

# Automazione e monitoraggio

Molte attività devono essere eseguite periodicamente o differite nel tempo.

Di questo si occupano i demoni *crond* e *atd*.

Un esempio tipico di strumenti che possono essere utilizzati interattivamente, ma anche molto spesso invocati automaticamente con una certa cadenza temporale, è quello delle funzioni di monitoraggio dello stato del sistema

---

## Cron

**crond** è un demone (programma residente in memoria in attesa di eventi che lo riguardino) che esamina una serie di file di configurazione ogni minuto, e determina quali compiti specificati nei file debbano essere eseguiti.

I file di configurazione (**crontab**) sono distinti in due insiemi:

- ◆ Uno per utente (**/var/spool/cron/<utente>**) → **crontab -e**
- ◆ System-wide (**/etc/crontab**)

Solitamente quest'ultimo non fa altro che richiamare l'esecuzione di tutto ciò che trova in alcune directory:

**/etc/cron.hourly/**, **/etc/cron.daily/**, **/etc/cron.weekly/**,  
**/etc/cron.monthly/**

## Cron

Ogni crontab contiene un elenco di direttive nella forma

**MINUTO ORA G.MESE MESE G.SETTIMANA <comando>**

Es.

```
* * 27 * * $HOME/bin/paga
30 8-18/2 * * 1-5 $HOME/bin/lavora
00 00 1 1 * /usr/sbin/auguri
30 4 1,15 * 6 /bin/backup
```

---

## At

**atd** è un demone che gestisce code di compiti da svolgere in momenti prefissati. L'interfaccia ad **atd** consiste di 4 comandi:

**at** [-V] [-q queue] [-f file] [-mldbv] TIME  
pianifica un comando al tempo TIME

**atq** [-V] [-q queue] [-v]  
elenca i comandi in coda

**atrm** [-V] job [job...]  
rimuove comandi dalla coda

**batch** [-V] [-q queue] [-f file] [-mv] [TIME]  
esecuzione condizionata al carico

## At

- ◆ Se non viene specificato un file comandi per *at* o *batch*, verrà usato lo standard input.
- ◆ La specifica dell'ora è flessibile e complessa. Per una definizione completa si veda la documentazione in ***/usr/doc/at-<versione>/timespec***. Alcuni esempi:

```
echo 'wall "sveglia"' | at 08:00
echo "$HOME/bin/pulisci" | at now + 2 weeks
echo "$HOME/bin/auguri" | at midnight 25.12.2003
```

## Monitoraggio delle risorse

Un ampio set di comandi consente di ispezionare lo stato delle risorse di un sistema linux in tempo reale e di visualizzare statistiche elaborate su dati raccolti con continuità:

Processi	Utenti	Spazio	File
ps	w	df	fuser
top	last	du	lsuf
uptime		free	
		vmstat	

## Monitoraggio delle risorse – ps

Il già citato *ps* consente di “fotografare” l’elenco dei processi attivi al momento dell’invocazione, con i dettagli di ognuno.

Esercizio: Registrare in un logfile il numero di demoni *httpd* attivi, con una frequenza di campionamento di 1 minuto.

Variante: abbassare la frequenza di campionamento a 5 s.

Esercizio: avvertire immediatamente *root* appena l’utente *xxx* lancia un qualunque processo

## Monitoraggio delle risorse – uptime e free

**uptime** restituisce informazioni molto più sintetiche sulla “storia di carico” del sistema, così come **free** riporta statistiche sulla memoria. La loro invocazione dà il “polso” del sistema:

```
# uptime
9:14am up 33 min,  2 users,  load average: 0.02, 0.03, 0.13

# free -o
      total        used        free      shared    buffers     cached
Mem:   384480     379276         5204         1312       51284     138752
Swap:  128516              0       128516
```

La maggior parte di queste informazioni è accessibile tramite il filesystem speciale **/proc**, questi comandi sono semplicemente interfacce utente.

## Monitoraggio delle risorse – top

**top** è una interfaccia allo stesso tipo di informazioni più adatta alla sorveglianza in tempo reale, in quanto:

- ◆ Aggiorna l'elenco con cadenza configurabile
- ◆ Raccoglie lo stato complessivo del sistema e lo stato dettagliato dei processi in un'unica presentazione
- ◆ Ordina l'elenco dei processi con criteri configurabili
- ◆ Permette la variazione dei criteri di visualizzazione con comandi interattivi, runtime
- ◆ Permette l'invio di segnali ai processi

... sostanzialmente raggruppando le funzionalità di ps, uptime, free e kill in un unico strumento **interattivo**

## Monitoraggio delle risorse – top

```
9:31am up 50 min, 2 users, load average: 0.02, 0.02, 0.04
71 processes: 70 sleeping, 1 running, 0 zombie, 0 stopped
CPU states: 4.3% user, 5.2% system, 0.1% nice, 90.2% idle
Mem: 384480K av, 380688K used, 3792K free, 1312K shrd, 51312K buff
Swap: 128516K av, 0K used, 128516K free, 139136K cached
```

PID	USER	PRI	NI	SIZE	RSS	SHARE	STAT	%CPU	%MEM	TIME	COMMAND
1179	root	13	0	3092	3092	2592	S	2.8	0.8	0:50	magicdev
9299	root	16	0	1044	1040	832	R	2.8	0.2	0:00	top
1	root	8	0	520	520	452	S	0.0	0.1	0:03	init
2	root	9	0	0	0	0	SW	0.0	0.0	0:00	keventd
3	root	9	0	0	0	0	SW	0.0	0.0	0:00	kapm-idled
4	root	19	19	0	0	0	SWN	0.0	0.0	0:00	ksoftirqd_CPU0
5	root	9	0	0	0	0	SW	0.0	0.0	0:00	kswapd
6	root	9	0	0	0	0	SW	0.0	0.0	0:00	kreclaimd
7	root	9	0	0	0	0	SW	0.0	0.0	0:00	bdflush
8	root	9	0	0	0	0	SW	0.0	0.0	0:00	kupdated
9	root	-1	-20	0	0	0	SW<	0.0	0.0	0:00	mdrecoveryd
71	root	9	0	0	0	0	SW	0.0	0.0	0:00	khubd
465	root	9	0	0	0	0	SW	0.0	0.0	0:00	eth0
546	root	9	0	592	592	496	S	0.0	0.1	0:00	syslogd
551	root	9	0	1124	1124	448	S	0.0	0.2	0:00	klogd
569	rpc	9	0	592	592	504	S	0.0	0.1	0:00	portmap
597	rpcuser	9	0	788	788	688	S	0.0	0.2	0:00	rpc.statd

## Monitoraggio delle risorse – df

**df** mostra l'utilizzo dello spazio disco:

Filesystem	1k-blocks	Used	Available	Use%	Mounted on
/dev/hdb1	202220	126789	64991	67%	/
/dev/hdb4	5558076	3916612	1359124	75%	/usr
/dev/hdb3	303336	49822	237851	18%	/var
none	192240	0	192240	0%	/dev/shm
/dev/hda1	10231392	9473248	758144	93%	/win/c
/dev/hda5	9790032	4247000	5543032	44%	/win/d

Esercizio: avvertire *root* quando lo spazio libero di qualunque partizione scende sotto il 10%

Variante: lo script accetta un parametro che indica quale partizione sorvegliare, se il parametro è assente → tutte

## Monitoraggio delle risorse – du

**du** permette di calcolare lo spazio occupato dai file (in una directory). Senza opzioni particolari **du** riporta l'occupazione totale delle dir passate come argomento ed anche di tutte le subdir in esse presenti. Es:

```
# du /tmp
1  /tmp/.font-unix
1  /tmp/.X11-unix
1  /tmp/.ICE-unix
5  /tmp/orbit-root
72 /tmp
```

**du -s** riporta invece il *summary*, senza dettagli sulle subdir.

Esercizio: a mezzanotte inviare a *root* un rapporto sulle dir più ingombranti: 10 di primo livello, 20 di secondo, 30 di terzo.

## Monitoraggio delle risorse – fuser

I processi ed i file sono legati da due relazioni d'uso interessanti per l'amministratore: quali file sta usando un processo e quali processi stanno usando un file.

Alla prima domanda si può rispondere esaminando il filesystem speciale **/proc**, es.:

```
# ls -l /proc/2208/fd/
total 0
lrwx----- 1 root    root    64 Apr 26 10:11 0 -> /dev/pts/0
lrwx----- 1 root    root    64 Apr 26 10:11 1 -> /dev/pts/0
lrwx----- 1 root    root    64 Apr 26 10:11 2 -> /dev/pts/0
lr-x----- 1 root    root    64 Apr 26 10:11 3 -> /etc/man.config
```

**Esercizio: elencare tutti i file impegnati da un utente per mezzo dei suoi processi**

## Monitoraggio delle risorse – fuser

Il comando **fuser** calcola la risposta alla seconda domanda, e la visualizza insieme ad utili informazioni accessorie:

```
# fuser /etc/man.config
/etc/man.config:      2208    2212    2213    2219
```

**fuser** permette anche di indagare un intero filesystem:

```
# fuser -m /var
/var:                546    597c    714    714c    879c    898    916
916c    964    1013    1020    1021    1318    6493    9244    9244m    9249    9249m
9275    9275m
```

```
c    current directory.
e    executable being run.
f    open file. f is omitted in default display mode.
r    root directory.
m    mmap'ed file or shared library.
```