LILO mini-HOWTO

Miroslav "Misko" Skoric, skoric at eunet dot yu v3.31, 13 gennaio 2008

LILO è il Linux Loader più utilizzato per la versione di Linux che gira su processori x86; poiché non apprezzo le maiuscole lo chiamerò Lilo invece di LILO. Questo documento descrive alcune installazioni tipiche di Lilo: il suo ruolo è quello di integrare la guida dell'utente di Lilo. Penso che gli esempi siano istruttivi anche se la vostra configurazione non somiglia alla mia: spero che questo documento vi risparmi problemi. Dato che la documentazione di Lilo è molto ben fatta, chi è interessato ai dettagli può fare riferimento a /usr/doc/lilo* (Cameron Spitzer e Alessandro Rubini, persone che avremmo chiamato in passato gentiluomini, hanno scritto le versioni precedenti di questo documento). Traduzione a cura di Elisabetta Galli (lab at kkk.it). Revisione a cura di Gualtiero Testa (gualtiero.testa at gmail.com). Questa versione del Lilo mini-HOWTO è basata sul lavoro di Cameron Spitzer (`cls@truffula.sj.ca.us`) e Alessandro Rubini (`rubini@linux.it`). Ci sono inoltre contributi di Tony Harris (`tony@nmr.mgh.harvard.edu`) e Marc Tanguy (`mtanguy@ens.uvsq.fr`). Sono stati usati materiali degli autori menzionati, **senza modifiche**, e aggiunti alcuni riferimenti relativi alla configurazione di LILO per l'uso con Windows NT e Windows 2000. Si possono trovare informazioni più dettagliate sull'attivazione di Windows NT/2000 dal menu di LILO nel meraviglioso *Linux+WindowsNT* http://tldp.org/HOWTO/Linux+WinNT.html mini-HOWTO.

1 Introduzione

Sebbene la documentazione presente nei sorgenti di Lilo (quella installata in `/usr/doc/lilo`-versione) sia molto esauriente, molti utenti Linux trovano difficoltà nel creare il proprio file `/etc/lilo.conf`. Questo documento vuole supportarli, fornendo le informazioni di base e mostrando cinque esempi di installazione:

- Il primo esempio è la classica installazione "Linux e altro".
- Il successivo mostra come installare Lilo su un disco rigido connesso come `/dev/hdc`, che verrà avviato come `/dev/hda`. Di solito serve quando si vuole installare Linux su una nuova unità da un sistema già funzionante. Spiega anche come avviare da dischi SCSI se il BIOS è abbastanza moderno.
- Il terzo esempio mostra come avviare un sistema Linux la cui partizione root non è accessibile dal BIOS.
- L'esempio successivo si usa per accedere a dischi molto grossi, a cui né il BIOS né il DOS possono accedere con facilità (questo esempio è in qualche modo obsoleto).
- L'ultimo esempio mostra come ripristinare un disco danneggiato, se il danno deriva dall'installazione di un altro sistema operativo.

Gli ultimi tre esempi sono di Cameron, `cls@truffula.sj.ca.us`, che ha scritto il documento originale. Alessandro `rubini@linux.it` usa solo Linux, quindi non ha potuto verificarli o aggiornarli per suo conto. Inutile dire che ogni commento è benvenuto.

2 Informazioni di base e installazione standard

Quando Lilo avvia il sistema usa le chiamate del BIOS per caricare il kernel Linux dal disco (disco IDE, floppy o altro). Perciò il kernel deve risiedere in un luogo a cui il BIOS possa accedere.

All'avvio Lilo non è in grado di leggere i dati del file system, e ogni percorso inserito in `/etc/lilo.conf` viene risolto durante l'installazione (quando si invoca **/sbin/lilo**). L'installazione è il momento in cui il programma costruisce le tabelle che elencano quali settori sono usati dai file usati per il caricamento del sistema operativo. La conseguenza è che tutti questi file devono risiedere in una partizione accessibile dal BIOS (questi file di solito risiedono nella directory `/boot`, il che significa che il BIOS avrà bisogno di accedere solo alla partizione root del proprio sistema Linux).

Un'altra conseguenza dell'appoggiarsi al BIOS è che bisogna reinstallare il caricatore (cioè bisogna invocare nuovamente **/sbin/lilo**), ogni volta che si modifica la configurazione di Lilo. Ogni volta che si ricompila il kernel e si sovrascrive la vecchia immagine, bisogna reinstallare Lilo.

2.1 Dove dovrebbe essere installato Lilo?

La direttiva `boot=` contenuta in `/etc/lilo.conf` dice a Lilo dove installare il suo gestore di avvio primario. In generale si può specificare sia il Master Boot Record (`/dev/hda`) che la partizione root della propria installazione Linux (di solito `/dev/hda1` o `/dev/hda2`).

Se il proprio disco fisso contiene un altro sistema operativo, conviene installare Lilo sulla partizione root invece che sul Master Boot Record. In questo caso occorre marcare la partizione come "avviabile" usando il comando "a" di **fdisk** o il comando "b" di **cfdisk**. Se non si sovrascrive il Master Boot Sector sarà più facile disinstallare Linux e Lilo, se ce ne fosse bisogno.

Naturalmente c'è sempre un modo per eludere alcune regole come quelle precedenti. Si potrebbe installare Lilo nell'MBR anche se c'è già un altro sistema operativo presente. Per esempio, se si è installato Windows NT 4.0 come primo sistema operativo sulla macchina, il suo gestore di avvio si sarà installato nell'MBR per poter avviare NT senza problemi. Dopo aver installato Linux, e aver scelto di installare Lilo nell'MBR, Lilo avrà riscritto il gestore di avvio di NT. All'avvio successivo non sarà possibile avviare NT, ma non è un problema. Basterà modificare il file `/etc/lilo.conf` e aggiungere una nuova voce per NT. Al successivo riavvio del sistema ci sarà la nuova voce per NT nel menu di Lilo. La stessa cosa succede installando Windows 2000 invece di Windows NT.

2.2 Come si dovrebbero configurare i dischi IDE?

Personalmente nel BIOS preferisco non usare le impostazioni LBA o LARGE (ma io uso solo Linux): si tratta di orribili trucchetti introdotti a forza per coprire alcune deficienze progettuali nel mondo dei PC. Questo richiede che il kernel risieda nei primi 1024 cilindri del disco, ma non è un problema a patto che i dischi fissi siano partizionati e la partizione root rimanga piccola (il che andrebbe fatto comunque).

Se il proprio disco fisso contiene un altro sistema operativo, non si potranno modificare le impostazioni del BIOS perché altrimenti il vecchio sistema non funzionerà più. Tutte le distribuzioni recenti di Lilo sono in grado di gestire dischi impostati con LBA e LARGE.

Si noti che la parola chiave `linear` nel file `/etc/lilo.conf` può essere d'aiuto nell'affrontare problemi di geometria. La parola chiave ordina a Lilo di usare indirizzi di settore lineari al posto di tuple settore/testa/cilindro. La conversione ad indirizzi 3D è posticipata alla fase di esecuzione, rendendo l'impostazione maggiormente immune ai problemi di geometria.

Se si ha più di un disco rigido, e alcuni di essi sono usati solo da Linux senza far parte del processo di avvio, si può dire al BIOS che non sono installati. Il sistema si avvierà più velocemente e Linux riconoscerà automaticamente e velocemente tutti i dischi. Mi capita spesso di cambiare i dischi dei miei sistemi, senza mai toccare la configurazione del BIOS.

2.3 Come si può interagire durante l'avvio?

Quando appare il prompt di Lilo, premendo il tasto <Tab> si ottiene la lista di tutte le scelte possibili. Se Lilo non è configurato per interagire con l'utente, si può premere (e tenere premuto) il tasto <Alt> o il tasto <Shift> prima che appaia il messaggio "LILO".

Se si sceglie di avviare un kernel Linux è possibile passare dei parametri da linea di comando dopo il nome del sistema scelto. Il kernel accetta molti parametri da linea di comando, tutti elencati nel "BootPrompt-HOWTO" di Paul Gortmaker, documento che non verrà replicato in questa sede. Alcuni di questi parametri, comunque, sono particolarmente importanti e meritano di essere descritti:

- `root=`: si può dire al kernel Linux di montare come root una partizione diversa da quella che appare nel file `lilo.conf`. Per esempio, nel mio sistema c'è una piccola partizione che ospita una installazione minimale di Linux; grazie a questa sono stato in grado di avviare il sistema dopo aver distrutto per sbaglio la partizione root.
- `init=`: la versione del kernel Linux 1.3.43 e successive possono eseguire un altro comando al posto di /**sbin**/**init**, come specificato dalla linea di comando. Se si verificano problemi all'avvio, si può accedere al nudo sistema specificando `init=/bin/sh` (arrivati al prompt della shell probabilmente si dovranno montare i propri dischi: si provi con "**mount -w -n -o remount /; mount -a**", e ci si ricordi di usare "**umount -a**" prima di spegnere il computer).
- Un numero: specificando un numero dalla linea di comando del kernel si ordina a *init* di avviare uno specifico runlevel (quello predefinito di solito è il 3 o il 2, in base alla distribuzione usata). Per indagare ulteriormente si faccia riferimento alla documentazione di *init*, al file `/etc/inittab` e a `/etc/rc*.d`.

2.4 Come si può disinstallare Lilo?

Quando Lilo sovrascrive un settore di avvio, ne salva una copia di backup in `/boot/boot.`*xxyy*, dove *xxyy* sono i numeri maggiore e minore del dispositivo, in esadecimale. Si possono vedere i due numeri del proprio disco o partizione usando "**ls -l /dev//*device***". Per esempio, il primo settore di `/dev/hda` (numero maggiore 3, numero minore 0) verrà salvato in `/boot/boot.0300`; installando Lilo su `/dev/fd0` verrà creato `/boot/boot.0200`, installandolo su `/dev/sdb3` (numero maggiore 8, numero minore 19) verrà creato `/boot/boot.0813`. Si noti che Lilo non creerà il file se ce n'è già uno, quindi non occorre preoccuparsi della copia di backup quando si reinstalla Lilo (per esempio dopo aver ricompilato il kernel). Le copie di backup presenti in `/boot/` sono sempre l'istantanea della situazione precedente all'installazione di Lilo.

Se mai fosse necessario disinstallare Lilo (per esempio nella malaugurata ipotesi si decida di disinstallare Linux) basterà ripristinare il settore di avvio originale. Se Lilo è installato su `/dev/hda`, basterà eseguire "**dd if=/boot/boot.0300 of=/dev/hda bs=446 count=1**" (personalmente uso "**cat /boot/boot.0300 > /dev/hda**", ma non è sicuro, in quanto ripristina anche la tabella delle partizioni originale che potrebbe essere stata modificata nel frattempo). È molto più facile usare questo comando che chiamare "**fdisk /mbr**" da una shell DOS: permette di rimuovere in modo impeccabile Linux da un disco senza dover mai avviare nient'altro che Linux. Dopo aver rimosso Lilo occorre ricordarsi di far girare il comando **fdisk** di Linux per distruggere qualunque partizione Linux (il comando **fdisk** del DOS non è in grado di cancellare partizioni non-dos).

Se Lilo è stato installato sulla partizione root (per esempio su `/dev/hda2`), non occorrerà fare niente di particolare per disinstallare Lilo. Basterà usare il programma **fdisk** di Linux per rimuovere le partizioni Linux dalla tabella delle partizioni. Bisognerà anche marcare la partizione DOS come avviabile.

2.5 Come si crea un ramdisk?

Avvertimento: se la prossima sezione fosse difficile da leggere, si veda la pagina web `http://surfer.nmr.mgh.harvard.edu/partition/ramdisk.html`*, dove è disponibile l'originale di questo contributo.*

Di Tony Harris

16 Ottobre 2000

ramdisk eenie-weenie HOWTO

Se il proprio file system root si trova su un dispositivo senza che il kernel abbia il relativo driver incorporato, si dovrà usare `lilo` per caricare quel driver come modulo molto presto nel ciclo di avvio. Ci sono solo due facili passi da seguire:

- creare un'immagine per ramdisk con /**mkinitrd**
- modificare `lilo.conf` perché punti a questa immagine

Come prima cosa, usare **cd** per entrare nella directory `/boot`:

```
System.map                 chain.b                 module-info-2.2.16-3ext3
System.map-2.2.16-3        initrd-2.2.16-3.img     vmlinux-2.2.16-3
System.map-2.2.16-3ext3    vmlinux-2.2.16-3ext3
vmlinuz                    kernel.h
boot.b                     map                     vmlinuz-2.2.16-3
bz.2.2.15.juke.Image       module-info             vmlinuz-2.2.16-3ext3
bzImage-2.2.14             module-info-2.2.16-3
```

Come si può vedere io ho un kernel 2.2.16-3 al quale ho aggiunto un secondo kernel con supporto per ext3 (`vmlinuz-2.2.16-3ext3`). C'è già un'immagine per ramdisk per il primo kernel (`initrd-2.2.16-3.img`)

Per creare una nuova immagine per il secondo kernel usare quanto segue (la parte da digitare è in grassetto):

boot# **mkinitrd initrd-2.2-16-3ext3.img 2.2.16-3ext3**

`mkinitrd` è uno script per la shell che cerca i moduli necessari per il kernel, poi crea un file system di tipo ext2 che li contiene. Guardando all'interno dell'immagine si vedrà che è proprio così:

boot# **cat initrd-2.2.16-3ext3.img | gunzip > /tmp/myimage**

boot# **file /tmp/myimage**

/tmp/myimage: Linux/i386 ext2 filesystem/

Non è necessario guardare nell'immagine. I soli passi necessari sono creare l'immagine e modificare `lilo.conf`. Comunque si fornisce una discussione sull'immagine per ramdisk per scopi educativi.

Per guardare all'interno dell'immagine, è necessario montarla come se fosse un file system:

boot# **mount /tmp/myimage /mnt/tmp -t ext2 -o loop=/dev/loop3**

boot# **ls /mnt/tmp**

`bin dev etc lib linuxrc`

boot# **find /mnt/tmp**

mnt/tmp/

mnt/tmp/lib/

mnt/tmp/lib/aic7xxx.o/

mnt/tmp/bin/

mnt/tmp/bin/sh/

mnt/tmp/bin/insmod/

mnt/tmp/etc/

mnt/tmp/dev/

mnt/tmp/dev/console/

mnt/tmp/dev/null/

mnt/tmp/dev/ram/

mnt/tmp/dev/systty/

mnt/tmp/dev/tty1/

mnt/tmp/dev/tty2/

mnt/tmp/dev/tty3/

mnt/tmp/dev/tty4/

mnt/tmp/linuxrc/

La parte più importante di questa immagine per ramdisk è `aic7xxx.o`, che è il modulo per il mio disco scsi.

Per finire l'ultimo passo, la modifica di `/etc/lilo.conf`:

Ecco la voce nel mio `lilo.conf` che corrisponde al kernel e all'immagine appena creata:

`image=`boot/vmlinuz-2.2.16-3ext3/

`label=linux.ext3`

`initrd=`boot/initrd-2.2.16-3ext3.img/

`read-only`

`root=`dev/hdb3/

Ecco fatto. Eseguire /**lilo** come root e riavviare.

Se si verificano problemi, controllare il kernel `HOWTO`. Ci sono un paio di cose di cui assicurarsi: è necessario avere i moduli per il kernel compilati, e indicati in `/etc/conf.modules`.

3 La configurazione semplice

La maggior parte delle installazioni di Lilo utilizza un file di configurazione come il seguente:

```
boot = /dev/hda   # oppure la partizione root
delay = 10        # attesa, in decimi di secondo (permette di interagire)
vga = 0           # opzionale. Si usi "vga=1" per ottenere 80x50
#linear           # si provi ad usare "linear" se si verificano problemi di geometria.

image = /boot/vmlinux  # il proprio file zImage
  root = /dev/hda1     # la partizione root
  label = Linux        # o un altro nome stravagante
  read-only            # root verrà montata in sola lettura
```

```
other = /dev/hda4   # la partizione dos, se esiste
  table = /dev/hda  # la tabella delle partizioni attuale
  label = dos       # o un altro nome normale
```

Si possono specificare diverse sezioni “image” e “other”, se serve. È abbastanza comune avere nel proprio *lilo.conf* diverse immagini del kernel, almeno se ci si vuole mantenere aggiornati alle versioni di sviluppo del sistema.

3.1 Come affrontare grossi kernel

Se la compilazione di una “zImage” genera un’immagine più grande di mezzo megabyte (come capita spesso con i nuovi kernel 2.1), bisognerebbe invece creare una “grande zImage”, usando “`make bzImage`”. Per avviare una grande immagine del kernel non serve niente di speciale, se non avere la versione 18 di Lilo, o una successiva. Se la propria installazione è più vecchia, occorrerà aggiornare il pacchetto Lilo.

3.2 Come avviare Windows NT dal menu ’LILO boot:’

Ecco alcune routine da seguire per avere nel menu di Lilo le voci sia per Linux che per NT:

- Prima di tutto, si suggerisce di installare una nuova copia di Windows NT 4.0 sul disco rigido. Si suppone che sia già stato preparato un backup dei dati importanti, così da non avere problemi con l’installazione di NT. Nel corso dell’installazione di NT, la configurazione non chiederà dove posizionare il gestore di avvio: lo metterà nell’MBR (Master Boot Record) del proprio disco rigido. C’è sempre la possibilità che il contenuto precedente dell’MBR rimanga al suo interno (specialmente una precedente installazione di Lilo), quindi si suggerisce (prima di installare NT) di avviare il computer con un floppy DOS che contenga una versione per DOS di FDISK. Al prompt a:\ digitare il comando: fdisk /mbr e riavviare di nuovo il computer (senza quel floppy).
- Dopo aver installato con successo NT, si noterà che usa l’intero disco rigido o una specifica partizione del disco rigido (a seconda di quello che si è scelto durante il processo di configurazione). Quindi, è consigliabile ’ridurre’ la partizione dove risiede NT per creare dello spazio libero sul disco. In questo spazio libero verrà installato Linux. Quando NT è configurato e funzionante, si dovrà avviare il computer usando un floppy contenente l’utilità Partition Magic di Power Quest. Si tratta di uno strumento grafico in grado di vedere tutte le partizioni di tutti i dischi rigidi presenti. La cosa migliore è che si possono apportare alcuni cambiamenti alle partizioni senza distruggere i dati presenti. Una delle possibilità è rendere più piccole le partizioni presenti, per ottenere un po’ di spazio libero sul disco da usare per altri scopi. Anche se si consiglia di fare un backup prima di qualunque cambiamento alle partizioni, io ho l’abitudine di ’ridurre’ la partizione di NT prima di installare qualunque altra cosa (così, se necessario, una monotona reinstallazione non sarà un problema). Bene, Partition Magic (o altre simili utilità con cui si abbia familiarità) ridurrà la partizione di NT (sia di tipo NTFS che FAT) ad una dimensione più piccola, e la posizionerà o all’inizio o alla fine della dimensione precedente. Il che significa che si può scegliere di posizionarla all’inizio o alla fine del proprio disco (di solito scelgo di posizionarla all’inizio, rendendo la parte finale del disco ’spazio libero’). Al termine della procedura di ’riduzione’, si potrà riavviare NT per verificare la nuova situazione: per farlo si possono usare Windows Explorer o Disk Administrator.
- Fin qui tutto bene. Il prossimo passo è installare Linux. Se si ha familiarità con la distribuzione RedHat (spero che altre distribuzioni si comportino allo stesso modo o quasi), si può iniziare inserendo il CD di installazione nel lettore e riavviando il computer. Quando arriva il momento di scegliere il tipo

di installazione (Workstation Gnome o KDE, Custom, ecc.) si può scegliere qualunque cosa pianificata in precedenza, ma si suggerisce un'installazione prima di tutto come Workstation. È una buona scelta perché la configurazione di Linux troverà automaticamente lo spazio libero sul (primo) disco rigido, preparerà tutte le partizioni necessarie per Linux, le formatterà correttamente, impostando per la maggior parte le opzioni predefinite, così la configurazione non sarà molto dolorosa (più tardi, volendo, si potranno aggiungere componenti mancanti o reinstallare Linux come Custom sulla partizione Linux esistente). Lilo dovrebbe essere installato nell'MBR.

- Al termine dell'installazione di Linux si dovrà riavviare il computer, e si vedrà Lilo con una sola voce per Linux da avviare (o più di una voce se il proprio hardware è multi processore). Niente panico, Windows NT è sempre dove è stato installato prima di Linux. Sarà necessario diventare familiari con Linux il più presto possibile, per essere in grado di trovare e modificare il proprio nuovo file /etc/lilo.conf. Quando si aprirà questo file per la prima volta, si vedrà solo una (o più di una) voce per Linux: si dovrà conoscere la posizione esatta (leggi: una partizione) in cui Windows NT è stato installato, per poter aggiungere una voce adatta al file /etc/lilo.conf. Dopo averlo fatto, si dovrà far ripartire Lilo e, al prossimo riavvio, ci saranno le voci 'linux' e 'nt' nel menu di Lilo.

3.3 Come avviare Windows 2000 dal menu 'LILO boot:'

Si può usare la stessa procedura descritta prima. Si suggerisce di leggere il *Linux+WindowsNT* `http://tldp.org/HOWTO/Linux+WinNT.html` mini-HOWTO che parla anche di come avviare Windows 2000 installato nella stessa parte di disco dove *prima* era installato Windows NT. Si troveranno anche molti utili dettagli riguardanti varie combinazioni Linux+WinNT/2000/98.

4 Installare su `hdc` per avviarlo come `hda` usando `bios=`

Lilo permette di mappare l'immagine del kernel da un disco e di istruire il BIOS perché la recuperi da un altro disco. Per esempio, a me capita spesso di installare Linux su un disco collegato come `hdc` (disco principale del controller secondario), e di avviarlo come sistema autonomo sul controller IDE primario di un altro computer. Ho fatto una copia del floppy di installazione in una piccola partizione, per poter installare su `hdc` eseguendo *chroot* su una console virtuale, continuando ad usare il sistema per fare altro.

Il file *lilo.conf* usato per installare Lilo ha questo aspetto:

```
# Questo file deve essere usato da un sistema che giri su /dev/hdc
boot = /dev/hdc   # sovrascrive l'MBR di hdc
disk = /dev/hdc   # specifica come hdc verrà visto:
   bios = 0x80    #  il bios lo vedrà come primo disco
delay = 0
vga = 0

image = /boot/vmlinux  # l'immagine risiede su /dev/hdc1
  root = /dev/hda1     # ma all'avvio sarà su hda1
  label = Linux
  read-only
```

Questo file di configurazione deve essere usato da un Lilo funzionante **su /dev/hdc1**. Le mappe di Lilo che vengono scritte nel settore di avvio (`/dev/hdc`) devono far riferimento ai file residenti in `/boot` (attualmente installati su hdc); questi file verranno letti da hda quando il disco verrà avviato come sistema autonomo.

Io chiamo questo file di configurazione `/mnt/etc/lilo.conf.hdc` (`/mnt` è dove viene montato hdc durante l'installazione). L'installazione di Lilo avviene invocando "`cd /mnt; chroot . sbin/lilo -C /etc/lilo.conf.hdc`". Si faccia riferimento alla pagina di manuale di *chroot* se questo comando sembra una magia.

La direttiva "`bios=`" di `lilo.conf` si usa per dire a Lilo che cosa il BIOS pensa dei propri dispositivi. Le chiamate al BIOS identificano i floppy e i dischi rigidi con un numero: 0x00 e 0x01 selezionano i dischi floppy, 0x80 e i numeri successivi selezionano i dischi rigidi (i vecchi BIOS possono accedere solo a due dischi). Il significato di "`bios = 0x80`" usato nell'esempio precedente significa quindi "usa 0x80 nelle chiamate al BIOS per `/dev/hdc`".

Questa direttiva di Lilo può essere utile in altre situazioni, per esempio quando il proprio BIOS è in grado di avviare il sistema da dischi SCSI piuttosto che da dischi IDE. Quando sono presenti sia dispositivi IDE che SCSI, Lilo non può sapere se 0x80 si riferirà all'uno o all'altro, poiché l'utente può scegliere quale usare tramite i menu di configurazione del BIOS, e non è possibile accedere al BIOS mentre Linux è in funzione.

Lilo, come comportamento predefinito, assume che i dischi IDE vengano mappati per primi dal BIOS. Si può annullare questo comportamento usando nel proprio `/etc/lilo.conf` istruzioni come queste:

```
disk = /dev/sda
  bios = 0x80
```

5 Usare Lilo quando il BIOS non può vedere la partizione root

Ho due dischi IDE e un disco SCSI. Il disco SCSI non può essere visto dal BIOS. Lilo, il Linux Loader, usa le chiamate al BIOS e può vedere solo i dischi che il BIOS vede. Il mio stupido BIOS AMI può avviare il sistema solo da A: o da C:. Il mio file system root risiede su una partizione del disco SCSI.

La soluzione consiste nel memorizzare il kernel, la mappa dei file e il chain loader in una partizione Linux nel primo disco IDE. Si noti che non serve tenere il proprio kernel nella partizione root.

La seconda partizione del primo disco IDE (`/dev/hda2`, la partizione Linux usata per avviare il sistema) è montata su `/u2`. Ecco il file `/etc/lilo.conf` che ho usato.

```
#  Installa LILO sul Master Boot Record del primo disco IDE
#
boot = /dev/hda
#  /sbin/lilo (l'installatore) copia il boot record di Lilo
#  dal file seguente all'MBR
install = /u2/etc/lilo/boot.b
#
#  Ho scritto un menu di avvio prolisso.  Lilo può trovarlo qui.
message = /u2/etc/lilo/message
#  L'installatore creerà il file seguente. Dirà
#  al gestore di avvio dove sono i blocchi del kernel.
map = /u2/etc/lilo/map
compact
prompt
#  Aspetta 10 secondi, poi avvia il kernel 1.2.1 predefinito.
timeout = 100
#  Il kernel è memorizzato dove il BIOS può vederlo facendo:
#      cp -p /usr/src/linux/arch/i386/boot/zImage /u2/z1.2.1
image = /u2/z1.2.1
```

```
        label = 1.2.1
#  Lilo dice al kernel di montare la prima partizione SCSI
#  come root. Non è necessario che il BIOS sia in grado di vederlo.
        root = /dev/sda1
#  Questa partizione verrà controllata e rimontata da /etc/rc.d/rc.S
        read-only
#  Ho tenuto da parte un vecchio kernel Slackware, nel caso io compili un
#  kernel non funzionante. Una volta, in effetti, mi è servito.
image = /u2/z1.0.9
        label = 1.0.9
        root = /dev/sda1
        read-only
#  La mia partizione DR-DOS 6.
other = /dev/hda1
        loader=/u2/etc/lilo/chain.b
        label = dos
        alias = m
```

6 Come scoprire il numero assegnato dal BIOS ai dischi SCSI

Contributo di Marc Tanguy (mtanguy@ens.uvsq.fr), 27 settembre 2001

6.1 La teoria

Per la verità ci sono due modi per saperlo:

Chi possiede una scheda scsi adaptec (2940u2, 29160, 39160) può semplicemente usare la modalità 'diagnose' (si raccomanda di usare un BIOS v3.10.0), che deve essere attivata dal menu del BIOS della scheda scsi. Basterà quindi aspettare di vedere qualcosa di questo genere:

...	ID	LUN	Produttore	Prodotto	Rev	Dim.	Sync	Bus	**HD#**
...	0	0	QUANTUM	ATLAS10K2	DDD6	17GB	160	16	**80h**
...	1	0	QUANTUM	ATLAS10K2	DDD6	17GB	160	16	**81h**
...	2	0	IBM	DDRS	DC1B	4GB	80	16	**82h**
...	3	0	IBM	DNES	SAH0	9GB	80	16	**83h**

Chi non possiede una scheda adaptec dovrà scoprire qual è il disco 'avviabile' (di solito quello con ID 0, ma non sempre: può essere definito nel BIOS della scheda scsi) nel quale si potrà trovare ed avviare LILO: è il primo disco, quindi avrà il numero 0x80. Dopo di che è molto facile, il BIOS segue gli ID.

Per esempio:

```
ID 0 -> avvio  -> 0x80
ID 1 -> vuoto
ID 2 -> disco  -> 0x81
ID 3 -> disco  -> 0x82
```

o

```
ID 0 -> disco  -> 0x81
```

```
ID 1 -> vuoto
ID 2 -> disco  -> 0x82
ID 3 -> avvio  -> 0x80
ID 4 -> disco  -> 0x83
```

Questa parte non si preoccupa affatto di cosa c'è installato sui dischi scsi. Si noti però che si avranno problemi se si usa un ID più alto di quello dell'adattatore SCSI. Dunque si dovrà sempre cercare di impostare l'ID dell'adattatore SCSI dopo l'ID dei dispositivi SCSI.

6.2 Come scambiare l'avvio di linux e NT?

Supponiamo che NT debba essere il primo disco avviabile, cioè con il numero 0x80, ma nella posizione 0x80 abbiamo già un disco con LILO e un file system ext2 pieno, mentre il disco con NT ha il numero 0x83. Come si possono 'scambiare' linux e NT? È molto facile: basta dire al BIOS che il disco con NT ora è il numero 0x80, mentre quello con Linux è il 0x83.

```
other=/dev/sdd1
      label=nt
      map-drive = 0x83
      to = 0x80
      map-drive = 0x80
      to = 0x83
```

Questo cambiamento produrrà un avviso:

```
  Warning: BIOS drive 0x8? may not be accessible
```

ma sapendo ciò che si sta facendo dovrebbe funzionare senza problemi.

Io l'ho usato con una configurazione composta da una Red Hat Linux 7.1 e Windows 2000 Pro:

```
Nome        Flag      Tipo part.  Tipo FS          [Etichetta]      Dimensione (MB)

Disk Drive: /dev/sda - 0x80
sda1        Boot        Primary   Linux ext2       [/boot]             24.68
sda2                    Primary   Linux Swap                          139.83
sda3                    Primary   Linux ext2       [/usr]            3150.29
sda4                    Primary   Linux ext2       [/home]          15044.04

Disk Drive: /dev/sdb - 0x81
sdb1                    Primary   Linux Swap                          139.83
sdb2                    Primary   Linux ext2       [/]               3150.29
sdb3                    Primary   Linux ext2       [/opt]            1052.84
sdb4                    Primary   Linux ext2       [/public]        14015.88

Disk Drive: /dev/sdc - 0x82
sdc1                    Primary   Linux ext2       [/var]            1052.84
sdc2                    Primary   Linux ext2       [/tmp]             106.93
sdc3                    Primary   Linux ext2       [/cache]          1052.84
sdc4                    Primary   Linux ext2       [/chroot]         2352.44
```

```
Disk Drive: /dev/sdd - 0x83
sdd1        Boot        Primary   NTFS             [WINDOWS_2000]   9162.97
```

Il mio file /etc/lilo.conf:

```
boot=/dev/sda
map=/boot/map
install=/boot/boot.b
prompt
default=Linux
read-only
compact
image=/boot/vmlinuz
  label=Linux
  root=/dev/sdb2
other=/dev/sdd1
  label=Windows
  map-drive = 0x83
  to = 0x80
  map-drive = 0x80
  to = 0x83
```

6.3 Varie

Ho inserito un nuovo disco scsi, e ora Lilo si rifiuta di partire. Cosa sta succedendo?

Quando si inserisce un disco bisogna fare attenzione agli ID. Quando si aggiunge un disco tra due dischi già inseriti i numeri del BIOS cambiano:

```
         Prima            ---->             Dopo
scsi id -       - BIOS id        scsi id -          - BIOS id
ID 0    - disk  - 0x80           ID 0    - disk     - 0x80
ID 1    - empty                  ID 1    - new disk - 0x81
ID 2    - disk  - 0x81           ID 2    - disk     - 0x82 !!
```

Quando gli ID del BIOS cambiano, bisogna valutarli nuovamente.

7 Accedere a grossi dischi che il BIOS non vede

Avviso: 1 GB è grosso? Forse una volta...

Il sistema che ho in ufficio ha un disco IDE da 1 GB. Il BIOS può accedere solo ai primi 504 MB del disco (con MB intendo 2**10 byte, non 10**6 byte), perciò ho installato il sistema MS-DOS in una partizione da 350 MB su `/dev/hda1` e la root di Linux in una partizione da 120MB su `/dev/hda2`.

*Hauke Laging (hauke@laging.de) e Bob Hall (bhall@hallfire.org) hanno notato un piccolo errore, suggerendomi che un MB è formato da 2**20 byte, non 2**10 byte. Grazie per la correzione. Inoltre, Hauke vorrebbe saperne di più su quelli che chiama codici a caratteri all'avvio di LILO, quando LILO si interrompe con scritte come LI, LI-, LIL- o simili. Sarebbe gradito un contributo relativo a questo problema o un valido collegamento sul web a riguardo.*

Eccolo qua (il contributo è di Zohar Stolar, zohar@numericable.fr):

B. codici di errore all'avvio di LILO

http://www.tldp.org/HOWTO/Bootdisk-HOWTO/a1483.html http://www.tldp.org/HOWTO/Bootdisk-HOWTO/a1483.h

Grazie per il collegamento.

MS-DOS non è stato in grado di installarsi correttamente quando il disco era nuovo. La versione 7 del DOS Novell aveva lo stesso problema. Fortunatamente il servizio opzioni di IBM aveva dimenticato di mettere il dischetto OnTrack nella scatola del disco, che sarebbe dovuto arrivare con un prodotto chiamato OnTrack Disk Manager. Chi ha installato solo MS-DOS immagino lo debba usare.

A questo punto è stata creata una tabella delle partizioni con il programma fdisk di Linux. MS-DOS 6.2 si è rifiutato di installarsi su `/dev/hda1` dicendo qualcosa come "questa versione di MS-DOS è per nuove installazioni. Il vostro computer ha già MS-DOS installato, quindi vi serve un rilascio di aggiornamento." In effetti, il disco era nuovo di zecca.

Quindi è stato usato nuovamente il programma fdisk di Linux per cancellare la partizione 1 dalla tabella. Questo ha accontentato MS-DOS 6.2, che ha creato esattamente la stessa partizione 1 appena rimossa per installarsi. MS-DOS 6.2 ha scritto il suo Master Boot Record sul disco, ma non è riuscito ad avviarsi.

Fortunatamente avevo un kernel Slackware su un floppy (creato dal programma di installazione della Slackware setup), per cui ho potuto avviare Linux e installare Lilo sull'MRB difettoso del DOS. Così ha funzionato. Ecco il file `/etc/lilo.conf` usato:

```
boot = /dev/hda
map = /lilo-map
delay = 100
ramdisk = 0             # Disattiva il ramdisk del kernel Slackware
timeout = 100
prompt
disk = /dev/hda         # Il BIOS vede solo i primi 500 MB.
   bios = 0x80          # Specifica il primo disco IDE.
   sectors = 63         # Si può ottenere questo numero leggendo la documentazione del proprio disco
   heads = 16
   cylinders = 2100
image = /vmlinuz
  append = "hd=2100,16,63"
  root = /dev/hda2
  label = linux
  read-only
  vga = extended
other = /dev/hda1
  label = msdos
  table = /dev/hda
  loader = /boot/chain.b
```

Dopo l'installazione di questi sistemi si potrà verificare che la partizione contenente i file zImage, boot.b, map, chain.b e message è in grado di usare un file system MSDOS, purché questo non sia compresso con stacker o doublespace. In questo caso avrei potuto creare una partizione DOS di 500 MB su `/dev/hda1`.

Ho anche imparato che OnTrack avrebbe scritto una tabella delle partizioni partendo da alcune dozzine di byte all'interno del disco invece che all'inizio, e quindi sarebbe stato possibile modificare il driver IDE di Linux per risolvere questo problema. L'installazione però sarebbe stata impossibile con il kernel Slackware precompilato. Alla fine, IMB mi ha mandato il dischetto onTrack. Ho chiamato il loro supporto tecnico che mi ha detto che Linux non funziona perché non usa il BIOS: ho regalato il loro dischetto.

8 Avvio da un floppy di recupero

Successivamente ho installato Windows 95 sul sistema dell'ufficio. Questo ha fatto sparire l'MBR di Lilo, ma non ha toccato la partizione Linux. I kernel impiegano molto tempo a caricarsi dal floppy, così ho creato un floppy con Lilo installato, per poter caricare il kernel dal disco IDE.

Il floppy è stato creato così:

```
fdformat /dev/fd0H1440      #  prepara le tracce sul dischetto vergine
mkfs -t minix /dev/fd0 1440 #  crea un file system di tipo minix
mount /dev/fd0 /mnt         #  lo monta sul punto di mount predefinito tmp
cp -p /boot/chain.b /mnt    #  copia al suo interno il chain loader
lilo -C /etc/lilo.flop      #  installa Lilo e la mappa sul dischetto.
umount /mnt
```

Si noti che il dischetto **deve essere montato quando si fa partire il programma di installazione**, in modo che Lilo possa scrivere correttamente il file della mappa.

Questo file è /etc/lilo.flop, ed è molto simile all'ultimo che abbiamo visto:

```
#  Crea un floppy in grado di avviare i kernel dall'HD.
boot = /dev/fd0
map = /mnt/lilo-map
delay = 100
ramdisk = 0
timeout = 100
prompt
disk = /dev/hda     # IDE da 1 GB, il BIOS vede solo i primi 500 MB.
   bios=0x80
   sectors = 63
   heads = 16
   cylinders = 2100
image = /vmlinuz
  append = "hd=2100,16,63"
  root = /dev/hda2
  label = linux
  read-only
  vga = extended
other = /dev/hda1
  label = msdos
  table = /dev/hda
  loader = /mnt/chain.b
```

Per finire, in ufficio c'era bisogno del DOS 6.2 ma non volevo toccare il primo disco. Ho aggiunto un controller e un disco SCSI, e ho creato un file system msdos con il programma di Linux mkdosfs. Questo disco viene visto come D: da Windows 95, ma ovviamente MSDOS non vuole avviarsi dal disco D:. Questo non è un problema con Lilo. Basterà aggiungere quanto segue al file `lilo.conf` dell'esempio 2.

```
other = /dev/sda1
  label = d6.2
  table = /dev/sda
  loader = /boot/any_d.b
```

Con questa modifica MSDOS-6.2 parte, credendo di trovarsi sul disco C: e che Windows 95 si trovi su D:.

9 Lilo dopo l'installazione di Mandrake Linux 9.1 su prodotti HP

19 novembre 2003

9.1 Descrizione dei prodotti usati in questo esperimento

Avviso: questa sezione ***NON*** *si deve intendere come pubblicità della produzione HP. In realtà, tutta una serie di computer HP usati hanno presentato mancanze negli alimentatori, problemi con i dischi rigidi e così via. Dall'altro lato, le batterie dei portatili si esauriscono prima di quanto ci si aspetti. A parte questi problemi, le macchine HP vanno bene.*

HP Omnibook 6000

Il computer portatile Omnibook 6000 è equipaggiato con un'unità DVD 'avviabile' e recentemente, ad una conferenza ICT, ho comprato un DVD-ROM avviabile con l'installazione di Mandrake Linux 9.1. Dopo aver avviato il portatile con quel DVD, esso è passato direttamente al menu di installazione di Linux.

HP Vectra VL420 (usato come server)

Al contrario, l'HP Vectra VL420 non ha un'unità DVD (ha solo un'unità CD), dunque l'installazione diretta da quel particolare DVD di installazione non è possibile. Però è possibile creare un floppy avviabile per far partire la procedura d'installazione. In effetti sono disponibili diverse immagini di avvio per quegli utenti che non hanno un'unità DVD (avviabile o no). Una di queste è un'immagine 'di rete'. Questo significa che in una rete locale ci deve essere un server NFS, FTP o HTTP dal quale possa partire l'installazione.

HP Vectra VL420 (usato come postazione di lavoro)

Un altro sistema desktop VL 420 in uso ha un disco rigido di riserva proveniente da una precedente installazione di Windows 2000 (in effetti quel disco IDE è stato spostato da un altro computer dove era un disco primario, mentre su questo sistema è il secondo disco per i dati di backup). La cosa carina è che ha installato un server HTTP e uno FTP (naturalmente utilizzabili se si avvia il sistema da quel disco). È una cosa buona perché ora è possibile usare uno di quei server.

Quindi è stato creato un floppy avviabile 'di rete', e con esso è stato avviato il primo Vectra VL420 (per essere usato come server Linux). Dopo poco tempo si è giunti al punto in cui si doveva scegliere il metodo di installazione (tramite il server NFS, o FTP, o HTTP). Il primo tentativo è stato tramite il secondo server HTTP 'di riserva' presente sull'altro sistema Vectra prima menzionato ma, indipendentemente dai permessi attribuiti al gruppo 'Everyone degli utenti Windows, si è sempre ottenuta la seguente risposta dall'impostazione di Linux:

Errore: impossibile ottenere il file... (o qualcosa di simile)

Poi si è tentato di usare il server FTP 'di riserva' dal secondo disco Vectra, che chiedeva anche gli indirizzi IP locale e remoto. Il tentativo ha avuto successo, ed è iniziato il caricamento di una parte dei file remoti in memoria senza proteste. Poco dopo l'installazione è arrivata allo stesso punto dell'Omnibook 6000: direttamente al menu di installazione, che chiede all'utente di scegliere una lingua da usare per l'installazione.

>Da questo punto il processo di impostazione è sempre lo stesso.

Sono state scelte/confermate le seguenti voci:

- una lingua da usare, a parte l'inglese (americano) che è la predefinita: sono state aggiunte l'Unicode e il serbo (sia Cyrillic che Latin); - un mouse e una tastiera; - un livello di sicurezza: sono stati accettati quelli predefiniti ('Standard' per il portatile e 'Higher' per il server);

Il successivo importante compito è stato scegliere una delle opzioni di partizionamento di *DrakX*:

- per il portatile è stata scelta l'opzione 'Usa lo spazio libero sulla partizione Windows', perché il portatile ha un disco IDE e se ne vuole usare una parte per Linux (a parte Windows 2000 Prof. già installato). Il Disk Management di Windows ha riferito:

```
Disk 0  15      MB      FAT     (Diagnostiche HP o simili)
        7.13    GB      FAT32   (C: "HPNOTEBOOK")
        20.80   GB      Spazio libero
```

Le due partizioni (FAT e FAT32) erano state create durante la procedura di installazione usando il CD di installazione fornito dalla HP.

All'inizio la configurazione di Linux si è lamentata perché la partizione di Windows è troppo frammentata, ed è stato necessario riavviare Windows, far girare l'utilità defrag, e poi far ripartire l'installazione di Mandrake Linux. Il processo di deframmentazione è durato circa 1 ora e mezza. Al riavvio, la configurazione di Linux voleva usare la partizione da 7.13 GB di Windows, al posto di 20.80 GB. È stata scelta l'opzione Usa lo spazio libero. Quindi la configurazione ha creato le partizioni per Linux: /dev/hda5 e /dev/hda7.

- per il Vectra VL420 è stato usato il partizionamento del disco personalizzato per via dei due dischi SCSI. Su uno dei due dischi è già installato Windows 2000 Server, mentre l'altro verrà interamente usato per un server Linux. A proposito, non ero sicuro di cosa avrebbe fatto l'opzione Cancella l'intero disco al passo successivo (cancellare tutto il disco o una partizione?), anche se forse sarebbe stata la giusta soluzione. *DrakX* ha riconosciuto i due dischi SCSI come **sda** e **sdb**, e si è scelto di installare Linux su **sdb**. Il primo passo è stato Cancella tutto, quello successivo è stato Auto allocare lo spazio sul secondo disco. Infine, dopo aver cliccato su 'Fatto', sono state create le partizioni per Linux /dev/sdb1 e /dev/sdb6.

9.2 Come appare LILO su questi prodotti HP

HP Omnibook 6000

```
boot=/dev/hda
map=/boot/map
vga=normal
default="windows"
keytable=/boot/us.klt
prompt
nowarn
timeout=100
message=/boot/message
menu-scheme=wb:bw:wb:bw
image=/boot/vmlinuz
        label="linux"
        root=/dev/hda5
        initrd=/boot/initrd.img
        append="quiet devfs=mount acpi=off"
        read-only
image=/boot/vmlinuz
        label="failsafe"
        root=/dev/hda5
        initrd=/boot/initrd.img
        append="failsafe devfs=nomount acpi=off"
```

```
        read-only
other=/dev/hda2                                 <--- /dev/hda1 sembra essere riservato per alcune di
        label="windows"
        table=/dev/hda
other=/dev/fd0
        label="floppy"
        unsafe
```

HP Vectra VL420 (installato come sistema client Linux desktop)

```
boot=/dev/hda
map=/boot/map
vga=normal
default="windows"
keytable=/boot/us.klt
prompt
nowarn
timeout=100
message=/boot/message
menu-scheme=wb:bw:wb:bw
image=/boot/vmlinuz
        label="linux"
        root=/dev/hda5
        initrd=/boot/initrd.img
        append="quiet devfs=mount acpi=off"
        vga=788                                 <--- questa riga manca su portatili con schermo LCD.
        read-only
image=/boot/vmlinuz
        label="linux-nonfb"
        root=/dev/hda5
        initrd=/boot/initrd.img
        append="devfs=mount acpi=off"
        read-only
image=/boot/vmlinuz
        label="failsafe"
        root=/dev/hda5
        initrd=/boot/initrd.img
        append="failsafe devfs=nomount acpi=off"
        read-only
other=/dev/hda1                                 <--- /dev/hda1 in questo caso non sembra riservato p
        label="windows"                              Qui era già installato Windows 2000 Professi
        table=/dev/hda                               (probabilmente senza il CD fornito da HP).
other=/dev/hdb1                                 <--- questo è il disco di riserva con Windows 2000 S
        label="windows2"                             Ho alcuni dati su questo disco e lo uso come
        table=/dev/hdb                               disco di backup su quella postazione desktop
        map-drive=0x80                               Non ho mai provato ad avviare il computer da
           to=0x81                                   l'impostazione della Mandrake lo dava come o
        map-drive=0x81                               (Ed è stato utile come server FTP, avendo bi
           to=0x80                                   Mandrake Linux sull'altro computer senza uni
other=/dev/fd0
        label="floppy"
```

```
        unsafe
```

HP Vectra VL420 (installato come sistema desktop Linux con funzionalità da server)

```
boot=/dev/sda                                    <--- /dev/sda è il primo disco SCSI, su cui risiede
map=/boot/map
vga=normal
default="windows"
keytable=/boot/us.klt
prompt
nowarn
timeout=100
message=/boot/message
menu-scheme=wb:bw:wb:bw
image=/boot/vmlinuz
        label="linux"
        root=/dev/sdb1                           <--- /dev/sdb1 è il secondo disco SCSI, su cui risie
        initrd=/boot/initrd.img
        append="quiet devfs=mount acpi=off"
        vga=788
        read-only
image=/boot/vmlinuz
        label="linux-nonfb"
        root=/dev/sdb1
        initrd=/boot/initrd.img
        append="devfs=mount acpi=off"
        read-only
image=/boot/vmlinuz-secure
        label="linux-secure"                     <--- c'entra qualcosa con la sicurezza del server Li
        root=/dev/sdb1
        initrd=/boot/initrd-secure.img
        append="quiet devfs=mount acpi=off"
        read-only
image=/boot/vmlinuz
        label="failsafe"
        root=/dev/sdb1
        initrd=/boot/initrd.img
        append="failsafe devfs=nomount acpi=off"
        read-only
other=/dev/sda1                                  <--- /dev/sda1 è la prima partizione del primo disco
        label="windows"
        table=/dev/sda
```

9.3 Conclusioni

Dagli esempi precedenti si può notare che sono stati usati diversi tipi di computer con diversi tipi di dischi rigidi. In alcuni casi c'è solo un disco IDE, in altri ce ne sono due, in altri ancora ci sono un paio di dischi SCSI, ecc. Indipendentemente da questo, si è sempre cercato di installare LILO nell'MBR, posizionato sul *primo* disco. Pare che ora, finalmente, Linux abbia risolto il vecchio problema del *cilindro 1024*. In effetti LILO sembra in grado di avviare Linux indipendentemente dal fatto che sia posizionato vicino al resto della partizione Linux o no.

Ci sono alcune altre considerazioni relative all'esperimento precedente, ma fanno parte di un altro bel documento:

Linux+WindowsNT http://tldp.org/HOWTO/Linux+WinNT.html mini-HOWTO.

10 Bibliografia

13 gennaio 2008

Avvertimento: spesso ho partecipato a conferenze ICT (inter)nazionali in Serbia e Montenegro, così come all'estero, presentando relazioni e tutorial. Quello che ho cercato di fare è diffondere il più possibile le idee di base e l'utile missione dell'hobby del radioamatore, e del suo possibile uso nell'insegnamento dell'ingegneria. Potete scommettere che, dove possibile, ho cercato di farlo fare ai miei lettori con Linux. Oltre a questo, ho scritto vari articoli per una serie di giornali scientifici e di altro tipo. Ecco un elenco degli articoli che ho scritto, e delle relazioni presentate alle conferenze fino ad oggi.

Chi volesse ripubblicare o inoltrare le mie spontaneamente offerte relazioni ad alcuni giornali o ad altri media pubblici, si senta libero di contattarmi. Alcuni dei miei articoli sono scritti in serbo cirillico, altri sono in inglese, e alcuni sono scritti in entrambe le lingue.

- "U prilog I.A.C.", MI (il giornale dell'organizzazione dei giovani scienziati), No. 69, 1990.

- "U prilog I.A.C. (2)", MI (il giornale dell'organizzazione dei giovani scienziati), No. 70, 1990.

- "Vise od radio-amaterskog hobija", Vojska, No. 163, 1995.

- "Korak ka zvezdama", Vojska, No. 200, 1996.

- "Die Gefahr von Innen - Internet gegen Amateurfunk", AMSAT-DL Journal, No. 4, dic./feb. 96/97.

- "Kakva nam organizacija (ne) treba?", Radioamater, feb. 1997.

- "Kakva nam organizacija (ne) treba? (2)", Radioamater, apr./mag. 1997.

- "Sateliti umiru padajuci", Vojska, No. 235, 1997.

- "The Internet is not the Enemy", QST, ago. 1998.

- "Novi radio-amateri za novi vek", Antena, giugno 2000.

- "Racunarske komunikacije putem radio-veza i zastita pristupa", Bezbednost, No. 3, 2000.

- "Paket-radio - Racunarske komunikacije putem radio-veza", dibattito, "Info-Teh", Vrnjacka Banja, Serbia, 2001.

- "Racunarske komunikacije putem radio-amaterskih veza", dibattito, "YU-Info", Kopaonik, Serbia, 2002.

- "Computer Communications over radio", presentazione, "Linux FEST", Belgrado, Serbia, 2002.

- "Paket-radio - Radio-amaterske digitalne veze", dibattito, "Kongres JISA", Herceg Novi, Montenegro, 2002.

- "Paket-radio (2) - Modemi za radio-veze", dibattito, "Info-Teh", Vrnjacka Banja, Serbia, 2002.

- "Alternativne racunarske mreze", catalogo del festival, "INFOFEST", Budva, Montenegro, 2002.

- "Alternative computer networks", dibattito, "TELFOR", Belgrado, Serbia, 2002.

- "With rule and regulation improvements to the progress" dibattito, "TELFOR", Belgrado, Serbia, 2002.

- "Racunarske komunikacije putem radio-amaterskih veza (2)", dibattito, "YU-Info", Kopaonik, Serbia, 2003.

- "Racunarske komunikacije putem radio-amaterskih veza (3)", dibattito, "YU-Info", Kopaonik, Serbia, 2003.

- "Paket-radio (3) - Programske mogucnosti na strani servera", dibattito, "Info-Teh", Vrnjacka Banja, Serbia, 2003.

- "Paket-radio (4) - Legal rules and regulations in the amateur computer networks", dibattito, "Info-Teh", Vrnjacka Banja, Serbia, 2003.

- "Packet-radio (2) - With rule and regulation improvements to the progress", dibattito, "Kongres JISA", Herceg Novi, Montenegro, 2003.

- "Alternativne racunarske mreze (2)", catalogo del festival, "INFOFEST", Budva, Montenegro, 2003.

- "Alternativne racunarske komunikacije putem radio-veza", Info M, 6-7/2003.

- "Legal Rules and Regulations in the Amateur Radio Computer Networks", dibattito, "22nd ARRL and TAPR Digital Communications Conference", Hartford, CT USA, 2003.

- "Favoritism", IEEE Potentials, ott./nov. 2003

- "Alternative computer networks (2)", dibattito, "TELFOR", Belgrado, Serbia, 2003.

- "With rule and regulation improvements to the progress (2)" dibattito, "TELFOR", Belgrado, Serbia, 2003.

- "XI Telekomunikacioni forum - TELFOR 2003", Info M, 8/2003.

- "Aktivnosti organizacije IEEE Computer Society - YU Chapter"

Info M, 8/2003.

- "Yugoslavia IEEE Student Branch", IEEE Region 8 News, Vol. 7 No. 1, feb/mar 2004

- "Radio-amaterske racunarske mreze", tutorial, "Info-Teh", Vrnjacka Banja, Serbia, 2004.

- "Radio-amaterske racunarske mreze", tutorial, "Kongres JISA", Herceg Novi, Montenegro, 2004.

- "The Amateur Radio as a Learning Technology in Developing Countries", dibattito, "ICALT/TEDC 2004", Joensuu, Finlandia, 2004.

- "The Amateur Radio in Engineering Education", seminario, "IEEE-EESTEC Technical Conference 2004", Arcavacata di Rende (Cosenza), Italia, 2004.

- "The Conference Low-Down", IEEE Potentials, feb./mar. 2005

- "ICALT 2004, IEEE Novi Sad SB, Serbia & Montenegro", IEEE Region 8 News, Vol. 8 No. 1, Mar 2005

- "The Conference Low-Down", IEEE Potentials, apr./mag. 2005

- "The Amateur Radio Networking And Computing", dibattito, "PSU-UNS ICEE 2005", Novi Sad, Serbia, 2005

- "Radio-amaterske racunarske mreze", tutorial, "Info-Teh", Vrnjacka Banja, Serbia, 2005 (introduzione: dr D. Surla, PMF)

- "Radio-veze", sessione di tavola rotonda, "Kongres JISA", Herceg Novi, Montenegro, 2005.

- "The New Amateur Radio University Network - AMUNET", dibattito, "9th WSEAS CSCC Multiconference", Vouliagmeni, Atene, Grecia, 2005.

- "The perspectives of the Amateur University Network - AMUNET", WSEAS Transactions on Communications, Vol 4, pp 834, set. 2005.

- "Conferences in Serbia and Montenegro", IEEE Region 8 News, Vol. 8 No. 3, set. 2005.

- "The Conference Low-Down", IEEE Potentials, dic. 2005

- "Divided We Fall", The Institute, IEEE, dic. 2005

- "University Networking Through the Amateur Radio Communications", conferenza plenaria, "3rd WSEAS/IASME Int. Conf. on Engineering Education", Vouliagmeni, Atene, Grecia, 2006.

- "The New Amateur Radio University Network - AMUNET (Part 2)",

```
dibattito, "10th WSEAS CSCC Multiconference",
  Vouliagmeni, Atene, Grecia, 2006.

- "University Networking Through the Amateur Radio Communications",
  tutorial, "10th WSEAS CSCC Multiconference",
  Vouliagmeni, Atene, Grecia, 2006.

- "Education in Peril", The Institute, IEEE, set. 2006

- "The Conference Low-Down", IEEE Potentials, dic. 2006

- "Amateur Radio in Engineering Education", tutorial, "Info-Teh",
  Vrnjacka Banja, Serbia, 2007.

- "Paket-radio (5) - Programske mogucnosti na strani klijenta",
  dibattito, "Info-Teh", Vrnjacka Banja, Serbia, 2007.

- "Amateur Radio in Engineering Education", tutorial, "EUROCON",
  Warsaw, Polonia, 2007.

- "Bringing New Life in Old Computers", dibattito, "26th ARRL
and TAPR Digital Communications Conference", Hartford, CT USA, 2007.

- "The Conference Low-Down", IEEE Potentials, set/ott 2007

- "My trip to Eurocon - a member's view", IEEE Region 8 News,
Vol. 10 No. 3, Dec 2007
```

Oltre a questi articoli pubblicati e relazioni presentate, ho studiato per avere una laurea M. Sc. in informatica. Sono anche membro delle seguenti associazioni: IEEE Computer Society, IEEE Communications Society, WSEAS e ACM. Inoltre lavoro volontariamente per istituire una rete di computer accademica che usi le stazioni radio degli amatori come media. Queste reti esistono altrove nel mondo, e invito i loro amministratori a contattarmi per cooperare.

11 Ulteriori informazioni

11.1 Copyright

11.2 Liberatoria

Si usino le informazioni di questo documento a proprio rischio. Non si assume alcuna potenziale responsabilità per l'uso di questo documento. L'uso dei concetti, esempi o altro contenuto di questo documento è completamente a proprio rischio.

Si raccomanda di fare un backup del proprio sistema prima di effettuare installazioni sostanziali, e backup a intervalli regolari.

11.3 Novità

In aggiunta ai documenti di Lilo, ci sono diversi mini-howto che possono rivelarsi utili per le proprie necessità. Tutti si chiamano "Linux+*qualche-SO*", in alcuni casi *qualche-SO*, e trattano la coesistenza tra Linux e altri sistemi operativi. Per esempio, NT OS Loader + Linux mini-HOWTO di Bernd Reichert descrive come aggiungere una voce per Linux al menu del caricatore di Windows NT già esistente. Poi c'è il

Linux+WindowsNT `http://tldp.org/HOWTO/Linux+WinNT.html` mini-HOWTO scritto da me, che spiega come aggiungere una voce per NT al menu di Lilo già esistente (più dettagliata di quanto si sia fatto qui). Anche Multiboot-with-LILO descrive come far convivere varie versioni di Windows con Linux.

Questo mini-HOWTO verrà migliorato di tanto in tanto. Se si ritiene superata la versione di questo HOWTO presente sul proprio CD di installazione di Linux, si può controllare su Internet per trovare il rilascio più recente: può essere trovato nell'homepage principale del Linux Documentation Project `http://www.linuxdoc.org/` *o a questo indirizzo:* Linux Documentation Project `http://www.tldp.org/` .

11.4 Ringraziamenti

Questa versione esiste grazie a:

```
Cameron Spitzer (cls@truffula.sj.ca.us)
Alessandro Rubini (rubini@linux.it)
Tony Harris (tony@nmr.mgh.harvard.edu)
Marc Tanguy (mtanguy@ens.uvsq.fr)
Dragomir Kalaba, un 'guru' Linux locale
```

È possibile inviare commenti o suggerimenti al mio indirizzo email: skoric at eunet dot yu

11.5 HOWTO

I seguenti HOWTO sono punti di partenza per ottenere informazioni di base, così come per scoprire come risolvere uno specifico problema. Alcuni HOWTO pertinenti sono: `Bootdisk`, `Installation`, `SCSI` e `UMSDOS`. Il loro sito principale è l' *archivio LDP* `http://metalab.unc.edu/LDP/` presso Metalab (precedentemente conosciuto come Sunsite).

11.6 Mini-HOWTO

Questi sono i testi liberi più piccoli relativi agli HOWTO. Alcuni mini-HOWTO pertinenti sono: `Backup-With-MSDOS`, `Diskless`, `LILO`, `Large Disk`, `Linux+DOS+Win95+OS2`, `Linux+OS2+DOS`, `Linux+Win95`, `Linux+WindowsNT`, `Linux+NT-Loader`, `NFS-Root`, `Win95+Win+Linux`, `ZIP Drive`, `FBB packet-radio BBS`. Si possono trovare insieme agli HOWTO, di solito in una sottodirectory chiamata `mini`. Notare che è in programma la loro conversione a SGML, per diventare veri e propri HOWTO in tempi brevi.

11.7 Risorse locali

La maggior parte delle distribuzioni Linux installa una directory di documenti. Si provi a guardare nella directory */usr/doc* , dove molti pacchetti archiviano la propria documentazione, file README ecc. Al suo interno di troverà anche un archivio di HOWTO (*/usr/doc/HOWTO*) già formattati, insieme all'archivio dei mini-HOWTO (

/usr/doc/HOWTO/mini `file:///usr/doc/HOWTO/mini`) in puro testo.

Molti dei file di configurazione menzionati possono essere trovati nella directory */etc* . In particolare, si lavorerà con il file */etc/fstab* , che imposta i punti di mount delle partizioni, e forse anche con il file */etc/mdtab* , che viene usato dal sistema `md` per impostare il RAID.

Il sorgente del kernel nella directory

/usr/src/linux `file:///usr/src/linux` è, naturalmente, la documentazione definitiva. In altre parole, *usa il sorgente, Luke*. Bisogna anche far presente che non solo il kernel viene fornito con il suo codice sorgente, perfino commentato (almeno in parte), ma anche con una

directory di documentazione `file:///usr/src/linux/Documentation` istruttiva. Si dovrebbe leggere prima di tutto questa directory se si hanno domande sul kernel: risparmierà a se stessi e agli altri tempo e forse anche imbarazzo.

Si controlli sempre anche il file del registro di sistema (*/var/log/messages*) per controllare cosa succede, e in particolare com'è andato l'avvio nel caso in cui la schermata sia passata oltre. Si usi `tail -f /var/log/messages` in una finestra o schermo separato per avere un aggiornamento continuo di quello che succede nel proprio sistema.

Si può anche approfittare del file system */proc* , che è una finestra sul lavoro interno del proprio sistema. Si usi `cat` al posto di `more` per vedere i file che vengono indicati di lunghezza zero. Pare che anche `less` si comporti bene.

11.8 Pagine web

C'è un gran numero di istruttive pagine web, che per loro natura cambiano velocemente, quindi non ci si sorprenda se questi collegamenti diventano obsoleti rapidamente.

Un buon punto di partenza naturalmente è la home page del

Linux Documentation Project `http://www.linuxdoc.org/` , o questa: *Linux Documentation Project* `http://www.tldp.org/` , una centrale istruttiva per documentazione, progetti e molto, molto altro.

Si prega di far sapere se si conoscono altre guide interessanti.

12 Ottenere aiuto

Alla fine si potrebbe non riuscire a risolvere i propri problemi, e si potrebbe aver bisogno dell'aiuto di qualcun altro. Il modo più efficiente è chiedere a qualcuno nelle proprie vicinanze, o al più vicino gruppo di utenti Linux, che può essere trovato sul web.

Un'altra possibilità è chiedere su Usenet News in uno dei molti, molti newsgroups disponibili. Il problema è che di solito essi hanno un così alto volume e rumore (chiamato basso rapporto segnale/rumore) che la propria domanda può facilmente restare senza risposta.

La cosa più importante è chiedere correttamente, o non si verrà presi sul serio. Dire solo *il mio disco non funziona* non sarà di aiuto, anzi alzerà ulteriormente il livello del rumore e con un po' di fortuna qualcuno chiederà chiarimenti.

Piuttosto bisogna descrivere i propri problemi con tutti i dettagli necessari per permettere agli altri di aiutarci. Il problema potrebbe essere in qualcosa di inaspettato. Perciò ci si ricordi di elencare le seguenti informazioni sul proprio sistema:

Hardware

- Processore
- DMA
- IRQ
- Chipset (LX, BX ecc.)
- Bus (ISA, VESA, PCI ecc.)
- Scheda di espansione usata (controller del disco, video, IO ecc.)

Software

- BIOS (Sulla scheda madre, e possibilmente l'adattatore SCSI)
- LILO, se usato
- La versione del kernel di Linux usata, insieme ad eventuali modifiche e patch
- Parametri del kernel, se presenti
- Software che provoca l'errore (col numero di versione o la data)

Periferiche

- Tipo di disco rigido con il nome del produttore, versione e tipo
- Altre periferiche pertinenti connesse agli stessi bus

Ci si ricordi che il testo relativo all'avvio è registrato in `/var/log/messages`, che può rispondere a molte delle domande precedenti. Ovviamente se il disco fallisce non sarà possibile avere il registro salvato sul disco, ma si potrà far scorrere all'indietro la schermata usando i tasti `SHIFT` e `PAGE UP` . Può anche essere utile includere parti della schermata nella propria richiesta di aiuto, ma senza esagerare: la si mantenga *breve* , dato che un file di registro completo inviato su Usenet News è molto più che fastidioso.