

# La gestione dei processi in Linux

Prof. Fedeli Massimo

Dicembre 2025



# Indice

0.1	La gestione dei processi in Linux os . . . . .	4
0.2	Comprendere i Processi . . . . .	4
0.3	Elencare i Processi . . . . .	4
0.3.1	Elencare i processi con ps . . . . .	5
0.4	Elencare e modificare i processi con top . . . . .	5
0.5	Elencare i processi con System Monitor . . . . .	6
0.6	Gestione dei Processi in Background e Foreground . . . . .	7
0.6.1	Avviare processi in background . . . . .	7
0.7	Terminazione e Modifica della Priorità dei Processi . . . . .	7
0.7.1	Terminazione dei processi con kill e killall . . . . .	7
0.8	Limitazione dei Processi con cgroups . . . . .	8
0.9	Riassunto . . . . .	8

## 0.1 La gestione dei processi in Linux os

In questo documento ci occupiamo di:

- Visualizzazione dei processi
- Esecuzione dei processi in foreground e background
- Terminazione e modifica della priorità dei processi

Oltre a essere un sistema operativo multiutente, Linux è un sistema multitasking. Multitasking significa che molti programmi possono essere eseguiti contemporaneamente. Un'istanza di un programma in esecuzione è denominata processo. Linux fornisce strumenti per elencare i processi in esecuzione, monitorare l'utilizzo del sistema e terminare (o uccidere) i processi quando necessario.

Da una shell, è possibile avviare processi e poi metterli in pausa, fermarli o ucciderli. È anche possibile metterli in background e riportarli in foreground.

## 0.2 Comprendere i Processi

Un processo è un'istanza in esecuzione di un comando. Ad esempio, potrebbe esserci un solo comando `vi` sul sistema. Ma se `vi` è attualmente in esecuzione da 15 utenti diversi, quel comando è rappresentato da 15 processi in esecuzione diversi.

Un processo è identificato sul sistema da ciò che è denominato ID di processo (**PID**). Quel PID è unico per il sistema corrente. In altre parole, nessun altro processo può usare quel numero come suo ID di processo mentre il primo processo è ancora in esecuzione. Tuttavia, dopo che un processo è terminato, un altro processo può riutilizzare quel numero.

Insieme a un numero ID di processo, altri attributi sono associati a un processo. Ogni processo, quando viene eseguito, è associato a un particolare account utente e account gruppo. Quelle informazioni sull'account aiutano a determinare quali risorse di sistema il processo può accedere. Ad esempio, un processo eseguito come utente root ha molto più accesso ai file e alle risorse di sistema rispetto a un processo in esecuzione come utente regolare.

La capacità di gestire i processi sul sistema è cruciale per un amministratore di sistema Linux. A volte, processi fuori controllo possono ridurre le prestazioni del sistema.

## 0.3 Elencare i Processi

Dal prompt dei comandi, il comando `ps` è il più antico e comune per elencare i processi attualmente in esecuzione sul sistema. La versione Linux di `ps` contiene una varietà di opzioni da vecchi sistemi UNIX e BSD, alcune delle quali sono conflittuali e implementate in modi non standard. Vedi la pagina man di `ps` per descrizioni di quelle opzioni diverse.

Il comando `top` fornisce un approccio più orientato allo schermo per elencare i processi, e può anche essere usato per cambiare lo stato dei processi. Se stai usando il desktop GNOME, puoi usare lo strumento System Monitor (`gnome-system-monitor`) per fornire un mezzo grafico per lavorare con i processi. Questi comandi sono descritti nelle sezioni seguenti.

### 0.3.1 Elencare i processi con ps

L'utilità più comune per controllare i processi in esecuzione è il comando **ps**. Usalo per vedere quali programmi sono in esecuzione, le risorse che stanno usando e chi li sta eseguendo. Ecco un esempio del comando **ps**:

\$ ps u									
USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	
TIME	COMMAND								
jake	2147	0.0	0.7	1836	1020	tty1	S+	14:50	
0:00	-bash								
jake	2310	0.0	0.7	2592	912	tty1	R+	18:22	
0:00	ps u								

In questo esempio, l'opzione **u** (equivalente a **-u**) richiede che vengano mostrati i nomi utente, così come altre informazioni come l'ora di inizio del processo e l'uso della memoria e della CPU per i processi associati all'utente corrente. I processi mostrati sono associati al terminale corrente (**tty1**). Il concetto di terminale deriva dai vecchi tempi quando le persone lavoravano esclusivamente da terminali a caratteri, quindi un terminale tipicamente rappresentava una singola persona su uno schermo singolo. Oggigiorno, puoi avere molti "terminali" su uno schermo aprendo più terminali virtuali o finestre Terminal sul desktop.

In questa sessione shell, non sta accadendo molto. Il primo processo mostra che l'utente di nome **jake** ha aperto una shell **bash** dopo il login. Il processo successivo mostra che **jake** ha eseguito il comando **ps u**. Il dispositivo terminale **tty1** è usato per la sessione di login. La colonna **STAT** rappresenta lo stato del processo, con **R** che indica un processo attualmente in esecuzione e **S** che rappresenta un processo in sleep.

Molti processi in esecuzione su un computer non sono associati a un terminale. Un normale sistema Linux ha molti processi in esecuzione in background. I processi di sistema in background eseguono compiti come il logging dell'attività di sistema o l'ascolto di dati in arrivo dalla rete. Sono spesso avviati quando Linux si avvia e continuano a funzionare fino allo spegnimento del sistema.

Per scorrere tutti i processi in esecuzione sul sistema Linux per l'utente corrente, aggiungi il pipe (**|**) e il comando **less** a **ps ux**:

```
$ ps ux | less
```

Per scorrere tutti i processi in esecuzione per tutti gli utenti sul sistema, usa il comando **ps aux** come segue:

```
$ ps aux | less
```

Il comando **ps** può essere personalizzato per visualizzare colonne selezionate di informazioni e per ordinare le informazioni per una di quelle colonne.

## 0.4 Elencare e modificare i processi con top

Il comando **top** fornisce un mezzo orientato allo schermo per visualizzare i processi in esecuzione sul sistema. Con **top**, l'impostazione predefinita è visualizzare i processi in base a quanta CPU stanno consumando attualmente. Tuttavia, puoi ordinare per altre colonne. Dopo aver identificato un processo che si comporta male, puoi anche usare **top** per ucciderlo (terminarlo completamente) o **renice** (riprioritarlo).

Se vuoi essere in grado di uccidere o renice qualsiasi processo, devi eseguire **top** come utente root. Se vuoi solo visualizzare i processi e possibilmente uccidere o modificare i tuoi processi, puoi farlo come utente regolare.

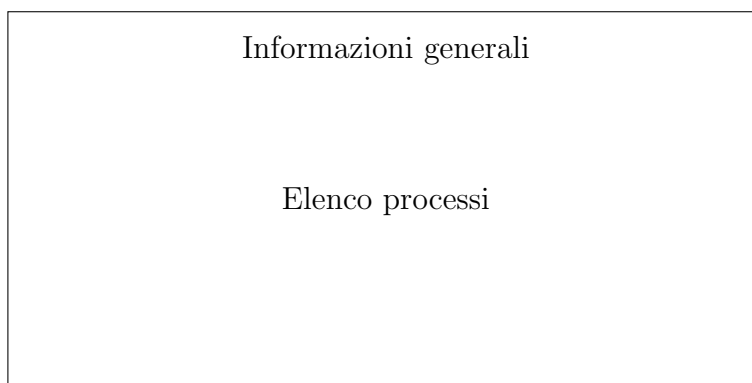


Figura 1: Rappresentazione di top.

## 0.5 Elencare i processi con System Monitor

Se hai il desktop GNOME disponibile sul tuo sistema Linux, System Monitor (**gnome-system-monitor**) è disponibile per fornire un modo più grafico di visualizzare i processi sul sistema. Ordini i processi cliccando sulle colonne. Puoi fare clic destro sui processi per fermarli, ucciderli o renice.

Per avviare System Monitor dal desktop GNOME, premi il tasto Windows e poi digita System Monitor e premi Enter. Poi seleziona la scheda Processes.

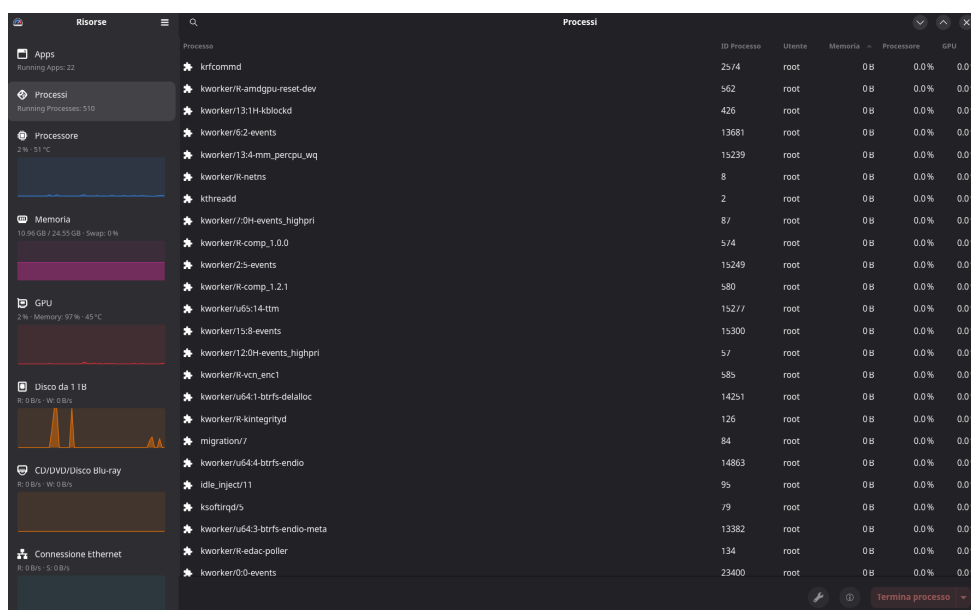


Figura 2: Finestra System Monitor.

## 0.6 Gestione dei Processi in Background e Foreground

Se stai usando Linux su una rete o da un terminale dumb (un monitor che permette solo input testuale senza supporto GUI), la tua shell potrebbe essere tutto ciò che hai. Potresti essere abituato a un ambiente grafico in cui hai molti programmi attivi contemporaneamente per poter passare tra di essi secondo necessità. Questa cosa della shell può sembrare piuttosto limitata.

Sebbene la shell bash non includa una GUI per eseguire molti programmi contemporaneamente, ti permette di spostare programmi attivi tra background e foreground. In questo modo, puoi avere molte cose in esecuzione e scegliere selettivamente quella con cui vuoi lavorare al momento.

Puoi mettere un programma attivo in background in diversi modi. Uno è aggiungere un ampersand (&) alla fine di una riga di comando quando esegui il comando per la prima volta. Puoi anche usare il comando `at` per eseguire comandi in modo che non siano connessi alla shell.

Per fermare un comando in esecuzione e metterlo in background, premi Ctrl+Z. Dopo che il comando è fermato, puoi riportarlo in foreground per eseguirlo (`fg`) o avviarlo in background (`bg`).

### 0.6.1 Avviare processi in background

Se hai programmi che vuoi eseguire mentre continui a lavorare nella shell, puoi metterli in background. Per mettere un programma in background al momento dell'esecuzione, digita un ampersand (&) alla fine della riga di comando, come questo:

```
$ find /usr > /tmp/allusrfiles &  
[3] 15971
```

Background

Figura 3: Processo in background.

## 0.7 Terminazione e Modifica della Priorità dei Processi

Proprio come puoi cambiare il comportamento di un processo usando strumenti grafici come System Monitor, puoi anche usare strumenti da riga di comando per uccidere un processo o cambiare la sua priorità CPU. Il comando `kill` può inviare un segnale di kill a qualsiasi processo per terminarlo, assumendo che tu abbia i permessi per farlo. Può anche inviare segnali diversi a un processo per cambiare altrimenti il suo comportamento. I comandi `nice` e `renice` possono essere usati per impostare o cambiare la priorità del processore di un processo.

### 0.7.1 Terminazione dei processi con `kill` e `killall`

Sebbene solitamente usato per terminare un processo in esecuzione, i comandi `kill` e `killall` possono essere usati per inviare qualsiasi segnale valido a un processo in esecu-

zione. Oltre a dire a un processo di terminare, un segnale potrebbe dire a un processo di rileggere i file di configurazione, mettere in pausa (stop), o continuare dopo essere stato in pausa, solo per nominarne alcuni.

I segnali sono rappresentati sia da numeri che da nomi. I segnali che potresti inviare più comunemente da un comando includono SIGKILL (9), SIGTERM (15) e SIGHUP (1).

Segnale	Numero	Descrizione
SIGHUP	1	Hang-up rilevato sul terminale controllante
SIGINT	2	Interruzione dalla tastiera
SIGQUIT	3	Quit dalla tastiera
SIGABRT	6	Segnale di abort da abort(3)
SIGKILL	9	Segnale di kill
SIGTERM	15	Segnale di terminazione
SIGCONT	19,18,25	Continua se fermato
SIGSTOP	17,19,23	Ferma processo

Tabella 1: Segnali disponibili in Linux.

## 0.8 Limitazione dei Processi con cgroups

Puoi usare una feature come "nice" per dare a un singolo processo più o meno accesso al tempo CPU. Impostare il valore nice per un processo, tuttavia, non si applica ai processi figli che un processo potrebbe avviare o ad altri processi correlati che fanno parte di un servizio più grande. In altre parole, "nice" non limita la quantità totale di risorse che un particolare utente o applicazione può consumare da un sistema Linux.

Man mano che il cloud computing prende piede, molti sistemi Linux saranno usati più come hypervisor che come computer generici. La loro memoria, potenza di elaborazione e accesso allo storage diventeranno commodities da condividere tra molti utenti. In quel modello, deve essere fatto di più per controllare la quantità di risorse di sistema a cui un particolare utente, applicazione, container o macchina virtuale in esecuzione su un sistema Linux ha accesso.

Ecco dove entrano in gioco i cgroups.

I cgroups possono essere usati per identificare un processo come task, appartenente a un particolare gruppo di controllo. I task possono essere impostati in una gerarchia dove, ad esempio, potrebbe esserci un task chiamato daemons che imposta limitazioni predefinite per tutti i processi daemon server, poi subtask che potrebbero impostare limiti specifici su un daemon server web (httpd) per servizio FTP daemon (vsftpd).

## 0.9 Riassunto

Anche su un sistema Linux dove non c'è molta attività, tipicamente decine o persino centinaia di processi sono in esecuzione in background. Usando gli strumenti descritti in questo capitolo, puoi visualizzare e gestire i processi in esecuzione sul tuo sistema.