl</code> accedeix a tots de forma transparent

12.4. Configuració

	Sistema clàssic	Sistema modern
Fitxer de config	<code>/etc/rsyslog.conf</code>	<code>/etc/systemd/journald.conf</code>
Límit de mida	<code>rotate, size a logrotate.d</code>	<code>SystemMaxUse=, SystemKeepFree=</code>
Retenció temporal	<code>maxage a logrotate.d</code>	<code>MaxRetentionSec=</code>
Nivell mínim a registrar	Configurable per <code>facility/severity a rsyslog</code>	<code>MaxLevelStore= a journald.conf</code>
Aplicar canvis	<code>systemctl restart rsyslog</code>	<code>systemctl restart systemd-journald</code>

12.5. Convivència: els dos sistemes junts

En la majoria de distribucions modernes **ambdós sistemes coexisteixen**:

- `systemd-journald` captura tot i escriu el journal binari.
- `rsyslog` llegeix el journal via socket i **escriu igualment** a `/var/log/syslog` per compatibilitat.

Això significa que en un sistema típic Ubuntu/Debian actual pots fer servir tant `grep /var/log/syslog` com `journalctl` i obtenir resultats equivalents --- però `journalctl` ofereix molt més control.

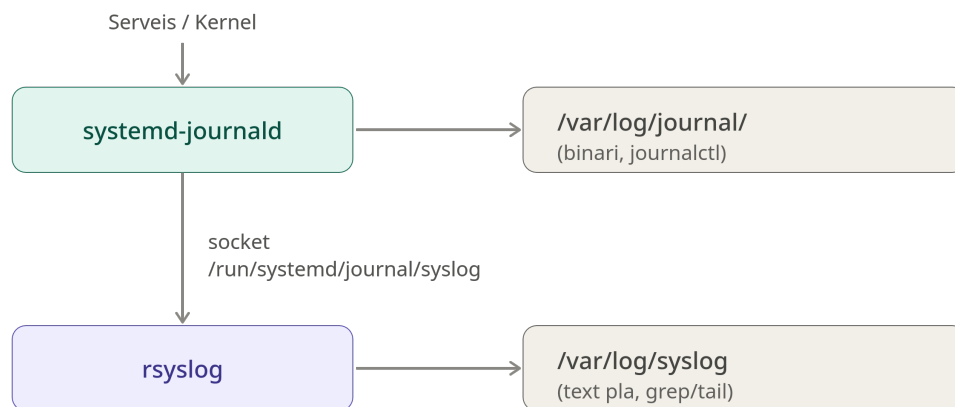


Figura 1: Coexistència

12.6. Quan usar cada eina

Situació	Recomanació
Diagnòstic ràpid d'un servei	<code>journalctl -u <servei> -f</code>
Script que processa logs amb <code>awk/sed</code>	<code>/var/log/syslog o journalctl -o cat</code>
Investigar un crash del sistema	<code>journalctl -b -1 -p err</code>
Integració amb eines externes (Elastic, Graylog)	Exportar via <code>journalctl -o json o rsyslog</code>
Sistema sense <code>systemd</code> (contenidors mínims)	<code>/var/log/</code> clàssic o <code>stdout</code> del contenidor
Auditoria de seguretat	<code>journalctl -u sshd -p warning</code>

Índex de figures

1 Coexistència 7

Versions d'aquest document

- [HTML](#)
- [PDF](#)
- [ODT](#)
- [MD](#)

[Domini Públic \(CC0\)](#)