
Planificació de tasques: at i cron

Índex

1. Introducció	1
2. L'ordre at	1
2.1. Descripció	1
2.2. Instal·lació	1
2.3. Sintaxi bàsica	1
2.4. Especificació del moment	2
2.5. Entrada des de fitxer o pipe	2
2.6. Gestió de tasques pendents	2
2.7. Cues d'at	3
2.8. batch: execució quan el sistema estigui disponible	3
2.9. Control d'accés a at	3
3. cron: tasques periòdiques	4
3.1. Descripció	4
3.2. Instal·lació i estat	4
3.3. El fitxer crontab	4
3.4. Sintaxi d'una entrada crontab	4
3.5. Operadors especials	5
3.6. Exemples d'entrades crontab	5
3.7. Variables d'entorn del crontab	6
3.8. Gestió de la sortida	6
3.9. Directoris de cron del sistema	6
3.10. El fitxer /etc/crontab	7
3.11. Control d'accés a cron	7
4. anacron: cron per a sistemes no continus	7
4.1. Descripció	7
4.2. Fitxer de configuració: /etc/anacrontab	7
5. systemd timers: alternativa moderna a cron	8
6. Taules de dreceres especials de cron	8
7. Bones pràctiques	9
8. Resolució de problemes habituals	9
9. Resum d'ordres (referència ràpida)	9
10. Referències	10

1. Introducció

Linux ofereix dos mecanismes principals per executar ordres de manera automàtica en moments determinats:

Eina	Ús principal
at	Execució puntual en un moment futur concret
cron	Execució periòdica i repetitiva

Ambdós sistemes permeten automatitzar tasques d'administració: còpies de seguretat, enviament de correus, rotació de logs, actualitzacions, neteja de fitxers temporals, etc.

2. L'ordre at

2.1. Descripció

at permet programar l'execució d'una ordre o script **una sola vegada** en un moment futur. Les tasques s'executen amb l'entorn de l'usuari que les ha programat.

2.2. Instal·lació

```
# Debian/Ubuntu
sudo apt install at

# Fedora/RHEL
sudo dnf install at

# Arch/Manjaro
sudo pacman -S at
```

El dimoni associat és atd. Cal assegurar-se que estigui actiu:

```
sudo systemctl enable --now atd
```

2.3. Sintaxi bàsica

```
at [opcions] MOMENT
```

Un cop executada l'ordre, s'obre un intèrpret interactiu on s'escriuen les ordres. Es finalitza amb Ctrl+D.

```
at 15:30
at> echo "Hola" > /tmp/prova.txt
at>
```

2.4. Especificació del moment

at accepta una gran varietat de formats de temps:

```
# Hora concreta
at 14:00
at 2pm
at 14:00 tomorrow
at 14:00 + 2 days
at now + 1 hour
at now + 30 minutes
at midnight
at noon
at teatime          # les 16:00

# Data concreta
at 10:00 25 Dec
at 10:00 2025-12-25
```

2.5. Entrada des de fitxer o pipe

```
# Des d'un fitxer de script
at 03:00 < script_backup.sh

# Des d'una pipe
echo "rsync -av /dades /backup" | at 02:00
```

2.6. Gestió de tasques pendents

```
# Llistar tasques en cua
atq
# o equivalent:
at -l

# Veure el contingut d'una tasca (número de treball)
at -c 3

# Eliminar una tasca
atrm 3
# o equivalent:
at -d 3
```

Exemple de sortida d'atq:

```
3  Mon Jun  9 03:00:00 2025 a ramon
5  Tue Jun 10 14:00:00 2025 a root
```

2.7. Cues d'at

at organitza les tasques en cues identificades per lletres (a--z, A--Z). Per defecte s'usa la cua a. La cua b és per a batch.

```
at -q b 23:00 # encua a la cua 'b'
```

2.8. batch: execució quan el sistema estigui disponible

batch és similar a at, però executa la tasca quan la càrrega del sistema cau per sota d'un llindar (per defecte, load average < 1.5):

```
batch
batch> /usr/bin/ffmpeg -i entrada.mkv sortida.mp4
batch>
```

2.9. Control d'accés a at

Els fitxers /etc/at.allow i /etc/at.deny controlen qui pot usar at:

- Si existeix /etc/at.allow → només els usuaris llistats hi tenen accés.
- Si no existeix /etc/at.allow, però existeix /etc/at.deny → tots els usuaris excepte els llistats hi tenen accés.
- Si cap dels dos existeix → només root hi té accés.

```
# Permetre accés només a ramon i root
echo "ramon" | sudo tee /etc/at.allow
echo "root" | sudo tee -a /etc/at.allow
```

3. cron: tasques periòdiques

3.1. Descripció

cron és el planificador de tasques periòdiques d'Unix/Linux. S'executa com a dimoni (crond o cron) i comprova cada minut si hi ha tasques a executar.

3.2. Instal·lació i estat

```
# Verificar que el dimoni està actiu
sudo systemctl status cron      # Debian/Ubuntu
sudo systemctl status crond     # Fedora/RHEL/Arch

# Habilitar i iniciar
sudo systemctl enable --now cron
```

3.3. El fitxer crontab

Cada usuari pot tenir el seu propi crontab (taula de tasques cron):

```
# Editar el crontab de l'usuari actual
crontab -e

# Llistar les tasques del crontab actual
crontab -l

# Eliminar tot el crontab
crontab -r

# Editar el crontab d'un altre usuari (root)
crontab -u alumne1 -e
```

3.4. Sintaxi d'una entrada crontab

```

┌─── minut          (0--59)
├─── hora           (0--23)
├─── dia del mes    (1--31)
├─── mes            (1--12 o jan--dec)
└─── dia de la setmana (0--7, on 0 i 7 = diumenge, o sun--sat)
* * * * * ORDRE
```

3.5. Operadors especials

Operador	Significat	Exemple
*	Qualsevol valor	* * * * *
,	Llista de valors	0 9,14,18 * * *
-	Rang de valors	0 9-17 * * *
/	Pas (cada N)	*/15 * * * *

3.6. Exemples d'entrades crontab

```
# Cada minut
* * * * * /usr/bin/script.sh

# Cada dia a les 3:00
0 3 * * * /usr/bin/backup.sh

# Cada dilluns a les 8:30
30 8 * * 1 /usr/bin/informe_setmanal.sh

# Cada hora, en punt
0 * * * * /usr/bin/comprovacio.sh

# Cada 15 minuts
*/15 * * * * /usr/bin/monitoratge.sh

# El primer dia de cada mes a les 00:00
0 0 1 * * /usr/bin/informe_mensual.sh

# De dilluns a divendres a les 9:00
0 9 * * 1-5 /usr/bin/obertura.sh

# A les 9:00 i a les 18:00 els dies laborables
0 9,18 * * 1-5 /usr/bin/notificacio.sh

# Cada 5 minuts entre les 8:00 i les 20:00
*/5 8-20 * * * /usr/bin/poll.sh
```

3.7. Variables d'entorn del crontab

El crontab té un entorn limitat. Es poden definir variables al principi del fitxer:

```
SHELL=/bin/bash
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
MAILTO=ramon@thosicodina.cat
HOME=/home/ramon

# Tasca que usa les variables definides
0 3 * * * /home/ramon/scripts/backup.sh
```

- **MAILTO:** envia la sortida de les tasques a aquesta adreça. Si s'assigna buit (MAILTO=""), no s'envia res.

3.8. Gestió de la sortida

Per defecte, cron envia la sortida estàndard i d'errors per correu a l'usuari. Per redirigir-la:

```
# Descartar tota la sortida
0 3 * * * /usr/bin/backup.sh > /dev/null 2>&1

# Guardar la sortida en un fitxer de log
0 3 * * * /usr/bin/backup.sh >> /var/log/backup.log 2>&1

# Guardar sortida estàndard i errors separatament
0 3 * * * /usr/bin/backup.sh > /tmp/backup_out.log 2>
↪ /tmp/backup_err.log
```

3.9. Directoris de cron del sistema

A més dels crontabs d'usuari, el sistema usa directoris especials:

Director	Freqüència
/etc/cron.hourly/	Cada hora
/etc/cron.daily/	Cada dia
/etc/cron.weekly/	Cada setmana
/etc/cron.monthly/	Cada mes
/etc/cron.d/	Fitxers amb sintaxi crontab completa

Els scripts col·locats en aquests directoris s'executen automàticament amb la freqüència indicada (gestionat per run-parts o anacron).

```
# Exemple: afegir un script de neteja diari
sudo cp neteja_tmp.sh /etc/cron.daily/
sudo chmod +x /etc/cron.daily/neteja_tmp.sh
```

3.10. El fitxer /etc/crontab

És el crontab del sistema, amb un camp adicional per especificar l'usuari d'execució:

```
# m h dom mon dow user  command
17 * * * * * root    cd / && run-parts --report
↪ /etc/cron.hourly
25 6 * * * root    test -x /usr/sbin/anacron || ( cd / &&
↪ run-parts --report /etc/cron.daily )
```

3.11. Control d'accés a cron

Igual que at, es controla amb /etc/cron.allow i /etc/cron.deny, amb la mateixa lògica:

- Si existeix /etc/cron.allow → accés restringit als usuaris llistats.
- Si no existeix, però existeix /etc/cron.deny → tots excepte els llistats.

4. anacron: cron per a sistemes no continus

4.1. Descripció

anacron complementa cron en sistemes que no estan encesos les 24 hores (portàtils, estacions de treball). Garanteix que les tasques periòdiques s'executin encara que el sistema estigui apagat en el moment previst.

4.2. Fitxer de configuració: /etc/anacrontab

```
# PERIOD DELAY JOB-ID COMMAND
1 5 cron.daily run-parts /etc/cron.daily
7 10 cron.weekly run-parts /etc/cron.weekly
@monthly 15 cron.monthly run-parts /etc/cron.monthly
```

- **PERIOD**: dies entre execucions.
- **DELAY**: minuts d'espera després de l'arrencada.
- **JOB-ID**: identificador per als logs.

5. systemd timers: alternativa moderna a cron

En sistemes amb systemd, els **timers** són una alternativa potent a cron:

```
# Llistar tots els timers actius
systemctl list-timers

# Exemple: timer que executa un servei cada dia
# /etc/systemd/system/backup.timer
[Unit]
Description=Backup diari

[Timer]
OnCalendar=daily
Persistent=true

[Install]
WantedBy=timers.target
```

Avantatges dels timers systemd respecte a cron: integració amb journald per als logs, gestió de dependències, suport per a Persistent=true (similar a anacron).

6. Taules de dreceres especials de cron

Algunes implementacions de cron admeten macros predefinides:

Macro	Equivalent	Descripció
@reboot	---	Un cop en arrencar el sistema
@hourly	0 * * * *	Cada hora
@daily	0 0 * * *	Cada dia a mitjanit
@midnight	0 0 * * *	Equivalent a @daily
@weekly	0 0 * * 0	Cada diumenge a mitjanit
@monthly	0 0 1 * *	El primer dia del mes
@yearly	0 0 1 1 *	L'1 de gener
@annually	0 0 1 1 *	Equivalent a @yearly

```
# Exemple d'ús al crontab
@reboot /home/ramon/scripts/inici.sh
@daily /home/ramon/scripts/backup.sh
```

7. Bones pràctiques

1. **Rutes absolutes:** usar sempre rutes completes a les ordres i scripts (/usr/bin/python3, no python3).
2. **Entorn limitat:** recordar que cron no carrega el .bashrc; definir PATH i variables necessàries al crontab.
3. **Redirigir la sortida:** evitar l'acumulació de correus redirigint a un log o a /dev/null.
4. **Permisos d'execució:** assegurar-se que els scripts tenen el bit +x.
5. **Provar els scripts manualment** abans d'automatitzar-los.
6. **Comentar les entrades** per facilitar el manteniment:

```
# Backup nocturn de /home - cada dia a les 2:00
0 2 * * * /home/ramon/scripts/backup.sh >> /var/log/backup.log
↪ 2>&1
```

7. **Usar flock** per evitar execucions simultànies de tasques llargues:

```
0 * * * * flock -n /tmp/tasca.lock /home/ramon/scripts/tasca.sh
```

8. **MAILTO=""** si no es vol rebre correu de les tasques.

8. Resolució de problemes habituals

Problema	Causa probable	Solució
La tasca no s'executa	El dimoni cron no està actiu	sudo systemctl start cron
Ordre no trobada	PATH insuficient al cron	Usar rutes absolutes o definir PATH
Script no s'executa	Manca el bit d'execució	chmod +x script.sh
Correu massiu a l'usuari	Sortida no redirigida	Afegir > /dev/null 2>&1 o MAILTO=""
Tasca executada dues vegades	Execució concurrent de tasca llarga	Usar flock
at no funciona	El dimoni atd no està actiu	sudo systemctl start atd

9. Resum d'ordres (referència ràpida)

```
# --- at ---
at 15:00                # Programar tasca puntual
at now + 2 hours        # En 2 hores
at 10:00 < script.sh   # Des d'un fitxer
atq                     # Llistar tasques pendents
atrm 3                  # Eliminar tasca número 3
at -c 3                 # Veure contingut de la tasca 3
batch                   # Executar quan el sistema estigui lliure
```

```
# --- cron ---
crontab -e           # Editar crontab
crontab -l           # Llistar entrades
crontab -r           # Eliminar tot el crontab
crontab -u usuari -e # Editar crontab d'un altre usuari (root)
systemctl status cron # Estat del dimoni

# --- logs ---
grep CRON /var/log/syslog # Debian/Ubuntu
journalctl -u cron        # Sistemes amb systemd
```

10. Referències

- `man at` --- Manual de l'ordre at
- `man atd` --- Manual del dimoni atd
- `man crontab` --- Manual de l'ordre crontab
- `man 5 crontab` --- Format del fitxer crontab
- `man anacron` --- Manual d'anacron
- `man anacrontab` --- Format del fitxer `/etc/anacrontab`
- [Arch Wiki: Cron](#)
- [Arch Wiki: Systemd/Timers](#)

Versions d'aquest document

- HTML - [at_cron.html](#)
- PDF - [at_cron.pdf](#)
- ODT - [at_cron.odt](#)
- MD - [at_cron.md](#)

[Domini Públic \(CC0\)](#)