

AT

Serveix per a programar tasques que s'han d'executar abans que la màquina s'apagui.

atd és el dimoni que s'encarrega de gestionar els treballs d'at.
at gestiona treballs per a una posterior execució.
atq llista les tasques pendents. (at -l)
atrm seguit del nombre de tasca esborra la tasca pendent. (at -d)

Les tasques pendents són arxius executables que estan en el directori /var/spool/cron/atjobs

Exemples:

```
# date
mar abr 12 03:22:31 CEST 2005
# at 03:30 today
warning: commands will be executed using /bin/sh
at> poweroff
at> <EOT> (Aquest EOT surt quan prems 'Ctrl + D')
job 1 at 2005-04-12 03:30
# atq
1            2005-04-12 03:30 a root
# atrm 1
# atq
#
```

CRON

És el dimoni que s'encarrega de gestionar els treballs de crontab i serveix per a programar tasques que s'han d'executar amb certa periodicitat.

```
crontab -e      edita tasques programades.
crontab -l      llista les tasques programades.
crontab -r      esborra tasques programades.
```

```
minuts      hora      dia mes      mes      dia setmana usuari      ordre
de 0 a 59   de 0 a 23   d'1 a 30/31 d'1 a 12   d'1 a 7
```

Exemples:

perquè cada dia la màquina es pari a les 21:00 escriuríem alguna cosa així:

```
0 21 * * * root poweroff
```

Si volem que una tasca s'executi cada 5 hores hem d'usar la barra '/' i escriure alguna cosa semblat a això:

```
* */5 * * * usuari tasca
```

Existeixen quatre carpetes on podem posar executables o enllaços a executables perquè s'executin amb la periodicitat que nosaltres vulguem (horària, diària, setmanal o mensual). Aquestes carpetes són:

```
/etc/cron.hourly
/etc/cron.daily
/etc/cron.weekly
/etc/cron.monthly
```

```
# cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file.
# This file also has a username field, that none of the other crontabs do.
```

```
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
```

```
# m h dom mon dow user  command
17 * * * * root    run-parts --report /etc/cron.hourly
25 6 * * * root    test -x /usr/sbin/anacron || run-parts --report
/etc/cron.daily
47 6 * * 7 root    test -x /usr/sbin/anacron || run-parts --report
/etc/cron.weekly
52 6 1 * * root    test -x /usr/sbin/anacron || run-parts --report
/etc/cron.monthly
#
```

Cron no pot interactuar amb l'entorn gràfic si no afegim l'element `DISPLAY=":0"` abans de l'ordre que volem que s'executi.

```
0 1 * * * DISPLAY=":0" gcacltool
```

Alguns exemples de línies que podem afegir a crontab són:

- Fer un ping a l'equip 192.168.0.1 cada minut, de cada hora, de cada dia, de cada mes. És a dir, fer un ping a cada minut (la terminació en >/dev/null 2>&1 per a no rebre la notificació).

```
* * * * * /sbin/ping -c 1 192.168.0.1 >> /dev/null
```

- Per a executar el mateix, però que el ping només s'executi els dies laborables de la setmana (de Dilluns a Divendres) a les 8:00 PM en punt i només durant el mes d'octubre.

```
0 20 * 10 1-5 /sbin/ping -c 1 192.168.0.1 >> /dev/null
```

- Per a executar un script en python de dilluns a diumenge a mitjanit (0:00 AM) els dies 1, 15 i 28 de cada mes (per a assegurar-nos que en febrer també es faci)

```
0 0 1,15,28 * * /home/mmonrdz/scripts/resp.py
```

- Per a executar un script que actualitza la base de dades de cotitzacions tots els dilluns a les 7:00 AM, 7:50 AM, 8:00 AM y 8:50 AM sense enviar e-mail de notificació.

```
0,50 7,8 * * 1 /home/a/scripts/dbcotiza.pl >/dev/null 2>&1
```

- Per a obrir un terminal de Gnome (consola d'ordres) i executar en ella rtorrent cada dia a la 1 a.m.:

```
0 1 * * * DISPLAY=":0" gnome-terminal --command=rtorrent
```