

Utilizzo di CRONTAB per schedulare processi (con esempi) sotto Linux



Nei sistemi operativi Linux, il comando **crontab** consente di pianificare l'esecuzione automatica periodica di attività o script. Ogni attività schedata è chiamata **cron job**.

Vediamo come usare **cron** per pianificare attività giornaliere, settimanali, orarie, ecc ... sul vostro sistema Linux. La premessa è che possiate accedere al terminale del sistema Linux.

Il formato di crontab

Ogni riga del crontab presenta una sequenza di campi, divisi da uno spazio, come in questo esempio

```
0 23 * * * /root/mybackup.sh
```

Dobbiamo leggerlo così: al minuto 0 delle ore 23, tutti i giorni del mese, tutti i mesi, tutti i giorni della settimana, esegui lo script `/root/mybackup.sh`. In poche parole, esegue lo script tutti i giorni alle ore 23.00. L'asterisco significa "tutti".

Vediamo nel dettaglio tutti i possibili valori di questi campi, in ordine di inserimento

Campo	Valore
Minuti	0 - 59
Ore	0 - 23

Giorno del mese	1 - 31
Mese	1 - 12
Giorno della settimana	0 - 7*
Comando	path dello script da eseguire

* 0 rappresenta la domenica, 1 lunedì.... 7 sabato.

Adesso mettiamo in pratica quanto appreso.

Per aggiungere/modificare/eliminare le attività schedulate

```
# crontab -e
```

Digitare quindi il carattere "i" (insert) per editare il crontab

Adesso possiamo scrivere le nostre righe di crontab.

Salviamo infine il crontab digitando ":wq" (write and quite)
Se invece non vogliamo salvare la modifica digitiamo semplicemente ":q" (quite)

Per verificare le attività schedulate, senza poterle modificare

```
# crontab -l
```

Vediamo alcuni esempi pratici

```
*/20 * * * * /root/mybackup.sh
```

Lo script viene eseguito ogni 20 minuti

```
0 23 * * * /root/mybackup.sh
```

Lo script viene eseguito alle ore 23.00 tutti i giorni

```
0 9,23 * * * /root/mybackup.sh
```

Lo script viene eseguito alle ore 09.00 e alle ore 23.00 tutti i giorni

```
0 7 * * 0 /root/mybackup.sh
```

Lo script viene eseguito alle ore 07.00 della domenica

```
0 7 * 5 6 /root/mybackup.sh
```

Lo script viene eseguito alle ore 07.00 ogni sabato del mese di maggio

```
15 7 * 5 6 /root/mybackup.sh
```

Lo script viene eseguito alle ore 07.15, ogni sabato di maggio

```
* * * * * /root/mybackup.sh
```

Lo script viene eseguito ogni minuto

```
* /5 * * * * /root/mybackup.sh
```

Lo script viene eseguito ogni 5 minuti

```
0 17 * * 1,5 /root/mybackup.sh
```

Lo script viene eseguito alle ore 17.00 di lunedì e domenica

```
0 */5 * * * /root/mybackup.sh
```

Lo script viene eseguito ogni 5 ore (05.00, 10.00, ...)

```
* * * * * sleep 30; /root/mybackup.sh
```

Lo script viene eseguito ogni 30 secondi

Vediamo alcuni casi in cui possiamo utilizzare degli alias

```
0 * * * * /root/mybackup.sh
oppure
@hourly /root/mybackup.sh
Lo script viene eseguito ogni ora
0 0 * * * /root/mybackup.sh
oppure
@daily /root/mybackup.sh
Lo script viene eseguito ogni giorno alle 00.00
0 0 * 0 0 /root/mybackup.sh
oppure
@weekly /root/mybackup.sh
Lo script viene eseguito ogni settimana (alle 00.00 di ogni domenica)
0 0 1 * * /root/mybackup.sh
oppure
@monthly /root/mybackup.sh
Lo script viene eseguito il primo di ogni mese (alle 00.00)
0 0 1 1 * /root/mybackup.sh
oppure
@yearly /root/mybackup.sh
Lo script viene eseguito il primo giorno dell'anno (alle 00.00)
0 0 1 1 * /root/mybackup.sh
oppure
@yearly /root/mybackup.sh
Lo script viene eseguito il primo giorno dell'anno (alle 00.00)
0 0 1 1 * /root/mybackup.sh
oppure
@yearly /root/mybackup.sh
Lo script viene eseguito il primo giorno dell'anno (alle 00.00)
```

E' possibile lanciare più script contemporaneamente, separandoli con il ";"

```
0 0 * * * /root/mybackup.sh; /home/myscript.sh
I due script vengono eseguiti ogni giorno alle ore 00.00
```

Vediamo come lanciare uno script al riavvio del server Linux

```
@reboot /root/backup.sh
```

Per verificare se i crontab sono stati eseguiti, leggiamo il log di cron

```
$ tail /var/log/cron
Nov 30 23:30:00 mioserver crond[18340]: (root) CMD (/bin/sh /root/mybackup.sh &)
Dec 1 00:03:00 mioserver crond[20399]: (pippo) CMD (/bin/sh /root/myscript.sh &)
```

Il log contiene le seguenti informazioni:
Timestamp: la data di esecuzione del cron job
Hostname: il nome host del server su cui è eseguito il cron
Il demone cron e, tra parentesi quadre, il PID del processo
Username: il nome utente che esegue il cron job, tra parentesi tonde
CMD: il comando/script da eseguire