Logs

Índex

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

6.2. Filtrar per servei

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

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

6.3. Filtrar per temps

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

7. Filtres avançats

7.1. Per prioritat (nivell)

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

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

7.3. Altres filtres útils

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

8. Format de sortida

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

9. Manteniment i espai en disc

Ordre	Descripció
`journalctl --disk-usage`	Quant espai ocupa el journal en disc.
`journalctl --vacuum-size=200M`	Elimina entrades antigues fins que el journal ocupi menys de 200 MB.
`journalctl --vacuum-time=2weeks`	Elimina entrades de fa més de 2 setmanes. Unitats: `s`, `m`, `h`, `days`, `weeks`, `months`, `years`.
`journalctl --rotate`	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`

	`/var/log/syslog` (sistema clàssic)	`journalctl` (systemd journal)
Format	Text pla, llegible directament	Binari estructurat, cal `journalctl` per llegir
Eina de lectura	`cat`, `less`, `tail`, `grep`	`journalctl` (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 `/var/log/journal/` creat manualment en algunes
Rotació	`logrotate` (configuració a `/etc/logrotate.d/`)	Automàtica per mida/temps (`journald.conf`)
Daemon responsable	`rsyslog` / `syslog-ng`	`systemd-journald`

12.1. Operacions habituals

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

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

12.2. Filtratge: potència real de la diferència

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

12.3. Estructura dels fitxers

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

12.4. Configuració

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

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.

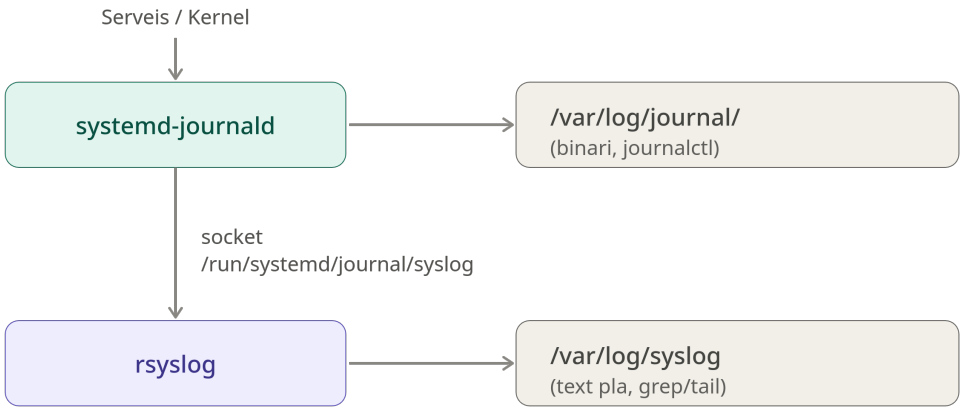

Figura 1: Coexistència

12.6. Quan usar cada eina

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

Versions d'aquest document

- HTML - journalctl.html
- PDF - journalctl.pdf
- ODT - journalctl.odt
- MD - journalctl.md