

# Gestione dei compiti

In questo capitolo imparerai come gestire le attività programmate.

---

**Obiettivi** : In questo capitolo, futuri amministratori Linux impareranno come:

- ✓ Linux si occupa della pianificazione dei compiti;
- ✓ limitare l'uso di `cron` a determinati utenti;
- ✓ pianificare le attività.

🚩 **crontab, crond, pianificazione, linux**

**Conoscenza:** ★★

**Complessità:** ★★

**Tempo di lettura:** 15 minuti

---

## Generalità

La pianificazione delle attività è gestita con l'utilità `cron`. Essa permette l'esecuzione periodica dei compiti.

È riservata all'amministratore per le attività di sistema ma può essere utilizzata da utenti normali per attività o script a cui hanno accesso. Per accedere all'utilità `cron`, usiamo: `crontab`.

Il servizio `cron` è usato per:

- Operazioni di amministrazione ripetitive;
- Backups;
- Monitoraggio dell'attività del sistema;
- Esecuzione di un programma.

`crontab` è un'abbreviazione per **cron table**, ma può essere pensato come una tabella di programmazione attività.

### **Attenzione**

Per impostare una pianificazione, il sistema deve avere l'ora locale impostata correttamente.

## Come funziona il servizio

Il servizio di `cron` è gestito da un demone `crond` presente in memoria.

Per verificare il suo stato:

```
[root] # systemctl status crond
```

### **Suggerimento**

Se il demone `crond` non è in esecuzione, dovrai iniziarlo manualmente e/o automaticamente all'avvio. Infatti, anche se i compiti sono pianificati, non saranno lanciati.

Inizializzazione manuale del demone `crond`:

```
[root]# systemctl {status|start|restart|stop} crond
```

Inizializzazione del demone `crond` all'avvio del sistema:

```
[root]# systemctl enable crond
```

## Sicurezza

Per implementare una pianificazione, un utente deve disporre dell'autorizzazione all'utilizzo del servizio `cron`.

Questa autorizzazione varia in base alle informazioni contenute nei file seguenti:

- `/etc/cron.allow`
- `/etc/cron.deny`

### **Attenzione**

Se nessuno dei file è presente, tutti gli utenti possono usare `cron`.

I files `cron.allow` and `cron.deny`

File `/etc/cron.allow`

Solo gli utenti contenuti in questo file sono autorizzati a utilizzare `cron`.

Se esiste ed è vuoto, nessun utente può usare `cron`.

### **Attenzione**

Se `cron.allow` è presente, `cron.deny` è **ignorato**.

File `/etc/cron.deny`

Gli utenti di questo file non sono autorizzati a utilizzare `cron`.

Se è vuoto, tutti gli utenti possono usare `cron`.

Per impostazione predefinita, `/etc/cron.deny` esiste ed è vuoto e `/etc/cron.allow` non esiste.

## Consentire ad un utente

Solo **user1** sarà in grado di utilizzare `cron`.

```
[root]# vi /etc/cron.allow
user1
```

## Proibire ad un utente

Solo **user2** non sarà in grado di usare `cron`.

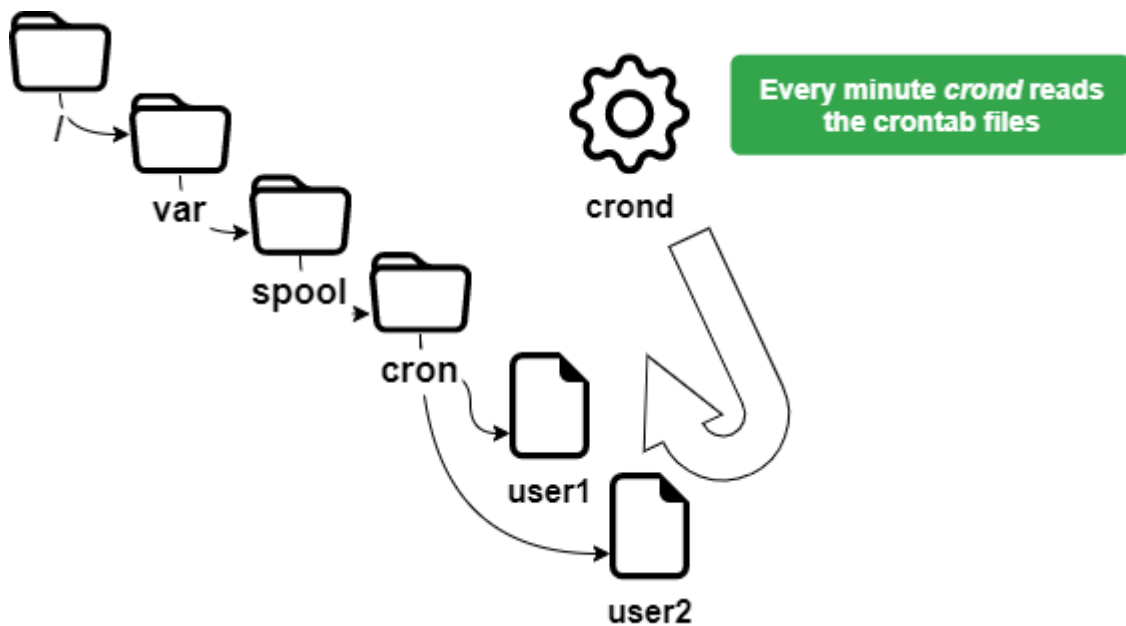
```
[root]# vi /etc/cron.deny
user2
```

`cron.allow` non deve essere presente.

## Pianificazione delle attività

Quando un utente pianifica un'attività, viene creato un file con il suo nome in `/var/spool/cron/`.

Questo file contiene tutte le informazioni che il `cron` deve sapere riguardo a tutte le attività create da questo utente, i comandi o i programmi da eseguire e quando eseguirli (ora, minuto, giorno ...).



### Il comando `crontab`

Il comando `crontab` viene utilizzato per gestire il file di pianificazione.

```
crontab [-u user] [-e | -l | -r]
```

Esempio:

```
[root]# crontab -u user1 -e
```

Opzione	Descrizione
<code>-e</code>	Modifica il file di pianificazione con VI
<code>-l</code>	Visualizza il contenuto del file di pianificazione
<code>-u</code>	Nome dell'utente il cui file di pianificazione deve essere manipolato

Opzione	Descrizione
<code>-r</code>	Elimina il file di pianificazione

### ⚠ Attenzione

`crontab` senza opzione elimina il vecchio file di pianificazione e attende che l'utente inserisca nuove righe. Devi premere `ctrl` + `d` per uscire da questa modalità di modifica.

Solo `root` può utilizzare l'opzione `-u utente` per gestire il file di pianificazione di un altro utente.

L'esempio sopra consente a root di pianificare un'attività per l'utente1.

## Usi di `crontab`

Gli usi di `crontab` sono molti e includono:

- Modifiche ai file `crontab` presi in considerazione immediatamente;
- Nessun bisogno di riavviare.

D'altra parte, devono essere presi in considerazione i seguenti punti:

- Il programma deve essere autonomo;
- Fornire reindirizzamenti (stdin, stdout, stderr);
- Non è rilevante per eseguire comandi che utilizzano richieste di ingresso/uscita su un terminale.

### 📌 Nota

È importante capire che lo scopo della programmazione è quello di eseguire i compiti automaticamente, senza la necessità di un intervento esterno.

## Il file `crontab`

Il file `crontab` è strutturato in base alle seguenti regole.

- Ogni riga di questo file corrisponde a una pianificazione;
- Ogni linea ha sei campi, 5 per il tempo e 1 per l'ordine;
- Ogni campo è separato da uno spazio o da una tabulazione;
- Ogni linea termina con un ritorno a capo;
- Un `#` all'inizio della linea la commenta.

```
[root]# crontab -e
10 4 1 * * /root/scripts/backup.sh
1 2 3 4 5      6
```

Campo	Descrizione	Dettaglio
1	Minuto(i)	Da 0 a 59
2	Ora(e)	Da 0 a 23
3	Giorno(i) del mese	Da 1 a 31
4	Mese dell'anno	Da 1 a 12
5	Giorno(i) della settimana	Da 0 a 7 (0=7=Domenica)
6	Compito da eseguire	Comando completo o script

### **Attenzione**

Le attività da eseguire devono utilizzare percorsi assoluti e, se possibile, usare reindirizzamenti.

Al fine di semplificare la notazione per la definizione del tempo, è consigliabile utilizzare simboli speciali.

Wildcards	Descrizione
*	Tutti i possibili valori del campo
-	Indica una gamma di valori

Wildcards	Descrizione
,	Indica un elenco di valori
/	Definisce un passo

Esempi:

Script eseguito il 15 Aprile alle 10:25am:

```
25 10 15 04 * /root/scripts/script > /log/...
```

Esegui alle 11am e quindi alle 4pm di ogni giorno:

```
00 11,16 * * * /root/scripts/script > /log/...
```

Esegui ogni ora dalle 11am alle 4pm di ogni giorno:

```
00 11-16 * * * /root/scripts/script > /log/...
```

Esegui ogni 10 minuti durante l'orario di lavoro:

```
*/10 8-17 * * 1-5 /root/scripts/script > /log/...
```

Per l'utente root, `crontab` ha anche alcune impostazioni speciali del tempo:

Impostazioni	Descrizione
@reboot	Eeguire il comando al riavvio del sistema
@hourly	Esegui il comando ogni ora
@daily	Esegui giornalmente dopo la mezzanotte
@weekly	Esegui il comando ogni domenica dopo la mezzanotte
@monthly	Esegui il comando il primo giorno del mese subito dopo mezzanotte
@annually	Esegui il 1 gennaio subito dopo mezzanotte

## Processo di esecuzione dell'attività

Un utente, rockstar, vuole modificare il suo file `crontab` :

1) `crond` controlla se è permesso ( `/etc/cron.allow` e `/etc/cron.deny` ).

2) Se lo è, accede al file `crontab` ( `/var/spool/cron/rockstar` ).

Ogni minuto `cron` legge il file di pianificazione.

3) Esegue le attività pianificate.

4) Riporta sistematicamente in un file di registro ( `/var/log/cron` ).

---

Ultimo aggiornamento: 6 gennaio 2022