



LINUX

Gestione dei Processi

Ing. Simone Giustetti
studiosg@giustetti.net

I Processi

Linux è un sistema **multitask** in grado di gestire molte attività in parallelo.

Ogni programma, demone, servizio o utente è gestito attraverso **uno o più processi**.

Linux consente di controllare ogni processo che giri sul sistema.

Linux consente di monitorare le risorse allocate da un processo, mutarne la priorità, inviargli segnali di sistema ed altro ancora.



Definizione di Processo

Un file eseguibile residente sul disco costituisce un programma. **Un programma avviato e caricato in memoria viene chiamato processo.**

Un programma può essere lanciato più volte perciò possiamo definire il processo come **l'istanza di un programma.**

I processi hanno un ciclo di vita simile agli organismi viventi: nascita (Viene lanciato), vita (È in esecuzione) e morte (Viene terminato).



Il PID

Ogni processo viene identificato in maniera univoca da un numero intero positivo chiamato **PID** (Process ID).

Il PID non muta durante il ciclo di vita di un processo.

I PID sono assegnati in maniera incrementale. I processi “anziani” hanno PID basso.

Noto il PID è possibile agire sul processo.



Il PPID

A ciascun processo viene associato, oltre al PID, un secondo identificatore detto **PPID** (Parent Process ID).

Il PPID è un numero intero positivo **non univoco** che indica il processo genitore.

Ogni processo può lanciare altri processi, che assumeranno il ruolo di figlio.

Un processo figlio termina automaticamente quando termina il processo che lo ha originato.



Elenco dei Processi

L'elenco dei processi che girano può essere visualizzato lanciando il comando **ps**.

ps rende il PID e la riga di comando con cui è stato lanciato il programma.

PID	TTY	TIME	CMD
2415	pts/0	00:00:00	su
2424	pts/0	00:00:00	bash
4688	pts/0	00:00:00	vb_start-1.2.sh
4729	pts/0	00:00:32	VirtualBox
5174	pts/0	00:00:19	dolphin
6980	pts/0	00:00:00	ps



Elenco dei Processi

Lanciato senza opzioni, **ps** considera solo i processi della sessione corrente.

Per vedere tutti i processi di sistema, compresi quelli degli altri utenti, di root e dei demoni, bisogna usare l'opzione **-e**.

ps -e Mostra l'elenco di tutti i processi



Elenco dei Processi

L'opzione *-f* aggiunge informazioni utili all'elenco reso da **ps**.

ps -f

UID	PID	PPID	C	STIME	TTY	TIME	CMD
root	2415	2400	0	10:45	pts/0	00:00:00	su -
root	2424	2415	0	10:45	pts/0	00:00:00	-su
root	4688	1	0	15:06	pts/0	00:00:00	
							/bin/sh ./script_start/vb_start-1.2.sh
root	4729	4688	0	15:06	pts/0	00:00:33	
							/opt/VirtualBox/VirtualBox
root	5174	2424	0	15:42	pts/0	00:00:19	dolphin
root	7033	2424	0	17:54	pts/0	00:00:00	ps -f



Analisi delle Prestazioni

Il modo più semplice per monitorare le prestazioni dei processi è lanciare il comando **top**.

```
top - 18:04:47 up 8:09, 5 users, load average: 0.76, 0.73, 0.69
Tasks: 255 total, 2 running, 253 sleeping, 0 stopped, 0 zombie
%Cpu(s): 6.3 us, 1.9 sy, 0.0 ni, 91.9 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 7407.2 total, 2014.4 free, 1897.2 used, 3495.6 buff/cache
MiB Swap: 8192.0 total, 7413.7 free, 778.3 used. 5168.5 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1971	repo	20	0	3219204	628052	108476	R	17.9	8.3	34:37.92	seamonkey
1622	root	20	0	1354072	114132	83412	S	8.0	1.5	26:13.02	Xorg
2388	repo	20	0	1079868	48252	40340	S	6.0	0.6	0:10.74	konsole
2658	repo	20	0	263560	2560	1312	S	2.0	0.0	8:16.54	xmms
14	root	20	0	0	0	0	I	1.3	0.0	4:00.09	rcu_preempt
692	root	-2	0	0	0	0	S	0.7	0.0	3:10.50	gfx
4760	root	20	0	1519612	8036	5736	S	0.7	0.1	0:38.66	VBoxSVC
7105	root	20	0	5456	3484	2648	R	0.7	0.0	0:00.04	top
...											



top funziona in maniera iterativa, ricalcolando le statistiche mostrate ogni 2 o 3 secondi.

È possibile modificare al volo le informazioni rese da top premendo opportuni tasti:

h: Mostra un breve manuale in linea.

i: Mostra solo i processi attivi.

k: richiede il PID di un processo da terminare.

m: Visualizzazione grafica della memoria.

q: Chiude il programma.

z: Attiva la colorazione delle righe.



Background e Foreground

Un processo può girare in due modalità:

- Background: Il processo **non blocca** il sistema / terminale perché gira “sullo sfondo”.
- Foreground: Il processo blocca il sistema / terminale fino alla sua conclusione, quando rende il controllo al chiamante.

La prima modalità è adatta a demoni e servizi che devono girare per lunghi intervalli di tempo.

La seconda per i programmi interattivi usati dagli utenti.



Processi in Background

Mettendo in background i processi, è possibile far girare più programmi contemporaneamente.

Per lanciare un programma in background bisogna accodare il carattere “&” (E commerciale) alla riga di comando:

```
find / -name “*test*” &
```

È comodo salvare l’output in un file dedicato:

```
find / -name “*test*” > out.log 2>&1 &
```



Processi in Background

Un processo attivo può essere messo in background con la seguente procedura:

- 1) Bloccare il processo con la combinazione di tasti CTRL + z.
- 2) Lanciare il comando **bg**.

L'elenco dei processi in background può essere visualizzato con il comando **jobs**.

jobs Rende l'elenco.

jobs -l Rende l'elenco corredato dai PID.



Processi in Foreground

Un processo può essere riattivato con la seguente procedura:

- 1) Determinare L'identificatore del job lanciando il comando **jobs**.
- 2) Lanciare il comando **fg** <Id del job>.

```
jobs -/
```

```
[1]+  5174 Running  dolphin > /dev/null 2>&1 &  
[2]   7624 Running  find / -name "*test*" &
```

```
fg 1
```



Terminare Processi

Un processo bloccato, che non risponde o sta occupando troppe risorse, può essere riattivato o terminato utilizzando il comando **kill**.

```
kill -s <SIG> <PID>
```

Bisogna conoscere il PID del processo.

È possibile usare sia il nome che l'identificatore per impostare il segnale.



Segnali

Esistono diversi segnali che possono essere inviati ad un processo per modificarne lo stato.

I segnali vengono gestiti al livello più basso del sistema operativo: il livello kernel.

kill -l Rende l'elenco dei segnali riconosciuti dal kernel



Segnali

SEGNALE	ID	AZIONE	DESCRIZIONE
SIGHUP	1	term	Termina programmi interattivi Riavvio demoni (lettura file conf)
SIGINT	2	term	Blocca un processo (Analogo a CTRL+C)
SIGKILL	9	term	Termina programma immediatamente
SIGTERM	15	term	Termina programma consentendo il rilascio delle risorse
SIGCONT	16	term	Riavvia l'esecuzione di un processo fermato con STOP o TSTP, come bg
SIGSTOP	19	term	Sospende l'esecuzione di un processo (Analogo a CTRL+D)
SIGTSTP	20	term	Ferma l'esecuzione di un processo (Analogo a CTRL+Z)



nohup

I comandi lanciati in background terminano con la chiusura della sessione di lavoro.

Per evitare che tali processi siano chiusi, bisogna lancialli usando il comando **nohup**.

nohup <comando>

Filtra SIGHUP consentendo ad un programma di continuare a girare a sessione chiusa.

È consigliabile reindirigere l'output su file.



Impostare la Priorità di un Processo

Alcuni processi devono avere priorità maggiore di altri.

È possibile impostare la priorità di un processo lanciandolo con il comando **nice**.

nice *-n* <priorità> <comando>

La priorità è un numero intero compreso nell'intervallo [-20; 19].

Valori negativi o bassi hanno maggiore priorità.



Impostare la Priorità di un Processo Attivo

Ogni utente ha la facoltà di modificare la priorità dei propri processi mentre stanno girando.

È possibile cambiare la priorità di un processo con il comando **renice**.

```
renice -n <priorità> -p <PID>
```

Ad un utente standard è consentito solo **aumentare** la priorità dei propri processi (Renderli meno prioritari).



Impostare la Priorità di un Processo Attivo

È possibile cambiare la priorità di tutti i processi di un utente:

```
renice -n <priorità> -u <UID | Utente>
```

È possibile cambiare la priorità di tutti i processi assegnati allo stesso gruppo:

```
renice -n <priorità> -g <GUID | Gruppo>
```



Informazioni & Licenze

LICENZA

Salvo dove altrimenti specificato grafica, immagini e testo della presente opera sono © Simone Giustetti. L'opera può essere ridistribuita per fini non commerciali secondo i termini della licenza:

Creative Commons Attribuzione - Non commerciale - Condividi allo stesso modo 4.0 Internazionale



È possibile richiedere versioni rilasciate sotto diversa licenza scrivendo all'indirizzo: studiosg@giustetti.net

TRADEMARK

- FreeBSD è un trademark di The FreeBSD Foundation.
- Linux è un trademark di Linus Torvalds.
- Macintosh, OS X e Mac OS X sono tutti trademark di Apple Corporation.
- MariaDB è un trademark di MariaDB Corporation Ab.
- MySQL è un trademark di Oracle Corporation.
- UNIX è un trademark di The Open Group.
- Windows e Microsoft SQL Server sono trademark di Microsoft Corporation.
- Alcuni algoritmi crittografici citati nella presente opera potrebbero essere protetti da trademark.

Si prega di segnalare eventuali errori od omissioni al seguente indirizzo: studiosg@giustetti.net

