

# LILO: introduzione



Organizzazione essenziale .....	562
Installazione del meccanismo di caricamento del sistema operativo .....	563
Configurazione .....	564
Installazione del sistema di avvio .....	569
LILO su un disco differente .....	570
Boot prompt .....	572
Riferimenti .....	573

LILO <sup>1</sup> è una procedura per il caricamento di GNU/Linux negli elaboratori con architettura x86. Permette di avviare anche altri sistemi operativi eventualmente residenti nello stesso elaboratore in cui si usa GNU/Linux. In questa sezione si vedono solo alcuni aspetti del suo funzionamento, quelli che dovrebbero bastare nella maggior parte delle situazioni. Nel capitolo [u49](#) viene descritta con maggiore dettaglio la sua configurazione, ma per un approfondimento sul suo funzionamento conviene consultare la documentazione che accompagna questa procedura: la pagina di manuale *lilo(8)*, quanto contenuto nella directory `/usr/share/doc/lilo/` e il *BootPrompt HOWTO*.

## Organizzazione essenziale



La procedura LILO è composta essenzialmente da:

- la directory `/boot/` e dal suo contenuto;
- l'eseguibile `lilo`;
- il file di configurazione `/etc/lilo.conf`.

La directory `/boot/` contiene i file utilizzati per effettuare l'avvio del sistema: sia per avviare GNU/Linux, sia per altri sistemi operativi eventuali. Può contenere anche il file del kernel, o più file di kernel differenti, quando per questo non si usa semplicemente la directory radice. Più precisamente, contiene almeno i file seguenti:

- `boot.b`;
- `map` che viene creato da LILO;
- `boot.n`, dove l'estensione è un numero esadecimale, che viene creato da LILO e contiene il settore di avvio dell'unità rappresentata dal numero stesso (non si tratta necessariamente di un solo file);
- il kernel se non risiede già nella directory radice.

Nella tabella u48.1 sono elencati i codici esadecimali corrispondenti ad alcuni dispositivi per le unità di memorizzazione.

Tabella u48.1. Elenco dei codici esadecimali dei dispositivi di alcune unità di memorizzazione.

Dispositivo	Codice	Dispositivo	Codice	Dispositivo	Codice
/dev/fd0	200 <sub>16</sub>	/dev/fd1	201 <sub>16</sub>		
/dev/hda	300 <sub>16</sub>	/dev/hda1	301 <sub>16</sub>	/dev/hda2	302 <sub>16</sub>
/dev/hdb	340 <sub>16</sub>	/dev/hdb1	341 <sub>16</sub>	/dev/hdb2	342 <sub>16</sub>
/dev/sda	800 <sub>16</sub>	/dev/sda1	801 <sub>16</sub>	/dev/sda2	802 <sub>16</sub>
/dev/sdb	810 <sub>16</sub>	/dev/sdb1	811 <sub>16</sub>	/dev/sdb2	812 <sub>16</sub>

## Installazione del meccanismo di caricamento del sistema operativo

L'installazione del meccanismo di caricamento del sistema operativo avviene modificando il contenuto di uno di questi settori: «

- MBR o *Master boot record*;
- il primo settore di una partizione;
- il primo settore di un dischetto.

Nel primo caso, LILO ha il controllo su tutti i sistemi operativi per il loro caricamento; nel secondo, LILO dipende da un sistema di avviamento di un altro sistema operativo che, a sua volta, passa a LILO il controllo quando ciò viene richiesto; nel terzo caso si utilizza un dischetto in modo da non alterare il sistema di avvio già presente.

L'installazione avviene per mezzo dell'eseguibile `'lilo'`, che a sua volta si basa sulla configurazione stabilita attraverso `'/etc/lilo.conf'`. Ogni volta che si cambia qualcosa all'interno della directory `'/boot/'`, o si modifica, o si sposta il file del kernel, è necessario ripetere l'installazione attraverso l'eseguibile `'lilo'`.

## Configurazione

«

Il file di configurazione utilizzato da LILO per installare il sistema di avvio è `'/etc/lilo.conf'`. Si tratta di una sorta di script contenente solo assegnamenti a variabili. Ne viene descritto il funzionamento in modo sommario partendo da un esempio in cui si ha un solo disco fisso, dove la prima partizione è riservata al Dos e la seconda a un sistema GNU/Linux. L'esempio permette di avviare GNU/Linux e il Dos selezionando una tra le parole `'linux'` o `'dos'` al momento dell'avvio. Il simbolo `'#'` rappresenta l'inizio di un commento che viene ignorato.

```
# Prima parte generale
boot=/dev/hda
prompt
timeout=50

# Caricamento di Linux
image=/boot/vmlinuz
    label=linux
    root=/dev/hda2
    read-only

# Caricamento del Dos
other=/dev/hda1
    label=dos
    table=/dev/hda
```

Segue la descrizione delle direttive che appaiono nell'esempio.

- **'boot=/dev/hda'**

Nella prima parte viene specificato che il settore di avvio deve essere collocato nel primo disco ATA, di conseguenza nell'MBR. Se invece viene indicata una partizione specifica, allora si tratta del primo settore di quella partizione (per esempio: **'boot=/dev/hda2'**). Volendo si potrebbe indicare anche un'unità per i dischetti, in modo da installare tale settore di avvio in quel dischetto (per esempio: **'boot=/dev/fd0'**).

- **'prompt'**

Si tratta di un'opzione (una variabile booleana) la cui presenza fa sì che all'atto del caricamento venga richiesto di inserire il nome del sistema che si desidera avviare (per la precisione, la parola chiave che vi fa riferimento).

- **'timeout=50'**

Dopo 50 decimi di secondo (cinque secondi), senza che sia stato selezionato alcunché, viene avviato il sistema predefinito (in questo caso **'linux'**).

- **'image=/boot/vmlinuz'**

Inizia la definizione di un kernel da avviare: **'/boot/vmlinuz'**.

Si tratta del file che si trova nel file system in funzione nel momento in cui si avvia l'eseguibile 'lilo'. Questo particolare potrebbe sembrare ovvio, ma non è sempre così. Se si vuole preparare un sistema di avvio per un sistema GNU/Linux residente in un'altra partizione (magari un dischetto), si vuole forse fare riferimento a un kernel che si trova lì (nel dischetto). La cosa potrebbe non essere tanto intuitiva e viene descritta più avanti.

- **'label=linux'**

Definisce il nome utilizzato per fare riferimento a questo kernel. Potrebbe essere qualunque cosa, in questo caso il nome 'linux' è utile per ricordare che si tratta dell'avvio di quel sistema operativo.

- **'root=/dev/hda2'**

Indica la partizione da utilizzare come file system principale (*root*).

- **'read-only'**

La presenza di questa opzione fa sì che la partizione specificata venga innestata inizialmente in sola lettura, in modo da permettere al kernel di eseguire un controllo prima di avviare il resto del sistema. Al termine del controllo, la partizione viene reinnestata regolarmente in lettura e scrittura, ma questo per opera della procedura di inizializzazione del sistema.

- **'other=/dev/hda1'**

Inizia la definizione dell'avvio di un altro sistema operativo, per il quale non è LILO a prendersi cura dell'avvio del kernel, ma

un altro settore di avvio. In questo caso il settore di avvio deve trovarsi all'inizio della partizione `‘/dev/hda1’`.

- **‘label=dos’**

Definisce il nome utilizzato per fare riferimento a questo sistema operativo. La parola **‘dos’** è utile per ricordare che si tratta dell'avvio di quel sistema operativo.

- **‘table=/dev/hda’**

Specifica il file di dispositivo che si riferisce all'unità che contiene l'indicazione della tabella delle partizioni. In effetti, questa è contenuta nella parte iniziale del disco fisso, quindi si fa riferimento all'intera unità `‘/dev/hda’`.

Volendo, è possibile avviare lo stesso file system con kernel differenti a seconda delle necessità. In tal caso si possono aggiungere al file `‘/etc/lilo.conf’` altri blocchetti come quello seguente:

```
# Caricamento di Linux con un kernel sperimentale
image=/boot/vmlinuz-prova
    label=prova
    root=/dev/hda2
    read-only
```

Se si vuole la possibilità di utilizzare come file system principale una partizione diversa da quella normale, magari per fare delle prove, o per qualunque altro motivo, si può indicare una voce alternativa come quando si vuole avviare con diversi kernel possibili.

```
# Caricamento di una partizione alternativa in un disco SATA
image=/boot/vmlinuz
    label=extra
    root=/dev/sda3
    read-only
```

Quello che conta è comprendere che il sistema di avvio resta nella directory `‘/boot/’` e senza il disco che la contiene, i file system in `‘/dev/hda2’` o `‘/dev/sda3’` non possono essere innestati. Inoltre, senza `‘/dev/hda’` (in questi esempi), non si avvierebbe alcunché. Per comprendere meglio il problema, si pensi a questo esempio:

- GNU/Linux sia avviato e stia utilizzando la partizione `‘/dev/hda2’` come file system principale;
- la directory `‘/boot/’` sia vuota e sia stata utilizzata per innestare un dischetto corrispondente al dispositivo `‘/dev/fd0’`;
- la directory radice del dischetto corrisponda esattamente a `‘/boot/’`;
- il dischetto contenga i file già visti, necessari per l’avvio (il kernel, `‘boot.b’`, `‘map’`, ecc.);
- il file `‘/etc/lilo.conf’` sia come quello visto sopra, per cui il settore di avvio si deve trovare nell’MBR del primo disco fisso (`‘/dev/hda’`).

In questo modo, se si esegue `‘lilo’`, viene creato un settore di avvio nell’MBR di `‘/dev/hda’` che fa riferimento ai file di avvio (kernel incluso) contenuti nel dischetto. Cioè, senza quel dischetto (proprio quello), il sistema non potrebbe avviarsi. Questo problema viene rivisto più avanti dove viene spiegato come costruire un dischetto contenente sia un settore di avvio, sia il kernel e i file di LILO.

Alle volte è necessario informare il kernel di qualche particolarità dell’hardware installato. In tal caso si utilizza la variabile `‘append’` alla quale si assegna la stringa necessaria. Nell’esempio seguente si

invia la stringa ‘`cdu31a=0x340,0`’ necessaria per poter attivare un vecchio lettore CD-ROM Sony.

```
# Caricamento di Linux con l'attivazione del CD-ROM
image=/boot/vmlinuz
  label=sony
  root=/dev/hda2
  append="cdu31a=0x340,0"
  read-only
```

## Installazione del sistema di avvio

L'eseguibile ‘`lilo`’ permette di installare il sistema di avvio basato sulla procedura LILO. Per farlo, legge il contenuto del file ‘`/etc/lilo.conf`’ o di quello indicato attraverso l'opzione ‘`-C`’.

```
lilo [opzioni]
```

### Tabella u48.6. Alcune opzioni.

Opzione	Descrizione
<code>-C file_di_configurazione</code>	Permette di indicare un file di configurazione differente rispetto al solito ‘ <code>/etc/lilo.conf</code> ’.
<code>-r directory_di_partenza</code>	Permette di definire una pseudo directory radice in modo da poter utilizzare quanto contenuto in un dischetto o in un altro disco innestato da qualche parte.

Segue la descrizione di alcuni esempi.

- # `lilo -C ./mia.conf` [Invio]

Installa il sistema di avvio utilizzando la configurazione del file ‘`mia.conf`’ contenuto nella directory corrente.

- # `lilo -r /mnt/floppy` [Invio]

Utilizza la configurazione del file `/mnt/floppy/etc/lilo.conf`, facendo riferimento (probabilmente) ai file contenuti in `/mnt/floppy/boot/`, utilizzando i file di dispositivo in `/mnt/floppy/dev/`.

## LILO su un disco differente

«

LILO parte dal presupposto che si stia operando sempre all'interno del file system attivo nel momento in cui si avvia l'eseguibile `lilo`. Si potrebbe pensare che per fare in modo di sistemare l'avvio su un altro disco, come un dischetto o un'altra unità rimovibile, si debba agire semplicemente sulla direttiva `boot=dispositivo`; ma questo non basta. Si deve utilizzare l'opzione `-r` per fare riferimento a una pseudo directory radice, a partire dalla quale LILO deve trovare tutto quello che gli serve, compreso il file di configurazione.

Di seguito viene mostrato l'esempio della preparazione di un dischetto contenente il kernel avviato da LILO, in modo completamente indipendente dal file system attivo nel momento in cui lo si realizza, con una configurazione simile a quella mostrata in precedenza, nella sezione [i48.2.1](#).

1. All'interno di un dischetto inizializzato e contenente un file system Second-extended (Ext2) si riproduce tutto quello che serve a LILO per definire il sistema di avvio. Si tratta della directory `boot/` contenente gli stessi file della stessa directory appartenente al file system generale, insieme al kernel; della directory `etc/` con il file `lilo.conf`; della directory `dev/` con i file di dispositivo corrispondenti alle unità di memorizzazione

cui si fa riferimento. Si suppone di avere innestato il dischetto utilizzando la directory `‘/mnt/floppy/’` come punto di innesto.

```
# fdformat /dev/fd0u1440 [Invio]

# mkfs.ext2 /dev/fd0 [Invio]

# mount -t ext2 /dev/fd0 /mnt/floppy [Invio]

# cp -dpR /boot /mnt/floppy [Invio]

# mkdir /mnt/floppy/etc [Invio]

# cp /etc/lilo.conf /mnt/floppy/etc/lilo.conf [Invio]

# mkdir /mnt/floppy/dev [Invio]

# cd /mnt/floppy/dev/ [Invio]

# /dev/MAKEDEV fd0 fd1 hda hdb hdc hdd sda sdb sdc sdd [Invio]
```

2. Il file `‘/mnt/floppy/etc/lilo.conf’` viene modificato in modo da fare riferimento al dispositivo `‘/dev/fd0’`.

```
boot=/dev/fd0
```

3. Si utilizza l'eseguibile `‘lilo’` con l'opzione `‘-r’` in modo da fargli usare i file nel dischetto e non quelli contenuti nel file system principale.

```
# lilo -r /mnt/floppy [Invio]
```

Il problema può presentarsi anche in modo inverso, quando si avvia il sistema attraverso dischetti di emergenza e si vuole sistemare l'avvio di GNU/Linux attraverso il disco fisso. La partizione principale del disco fisso potrebbe essere innestata nel sistema di emergenza, per esempio in corrispondenza della directory `‘/mnt/’`, mentre per il resto non dovrebbe essere necessario preoccuparsi d'altro, a parte la

versione di LILO presente nel dischetto, che deve essere compatibile con i file di avvio del disco fisso.

```
# lilo -r /mnt [Invio]
```

## Boot prompt

«

Subito dopo la prima fase dell'avvio del sistema, quella gestita da LILO, prima dell'avvio vero e proprio del kernel, in presenza di determinate condizioni viene visualizzato un invito particolare a inserire delle opzioni: il *boot prompt*. Questo appare:

- se è stata indicata l'istruzione '**prompt**' nel file '/etc/lilo.conf';
- se viene premuto il tasto [*Maiuscole*], oppure [*Ctrl*], oppure [*Alt*];
- se il tasto [*Fissamaiuscole*] oppure [*BlocScorr*] risultano inseriti.

Il *boot prompt*, ovvero l'invito dell'avvio, ha l'aspetto seguente:

```
boot:
```

Normalmente si utilizza la riga di comando di avvio per indicare il nome di una configurazione particolare. In altri casi è il mezzo per specificare un'opzione che per qualche motivo non è attiva automaticamente e si vuole che LILO la passi al kernel.

La digitazione all'interno di questa riga di comando è abbastanza intuitiva: per cancellare si possono usare i tasti [*Backspace*], [*Canc*] e le combinazioni [*Ctrl u*] e [*Ctrl x*]. Eventualmente, si può ottenere un elenco delle configurazioni, riferite a diverse voci del file '/etc/'

`lilo.conf`, attraverso la pressione del tasto [*Tab*]. Si conferma con il tasto [*Invio*]. Il vero problema è la tastiera: si deve considerare che la disposizione dei tasti è quella statunitense.

La sintassi di quanto si può inserire attraverso la riga di comando è la seguente:

```
[ configurazione [ opzione... ] ]
```

Se si preme semplicemente [*Invio*] viene avviata la configurazione predefinita, altrimenti è obbligatorio l'inserimento del nome di questa, seguita eventualmente da altre opzioni.

I vari argomenti inseriti attraverso la riga di comando (il nome della configurazione e le altre opzioni eventuali) sono separati tra loro attraverso uno spazio. Per questo, un argomento non può contenere spazi.

Nella sezione [6.8](#) vengono descritti alcuni tipi di parametri che possono essere inseriti in una riga di comando di avvio. Per una descrizione più ampia conviene consultare il *BootPrompt HOWTO*.

## Riferimenti

- Werner Almesberger, *LILLO Generic boot loader for Linux - User's guide*

<sup>1</sup> **LILLO** licenza speciale senza vincoli particolari

