Diskless Nodes HOW-TO document for Linux

Robert Nemkin *buci@math.klte.hu* , Al Dev (Alavoor Vasudevan) - Curatore di questo HOWTO *alavoor@yahoo.com* , Markus Gutschke *markus+etherboot@gutschke.com* , Ken Yap *ken.yap@acm.org* , Gero Kuhlmann *gero@gkminix.han.de*

v 15.0, 6 Settembre 2000

Questo documento descrive come preparare una macchina Linux diskless (senza dischi). Dato che la tecnologia sta avanzando rapidamente, le schede di rete stanno diventando più economiche e molto più veloci - al momento l'ethernet a 100 MBit è standard e tra circa uno o due anni le schede ethernet a 1000 MBit, ossia 1 GigaBit, diventeranno uno standard industriale. Con le schede di rete ad alta velocità l'accesso remoto diventerà veloce quanto l'accesso al disco locale, il che renderà i nodi diskless una possibile alternativa alle workstation nelle LAN. Inoltre i nodi diskless eliminano i costi di aggiornamento del software e i costi di amministrazione di sistema, tipo backup e recupero, che verranno centralizzati da parte del server. I nodi diskless inoltre permettono la "condivisione/ottimizzazione" delle CPU centralizzate del server, della memoria, dei dischi rigidi, dei lettori di nastri e dei lettori di CD-ROM. I nodi diskless permettono la mobilità degli utenti, cioè gli utenti possono effettuare il log on da uno qualsiasi dei nodi diskless e non sono legati ad una workstation. Le macchine Linux diskless eliminano completamente la necessità di avere floppy disk, lettori CD-ROM, lettori di nastri e dischi rigidi locali. Un nodo diskless ha SOLO una scheda di rete, 8MB RAM, una CPU di fascia bassa ed una semplicissima scheda madre senza slot o altri connettori per dischi rigidi, modem, CD-ROM, floppy ecc. Con i nodi Linux diskless si possono eseguire programmi su una macchima Linux remota dotata di 64 CPU SMP o anche su di un supercomputer Linux! I nodi diskless abbassano il "costo totale di possesso" del sistema. Questo documento è copyright di Robert Nemkin e degli altri autori precedentemente elencati. La politica di copyright è GPL. Grazie a Bela Kis *bkis@cartan.math.klte.hu* `<mailto:bkis@cartan.math.klte.hu>` per aver tradotto in inglese la versione iniziale, v 0.0.3, di questo documento (che era un mini-howto). Traduzione in italiano di *Fabrizio Stefani* `<mailto:f.stef@tiscalinet.it>`, 19 novembre 2000.

Indice

1 Comprare è più economico che assemblare!

Prima o poi comprare un computer linux diskless diventerà più economico che assemblarlo! Si vedano i seguenti siti commerciali che vendono computer Linux diskless e schede di rete per computer Linux diskless. Queste società producono **grosse quantità** di computer Linux diskless e vendendone milioni di unità possono ridurne il costo unitario. Tutte le prime 1000 compagnie USA della classifica di Fortune nel prossimo futuro sostituiranno i PC MS Windows con computer Linux diskless poiché questi possono far girare programmi sia Linux che MS Windows 95 (tramite il software BIOS VMWare). *VMWare* `<http://www.vmware.com>`

NON è un emulatore ma ha un BIOS che permette di installare Windows 98/NT come SO ospite di Linux. È possibile usare il comando 'xhost' e l'ambiente DISPLAY dal nodo diskless per eseguire i programmi Windows95/Linux. Vedere 'man xhost' su Linux. Per eseguire programmi per Windows95/NT su nodi diskless Linux, è possibile anche usare il Virtual Network Computing (VNC). Lo si può scaricare da <http://www.uk.research.att.com/vnc>.

- Linux Systems Labs Inc., USA <http://www.lsl.com>. Si selezioni "Shop On-line" e poi "HardWare" e verranno elencati tutti i computer diskless. Telefono: 1-888-LINUX-88.
- Diskless Workstations Corporation, USA <http://www.disklessworkstations.com>
- Unique Systems of Holland Inc., Ohio, USA <http://www.uniqsys.com>

Se siete intenzionati a comprare dei computer Linux diskless, potreste essere molto interessati alla lettura completa di questo documento.

2 Computer diskless con Microsoft Windows 95/NT !!

Sebbene Microsoft Windows 95/NT **NON** supporti nodi diskless esiste una intelligente scappatoia per aggirare l'ostacolo. La Microsoft corporation ne sarà **sorpresa**!!

2.1 Il pacchetto VMWare

Usare il BIOS software *VMWare* <http://www.wmware.com> con Linux che può ospitare Windows 95/98/NT. Linux sarà il SO "ospitante" e Windows 95/NT sarà il SO "ospitato". *VMWare* <http://www.vmware.com> NON è un emulatore ma ha un BIOS che permette di installare Windows 95/98/NT come SO ospite di Linux. Installare VMWare sul server Linux e poi installare Windows 95/NT in VMWare.

È possibile usare il comando 'xhost' e l'ambiente DISPLAY da **qualsiasi** nodo diskless. Vedere 'man xhost' su Linux. Sul nodo diskless digitare:

```
        export DISPLAY=nomecomp_server:0.0
```

dove nomecomp_server è il nome della macchina server. Lanciare un X-terminal con

```
        xterm
```

Usando *VMWare* <http://www.vmware.com> i computer Linux diskless possono far girare sia programmi Linux che MS Windows 95. Il sito VMWare è su <http://www.vmware.com>.

2.2 Il pacchetto VNC della AT&T

È possibile anche usare la tecnologia VNC (Virtual Network Computing) del gigante delle telecomunicazioni AT&T. VNC è rilasciata sotto GPL ed è software libero. Usando VNC è possibile eseguire programmi per Windows 95/NT su computer diskless Linux, ma che in realtà girano su un server remoto Windows 95/NT. VNC è reperibile presso <http://www.uk.research.att.com/vnc>.

3 Vantaggi dei computer diskless

I computer Linux diskless diventeranno **immensamente** popolari e saranno il prodotto di questo e del prossimo secolo. I computer Linux diskless avranno grande successo a causa della disponibilità di schede di rete ad alta velocità a prezzi bassissimi. Oggi le schede di rete a 100 Megabit per secondo (velocità di trasferimento di 12.5 MB per secondo) sono comuni e fra circa uno o due anni le schede di rete a 1000 Mbit (velocità di trasferimento di 125 MB per secondo) diventeranno molto economiche e saranno lo standard.

Nel prossimo futuro i costruttori di monitor metteranno CPU, NIC e RAM proprio **dentro** il monitor realizzando un computer diskless! Così facendo si elimina il case del computer diskless e si risparmia spazio. Il monitor avrà delle uscite per il mouse, la tastiera, una presa RJ45 per la rete e una per l'alimentazione.

Di seguito sono elencati i vantaggi dell'uso di computer diskless:

- I computer diskless Linux possono eseguire SIA programmi MS Windows 95/NT SIA programmi Linux.
- Il costo totale di possesso è molto basso per i computer diskless. Il costo totale di possesso è pari al costo iniziale di acquisto più il costo di manutenzione. Il costo di manutenzione è di solito **da 3 a 5 volte** il costo iniziale di acquisto del computer e tale costo ricorre di anno in anno. Per i computer diskless il costo di manutenzione viene **completamente eliminato**!!
- Tutti i backup sono centralizzati su un singolo server principale.
- I dati sono più sicuri essendo situati sul server.
- Per i client diskless non c'è nessun bisogno di batterie UPS, condizionamento d'aria, ambienti privi di polvere; solo il server ne ha bisogno.
- Protezione dall'attacco di virus: i virus per computer non possono attaccare i computer diskless poiché essi non hanno nessun disco rigido. I virus non possono fare nessun danno ai computer diskless. Solo una singola macchina server deve essere protetta dall'attacco dei virus; ciò fa risparmiare all'azienda milioni di dollari rendendo inutile l'installazione di vaccini e la pulizia dei dischi rigidi!!
- Il server può avere dischi rigidi di grande potenza ed elevate prestazioni, può ottimizzare l'uso dello spazio disco condividendo quello necessario a più utenti di computer diskless. La tolleranza ai guasti del disco rigido è ottenibile usando tecniche RAID sul server principale.
- Il server può avere più CPU a 64 bit funzionanti in SMP o anche essere un supercomputer Linux. La potenza delle CPU può essere condivisa da più utenti di computer diskless.
- È possibile la condivisione della RAM da parte di più utenti di computer diskless. Per esempio, se diversi utenti stanno usando un web browser, nella RAM del server ci sarà una sola copia del browser. Nel caso di PC Windows 95, diversi utenti devono invece avere ciascuno una copia individuale del browser nella RAM locale e si avrà così uno spreco di memoria RAM.
- I computer Linux diskless possono far girare programmi su server multipli usando "xhost" e l'ambiente DISPLAY.
- Sono necessari pochissimi amministratori per gestire il server centrale al contrario dei client PC Windows 95 che richiedono parecchi amministratori.
- Amministrazione zero da parte dei client diskless. I computer diskless non hanno assolutamente bisogno di manutenzione e non creano nessun problema.
- Lunga vita dei client diskless; più di **100 anni** senza aggiornamenti di hardware o software.
- Eliminazione dell'installazione/aggiornamento di hardware e software da parte dei client diskless.

- Eliminazione del costo di cdrom, floppy, lettori di nastri, modem e batterie UPS, porte parallele per le stampanti, porte seriali, ecc.
- Possono funzionare in luoghi, tipo i pavimenti delle fabbriche, in cui i dischi rigidi potrebbero risultare troppo fragili.

4 Linux Terminal Server Project - LTSP

LTSP è un progetto di codice open source per costruire computer Linux diskless.

Nel sito LTSP si trovano i pacchetti RPM per Linux Redhat e quelli per Linux Debian che possono far risparmiare un sacco di tempo. I capitoli successivi in questo documento sono da intendersi solo per fini accademici ed è possibile leggerli se si ha del tempo libero.

Si visiti il sito LTSP e i siti correlati presso:

- `<http://www.ltsp.org>`
- `<http://www.disklessworkstations.com>`
- `<http://www.slug.org.au/etherboot>`
- `<http://metalab.unc.edu/Linux/HOWTO/XFree86-Video-Timings-HOWTO.html>`

Argomenti correlati che vale la pena vedere:

- NCD X-terminal `<http://www.linuxdoc.org/HOWTO/mini/NCD-X-Terminal.html>`

5 Costruire un programmatore EEPROM

5.1 Che cos'è?

Questo capitolo è stato scritto da Abhijit Dasgupta ed è preso da `<http://www.nnaf.net/~abhijit/eep>`. L'indirizzo email di Abhijit è: *abhijit@ans.net*

Una foto del programmatore è visibile presso: `<http://www.nnaf.net/~abhijit/pictures/eeprom-burner.jpg>`

Questo è un programmatore per le EEPROM del tipo 2816 e 2864. Sono disponibili vari progetti, ma l'obiettivo fondamentale è di avere qualcosa che sia:

- economico (meno di 35$)
- facile da costruire e che usi solo componenti facilmente reperibili
- funzionante con Linux

Questo programmatore usa una manciata di integrati logici 74HCTxx, tutti disponibili nel negozio di componenti elettronici sotto casa! Usa l'interfaccia della porta parallela del PC. Abhijit ha scritto il codice del driver solo per Linux, ma dovrebbe essere facile modificarlo per altri sistemi operativi.

È stato usato per scrivere le PROM per il netboot di schede ethernet, che a loro volta sono state usate per realizzare delle macchine Linux diskless. Per i dettagli riguardanti le modalità operative si vedano i pacchetti netboot e/o etherboot. È anche possibile usarlo per costruire dei sistemi microcontrollori con ROM esterna (ad es. 8031).

5.2 Costruzione

ATTENZIONE: È facile distruggere la porta parallela del proprio PC collegandoci qualcosa. Facendo collegamenti sbagliati e in caso di incidenti di tipo elettrico è possibile danneggiare o distruggere l'intero PC, comprese le periferiche e tutto ciò che gli è attaccato, ed è anche possibile ferire chi si trova lì vicino. FATE MOLTA ATTENZIONE.

Liberatoria: Usate questo apparecchio a vostro rischio e pericolo. Non è data nessuna garanzia di nessun tipo.

Il programmatore può essere sperimentato su una breadboard, ma per realizzarne una versione permanente usate una protoboard. Usa abbondantemente i condensatori di bypass da 0.1uF per le linee di alimentazione. L'alimentazione a 5V può essere presa direttamente dal PC, ma fate attenzione.

5.3 Elenco dei componenti

- **Circuiti integrati:** 74HCT123, 74HCT132, 74HCT138, 74HCT157, 74HCT574 (1 ea) e 2 74HCT259.
- **Resistori:** 100K, 10K, 1K e 390 ohm (1 ea).
- **Condensatori:** 100pF, 1uF, (1 ea) e 3 condensatori da 0.1uF per il bypass della linea di alimentazione.
- **Vari:** 1 LED, 1 interruttore SPST, piattina da 25 fili con connettore maschio DB25, zoccolo ZIF a 28 pin (oppure si possono usare delle piccole breadboard), dei ponticelli.

5.4 Uso

Scaricate il software `<http://www.nnaf.net/~abhijit/eep/eeprom.tar.gz>`. Per compilarlo basta andare nella directory con i sorgenti e digitare 'make'.

```
readrom
-------
readrom legge da una EEPROM 2816/2864 un dato numero di byte, a partire
da una data posizione, e li invia allo standard output in formato
binario (raw) o ascii-hex.

Uso:

    readrom -b|-t posizione dimensione

dove

    -b            uscita in binario (raw)
    -t            uscita in formato testo (ascii-hex)
    posizione     indirizzo di inizio nella EEPROM: 0..8191
    dimensione    numero di byte da emettere: 0..8192

Esempi:

# legge il contenuto di una 2864 in binario (raw) e lo salva in un file:
readrom -b 0 8192 > contenuto.bin
```

```
# mostra 80 byte a partire dal 32esimo:
readrom -t 32 80


writerom
--------

writerom legge un dato numero di byte dallo standard input e li
scrive in una ROM 2816/2864 a partire dalla posizione specificata.
writerom verifica tutti i byte che scrive nella EEPROM.

Uso:

    writerom posizione dimensione

dove

    posizione     indirizzo di inizio nella EEPROM: 0..8191
    dimensione    numero di byte da scrivere: 0..8192

Esempio:

# scrive nella EEPROM 8192 byte prendendoli dal file ne.lzrom:
writerom 0 8192 < ne.lzrom
```

5.5 Schema circuitale

Lo schema è disegnato in ASCII, ma è disponibile una versione in PostScript (graficamente migliore) presso <http://www.nnaf.net/~abhijit/eep/eeprom/schematic.ps>

```
                       +-------+
              +5-------|RST    |               +5---o o o J1     +-----------+
              +5--o----|/CLR1  |         10K          |          |           |
                  |    |       |-----o--/VVV\-- +5    +----------|26 +5(NC)  |
      +------+    |    |1/2 123|     |              +----------->|27 NC(/WE) |
 16 o-|/CS2  |    |    |       |--||-+              | +--------->|23 /WE(A11)|
      |   CS1|----o----|B1     | 100pF              | |          |           |
      |      |         |    /Q1|---------->---------o o o J2     |  zoccolo  |
      |    Y1|---------|/A1    |                        |        |   ZIF28   |
      | 138  |         +-------+         _ 1/2 74HCT132 |        |    per    |
      |      |                     +5 --| \   __        |        | 2816(2864)|
      |    Y2|--------------------------|  O--| \       |        |           |
  8 o-|A2    |                +-------+ |_/   |  O-------------->|/OE        |
  7 o-|A1  Y4|--------------->|EN   Y7|-----o-|_/       |        |           |
  6 o-|A0  Y3|----+    +5-----|RST    |     |           |        |           |
      |    Y0|-+  |           |     Y6|--+  |           |        |           |
      |  /CS3| |  |           | 259   |  |  |           |        |           |
      +------+ |  |           |     Y4|--|--|-----------|------->|2 NC(A12)  |
            |  |  |           |     Y3|--|--|-----------+        |           |
  5 o--->---|--|--|--------o--|D    Y2|--|--|------------------->|A10       |
```

```
 4 o--->---|--|--|------o-|--|A2   Y1|--|--|------------------->|A9         |
 3 o--->---|--|--|----o-|-|--|A1   Y0|--|--|------------------->|A8         |
 2 o--->---|--|--|--o-|-|-|--|A0     |  |  |                    |           |
           |  |  |  | | | |  +-------+  |  |      +5------------|28 NC(+5)  |
           |  |  |  | | | |              |  |      +5----/VVV\---|1 NC(RDY)  |
           |  |  |  | | | |  +-------+  |  |             1K     |           |
           |  |  |  | | | |  |     Y7|--|--|------------o------>|A7         |
           |  |  +---------->|EN     |--|--|-----------o|------>|           |
           |  |     | | | |  |       |--|--|----------o||------>|           |
           |  |     | | | |  | 259   |--|--|---------o|||------>|           |
           |  |     | | | |  |       |--|--|--------o||||------>|           |
           |  |     | | | |  |       |--|--|-------o|||||------>|           |
           |  |     | | | +--|D      |--|--|------o||||||------>|           |
           |  |     | | +----|A2   Y0|--|--|-----o|||||||------>|A0         |
           |  |     | +------|A1     |  |  |     ||||||||       |           |
           |  |     +--------|A0  RST|  |  |     ||||||||       |   ZIF28   |
           |  |              +-------+  |  |   +------------+   |    per    |
           |  |                    |    |  |   |  dati In   |   |  2816/64  |
           |  |                   +5    |  +-->|/OE         |   |           |
           |  |                         |      |    574     |   |           |
           |  +------------------------------->|CLK         |   |           |
           |                            |      |  dati Out  |   |           |
           |                            V      +------------+   |           |
           +----+              +------------+     ||||||||      |           |
                |              |       SEL  |     ||||||||      |           |
                |              |          B3|<----|||||||o------|D7         |
11 o---<-----------------------|Y3        B2|<----||||||o-------|           |
12 o---<-----------------------|Y2        B1|<----|||||o--------|           |
13 o---<-----------------------|Y1  157   B0|<----||||o---------|           |
15 o---<-----------------------|Y0        A3|<----|||o----------|           |
                |              |          A2|<----||o--- bus ---|           |
                |              |          A1|<----|o---- dati --|           |
                |       GND----|/OE       A0|<----o-------------|D0         |
 +5--o--+       |              +------------+                   |           |
     |  |  __   o---------------------------------------------->|/CE        |
   100K +-|  \  |  __                                           +-----------+
 sw1 |    |   O-o-|  \ 1/2 74HCT132
 o-->o----|__/    |   O---390ohm--+
 |   |          +-|__/            |
 |  --- 1uF     |                LED
 |  ---     +5--+                 |
 |   |                            |
 +---o----------------------------o- GND
```

Note:

1. I numeri dei pin a sinistra sono quelli di una porta parallela con connettore DB25.
2. I numeri dei pin nel blocco a destra sono quelli dello _zoccolo ZIF28_, non del circuito integrato.
3. La 2816 va inserita nella parte bassa dello zoccolo ZIF28.

4. J1 e J2 sono ponticelli a tre contatti in linea. Per entrambi, per usare una 2816 vanno collegati i due contatti a sinistra, per la 2864 invece i due a destra.
5. Nel blocco dello zoccolo ZIF28, le etichette dei segnali per la 2864 (quando diverse da quelle della 2816) sono indicate fra parentesi. Per esempio, il pin 23 dello ZIF 28 ha l'etichetta "23 /WE(A11)" e quindi indica /WE per la 2816 e A11 per la 2864.
6. Per far funzionare il programmatore l'interruttore sw1 deve essere aperto.

6 Programmatori EPROM e chip di memoria

In questo capitolo sono date delle informazioni sulle EPROM e sui chip di memoria.

6.1 Chip di memoria non volatile

Segue una breve descrizione dei chip di memoria e un elenco dei tipi esistenti.

- **PROM**: Pronuncia prom, acronimo di programmable read-only memory (memoria a sola lettura programmabile). Una PROM è un chip di memoria in cui i dati possono essere scritti una sola volta. Una volta che un programma è stato scritto in una PROM resta lì per sempre. Al contrario delle RAM, le PROM mantengono il loro contenuto anche quando il computer viene spento. La differenza fra una PROM e una ROM (read only memory - memoria a sola lettura) è che la PROM viene costruita come memoria vuota, mentre la ROM viene programmata durante il processo di fabbricazione. Per scrivere dei dati in una PROM è necessario usare uno speciale dispositivo chiamato programmatore PROM o scrittore PROM. Il processo di programmazione di una PROM è anche indicato con il termine "bruciare la PROM". Una EPROM (erasable programmable read only memory) è un tipo particolare di PROM che può essere cancellata esponendola a luce ultravioletta. Una volta cancellata può essere riprogrammata. Una EEPROM è simile ad una PROM, ma necessita solo di elettricità per essere cancellata.
- **EPROM**: Acronimo di erasable programmable read-only memory (memoria a sola lettura programmabile e cancellabile), si pronuncia èprom. La EPROM è un tipo particolare di memoria che mantiene il suo contenuto finché non viene esposta a luce ultravioletta. La luce ultravioletta ne cancella il contenuto, rendendo possibile riprogrammare la memoria. Per scrivere e cancellare una EPROM serve un particolare dispositivo chiamato programmatore PROM o scrittore PROM. Una EPROM differisce da una PROM nel fatto che una PROM può venire scritta una sola volta e non può essere cancellata. Le EPROM sono ampiamente diffuse nei personal computer perché consentono al produttore di cambiare i contenuti delle EPROM prima che il computer venga effettivamente consegnato. Ciò significa che è possibile eliminare dei bug e possono essere installate delle nuove versioni subito prima della consegna.

  Una nota riguardo la tecnologia EPROM: i bit di una EPROM vengono programmati iniettando degli elettroni aventi elevata tensione nei gate flottanti dei transistor ad effetto di campo in cui si vuole un bit a 0. Gli elettroni così intrappolati forzano la conduzione del transistor e fanno leggere uno 0. Per cancellare la EPROM si fornisce agli elettroni abbastanza energia per fuggire dal gate flottante, cosa che si ottiene bombardando il chip con radiazione ultravioletta attraverso la finestrella di quarzo. Per impedire la lenta cancellazione, della durata di anni, ad opera della luce del sole e di luci fluorescenti, la finestrella di quarzo, durante l'uso normale, viene coperta con una etichetta opaca.
- **EEPROM**: Acronimo di electrically erasable programmable read-only memory (memoria a sola lettura programmabile e cancellabile elettricamente). Si pronuncia eèprom o èèprom. La EEPROM è un particolare tipo di PROM che può essere cancellata esponendola ad una carica elettrica. Come gli altri

tipi di PROM, la EEPROM mantiene il suo contenuto anche se viene tolta l'alimentazione. Così come gli altri tipi di ROM, la EEPROM non è veloce quanto la RAM. La EEPROM è simile alle memorie flash (a volte chiamate flash EEPROM). La differenza fondamentale è che nella EEPROM i dati vanno scritti un byte alla volta, mentre le meorie flash consentono la scrittura e la cancellazione in blocchi; ciò rende le memorie flash più veloci.

- **FRAM**: Abbreviazione di Ferroelectric Random Access Memory (memoria ad accesso casuale ferroelettrica), un tipo di memoria non volatile sviluppata dalla Ramtron International Corporation. La FRAM combina la velocità di accesso delle DRAM e SRAM con la non volatilità delle ROM. A causa della sua velocità sta rimpiazzando le EEPROM in parecchi dispositivi. Il termine FRAM è un marchio registrato dalla Ramtron.
- **NVRAM**: Abbreviazione di Non-Volatile Random Access Memory (memoria non volatile ad accesso casuale), è un tipo di memoria che mantiene il suo contenuto quando viene tolta l'alimentazione. Un tipo di NVRAM è la SRAM quando viene resa non volatile collegandola ad una fonte di alimentazione continua tipo una batteria. Un altro tipo di NVRAM usa i chip EEPROM per salvare il suo contenuto quando viene tolta l'alimentazione. In tal caso la NVRAM è composta da una combinazione di chip SRAM e EEPROM.
- **Memoria a bolle**: È un tipo di memoria non volatile costituita da un sottile strato di materiale che può essere facilmente magnetizzato in una sola direzione. Quando ad un'area circolare di questa sostanza viene applicato un campo magnetico che non ha la stessa direzione, l'area si riduce ad un piccolo cerchietto, o bolla. Una volta in molti ritenevano che le memorie a bolle sarebbero diventate una tecnologia cardine fra le tecnologie di memorizzazione, ma tali promesse non sono state mantenute. Altri tipi di memorie non volatili, come le EEPROM, sono più veloci e meno costose delle memorie a bolle.
- **Memorie Flash**: Sono un tipo particolare di EEPROM che possono essere cancellate e riprogrammate a blocchi piuttosto che un byte alla volta. Parecchi PC moderni hanno il loro BIOS memorizzato su un chip di memoria flash, in modo da poter essere facilmente aggiornato, se necessario. Un tale BIOS è anche chiamato flash BIOS. Le memoria flash sono diffuse anche nei modem, perché permettono al costruttore di modem di supportare nuovi protocolli non appena vengono standardizzati.

6.2 Elenco di produttori di scrittori EEPROM

Per trovare un elenco di **produttori di scrittori EPROM** si visiti il sito Yahoo (NdT: il sito .com, non quello .it) e si vada in economy->company->Hardware->Peripherals->Device programmers.

- L'URL di Yahoo riguardo le EPROM è a `<http://dir.yahoo.com/Business_and_Economy/Companies/Computers/Hardware/Peripherals/Device_Programmers>`
- *Advanced Research Technology B.V* `<http://www.artbv.nl>` - sviluppo, produzione e vendita di strumenti di programmazione elettronica; sviluppo di hardware e software.
- *Advin Systems Inc.* `<http://www.advin.com>` - dispositivi programmatori basati su PC che supportano i tipi di package e le teconologie dei dispositivi più recenti.
- *Andromeda Research Labs* `<http://www.arlabs.com>` - produce un sistema portatile per la programmazione di eprom e dispositivi.
- *B and C Microsystems, Inc* `<http://www.bcmicro.com>` - offre strumenti di verifica e duplicazione/programmazione di schede PCMCIA (PC), schede ISA/PCI, SIMM, memorie (incluse le FLASH), PLD.
- *BP Microsystems* `<http://www.bpmicro.com>` - programmatori di dispositivi.

- *Bytek* `<http://www.bytek.com>` - progetto, sviluppo, produzione e vendita di sistemi basati su microprocessori, sistemi di elettronica modulare usati per programmare e verificare dispositivi a semiconduttori. La linea di prodotti comprende il ChipBurner.
- *Concentrated Programming Ltd* `<http://www.logicaldevices.com>` - offre un ampio spettro di soluzioni per la programmazione dei dispositivi.
- *Dataman Programmmers Ltd.* `<http://www.dataman.com>` - produce programmatori/emulatori hand-help di EPROM. Vende anche programmatori basati su PC e programmatori Gang-Pro.
- *General Device Instruments* `<http://www.generaldevice.com>` - Programmatori di dispostivi a CI. Programmatori universali e Gang per Pld, flash, microcontrollori, Prom, EEProm, memorie, Epld, Mach e molti altri dispositivi a CI.
- *HI-LO System Research Co., Ltd.* `<http://www.hilosystem.com.tw>` - produttore di programmatori universali e gang.
- *ICE Technology* `<http://www.icetech.com>` - programmatori EPROM e di dispositivi universali che supportano memorie, microcontrollori e dispositivi logici programmabili.
- *Iceprom* `<http://www.inabyte.com/iceprom.html>` - memorie a sola lettura programmabili e cancellabili in-circuit.
- *Incept Ltd* `<http://www.incept.ie>`.
- *International Microsystems Inc* `<http://www.imtest.com>` - Programmatori gang affidabili ad elevata velocità (PROM, FLASH, microcontrollori, schede di memoria PCMCIA).
- *JED Microprocessors Pty. Ltd.* `<http://www.jedmicro.com.au>` - inseritelo in una porta D25 per stampante e programmate qualsiasi dispositivo FLASH ed EPROM a 28 o 32 pin.
- *Logical Devices, Inc* `<http://www.logicaldevices.com>` - dispositivi di programmazione di PLD, FPGA, PROM, microcontrollori. Produttore di compilatori CUPL per logiche programmabili e dei programmatori ALLPRO e Chipmaster.
- *Systems* `<http://www.mcl.dk>` - nuovo metodo non solo di programmazione ma anche di sviluppo di nuovo hardware con unità di controllo integrata. E non serve essere esperti.
- *MQP Electronics* `<http://www.mqp.com>` - produttore di programmatori di dispositivi universali, programmatori gang, software di produzione e convertitori di package. Elevata efficenza ed affidabilità.
- *Needham's Electronics* `<http://www.quiknet.com/~needhams/>` - produttore di programmatori di dispositivi.
- *NP Programming Services* `<http://www.npps.com>` - fornisce programmazione di memorie e parti logiche.
- *Program Automation, Inc.* `<http://www.progauto.com>` - compagnia di servizio indipendente specializzata nella programmazione di grossi volumi di PROM, inclusi CI flash.
- *Stag Programmers Inc* `<http://www.stagusa.com>` - produttore di programmatori prom e logici, produzione di strumenti di manipolazione e cancellatori UV.
- *Sunrise Electronics* `<http://www.sunriseelectronics.com>` - programmatori di dispositivi universali, programmatori gang e in-circuit con supporto a vita.
- *System General Co.* `<http://www.sg.com.tw>` - programmatori di dispositivi, scrittori di EPROM e tester di CI.

- *Tribal Microsystems* <`http://www.tribalmicro.com`> - programmatori di dispositivi gang e universali, emulatori di EPROM e 8051, zoccoli per il burn-in e per la produzione.
- *Universal Device Programmers* <`http://www.xeltek.com`>

7 Introduzione al boot via rete e all'Etherboot

Questo capitolo è stato scritto da Ken Yap *ken.yap@acm.org* e spiega come avviare il proprio computer da un programma immagazzinato in una memoria non volatile senza accedere al proprio disco rigido. È una tecnica ideale per gestire e configurare una "fattoria" di macchine linux.

7.1 Cos'è il boot via rete?

Il boot via rete è una vecchia idea; l'idea di base è che il computer abbia un qualche codice di bootstrap nella memoria non volatile, cioè un chip ROM, che gli permetterà di contattare un server per ottenere i file di sistema attraverso un collegamento di rete.

7.2 Come funziona

Perché sia possibile un boot attraverso la rete il computer deve ricevere:

1. un'identità,
2. un'immagine del sistema operativo e,
3. di solito, un file system funzionante.

Consideriamo un computer diskless (DC) che abbia una ROM per il boot via rete. Potrebbe essere uno di vari DC identici, come possiamo distinguere questo computer dagli altri? C'è una informazione che è unica per quel computer (in realtà per il suo adattatore di rete) ed è il suo indirizzo Ethernet. Ogni adattatore Ethernet nel mondo ha un indirizzo Ethernet univoco di 48 bit; ciò perché ad ogni costruttore di hardware Ethernet sono stati assegnati dei blocchi di indirizzi. Per convenzione tali indirizzi sono scritti con cifre esadecimali, con i due punti che separano ogni gruppo di due cifre; ad esempio: **00:60:08:C7:A3:D8**.

I protocolli usati per ottenere un indirizzo IP, dato un indirizzo Ethernet, sono chiamati **Boot Protocol (BOOTP)** e **Dynamic Host Configuration Protocol** (**DHCP**, protocollo di configurazione dinamica dell'host). DHCP è una evoluzione di BOOTP. Nella nostra discussione, salvo ove diversamente specificato, tutto ciò che si applica a BOOTP vale anche per DHCP (in realtà dire che BOOTP e DHCP traducono solo indirizzi Ethernet è una piccola bugia; nella loro lungimiranza, i progettisti hanno dato a BOOTP e DHCP la possibilità di funzionare con qualsiasi tipo di indirizzo hardware, ma Ethernet è quello che verrà usato nella maggioranza dei casi).

Un esempio di scambio BOOTP funziona così:

DC: Salve, il mio indirizzo hardware è **00:60:08:C7:A3:D8**, per favore dammi il mio indirizzo IP.

server BOOTP: (guardando nel database degli indirizzi) Il tuo nome è aldebaran, il tuo indirizzo IP è 192.168.1.100, il tuo server è 192.168.1.1, il file da cui effettuare il boot è /tftpboot/vmlinux.nb (più qualche altra informazione).

Ci si potrebbe chiedere come ha fatto il DC a trovare l'indirizzo del server BOOTP la prima volta. La risposta è che non l'ha fatto; la richiesta BOOTP è stata diffusa in broadcast sulla rete locale e qualsiasi server BOOTP in grado di rispondere lo ha fatto.

Dopo aver ottenuto un indirizzo IP, il DC deve scaricare l'immagine di un sistema operativo ed eseguirlo. Qui viene usato un altro protocollo Internet chiamato **Trivial File Transfer Protocol** (**TFTP** - protocollo elementare per il trasferimento file). Il TFTP è una specie di versione ridotta dell'FTP — non c'è autenticazione e funziona sullo User Datagram Protocol (UDP) invece che sul Transmission Control Protocol (TCP). È stato scelto l'UDP al posto del TCP per via della sua semplicità; l'implementazione dell'UDP sul DC può essere abbastanza piccola da riuscire facilmente a farne entrare il codice in una ROM. Visto che l'UDP è un codice orientato ai blocchi, al contrario di uno orientato al protocollo, il trasferimento avviene blocco per blocco, in questo modo:

```
DC: Dammi il blocco 1 di /tftpboot/vmlinux.nb.
server TFTP: Eccolo.
DC: Dammi il blocco 2.
```

...e così via, finché non viene trasferito l'intero file. L'handshaking è basato su una semplice conferma di ogni blocco e la perdita di pacchetti è gestita ritrasmettendo dopo un timeout. Quando sono stati ricevuti tutti i pacchetti la ROM di boot della rete passa il controllo all'entry point dell'immagine del sistema operativo.

Concludendo, per poter eseguire un sistema operativo deve essere fornito un filesystem principale. Il protocollo usato da Linux e da altri Unix di solito è il **Network File System** (**NFS** - file system di rete), sebbene siano possibili altre scelte. In questo caso il codice non deve risiedere nella ROM ma può essere parte del sistema operativo che si è appena scaricato. Comunque il sistema operativo deve essere capace di girare con un file system principale che sia un NFS invece che un vero disco. Linux ha le variabili di configurazione necessarie per compilare una versione che possa farlo.

7.3 Il boot via rete in pratica

Il Net Loader è un programmino che gira come estensione del BIOS, di solito su una EPROM nel NIC. Esso gestisce le richieste BOOTP e i caricamenti TFTP e poi trasferisce il controllo all'immagine caricata. Usa i protocolli TCP/IP ma l'immagine caricata non deve necessariamente essere Linux; può essere qualunque cosa, anche DOS. Possono anche essere caricate da un floppy per effettuare delle prove o per provare impostazioni temporanee.

Oltre alle ROM di boot commerciali ci sono **DUE** fonti di pacchetti per il boot via rete. Implementazioni libere (free) di loader per reti TCP/IP sono:

1. **ETHERBOOT** `<http://www.slug.org.au/etherboot>` e
2. **NETBOOT** `<http://www.han.de/~gero/netboot.html>`

Etherboot usa driver interni, Etherboot invece usa packet driver. Innanzi tutto bisogna accertarsi che la propria scheda di rete sia supportata da Etherboot o da Netboot. Eventualmente bisognerà trovare una persona disposta a mettere il codice in una EPROM (Erasable Programmable Read Only Memory) al posto proprio, ma all'inizio si può effettuare il **boot via rete da un floppy**.

Per creare un floppy di boot nella distribuzione viene fornito uno speciale blocco di boot. Questo piccolo programma di 512 byte carica in memoria i blocchi del disco che lo seguono nel floppy e inizia l'esecuzione. Quindi, per fare un floppy di boot, uno deve solo concatenare il blocco di boot con il binario Etherboot contenente il driver per una scheda di rete, in questo modo:

```
# cat floppyload.bin 3c509.lzrom > /dev/fd0
```

Si prenda il pacchetto nfdboot (disponibile presso il proprio sito mirror Linux preferito nella directory /pub/Linux/system/Linux-boot). Esso contiene una immagine bootprom per le schede di rete (come la wd8013) che può essere scritta direttamente. Vedere anche il sito LTSP presso `<http://www.ltsp.org>`.

Prima di inserire il floppy per il boot via rete bisogna preparare tre servizi su Linux:

1. BOOTP (o DHCP)
2. TFTP e
3. NFS.

Non è necessario prepararli tutti e tre contemporaneamente, si può farlo un passo alla volta, accertandosi che ogni passo funzioni prima di passare al successivo.

7.3.1 Bootp

Si installi Bootp. Vedere bootp*.rpm sul cdrom Redhat Linux. Vedere anche i pacchetti RPM sul sito LTSP presso `<http://www.ltsp.org>`. Vedere anche le pagine di manuale unix 'man 5 bootptab', 'man 8 bootpd', 'man 8 bootpef', 'man 8 bootptest'. Bisogna poi assicurarsi che il server stia aspettando richieste bootp. Il demone può essere lanciato direttamente col comando

```
bootpd -s
```

oppure usando inetd; si editi il file /etc/inetd.conf e si inserisca una riga come la seguente:

```
bootps dgram   udp     wait    root    /usr/sbin/in.bootpd    bootpd
```

Inserire o togliere il commento dalle seguenti due righe in /etc/services:

```
bootps          67/tcp          # server BOOTP
tftp            69/udp          # server TFTP
```

Se si è dovuto modificare /etc/inetd.conf allora è necessario far ripartire inetd inviando al processo il segnale HUP.

```
kill -HUP <id del processo inetd>
```

Poi, bisogna dare a bootp un database per mappare gli indirizzi Ethernet con indirizzi IP. Questo database è in /etc/bootptab. Bisogna modificarlo inserendo gli indirizzi IP del proprio gateway, del server dns e gli indirizzi ethernet delle proprie macchine diskless. Il database contiene righe aventi la forma:

```
aldebaran.foo.com:ha=006008C7A3D8:ip=192.168.1.100:bf=/tftpboot/vmlinuz.nb
```

Possono essere specificate altre informazioni, ma cominceremo nel modo facile.

Un altro esempio di /etc/bootptab è:

```
global.prof:\
        :sm=255.255.255.0:\
        :ds=192.168.1.5:\
        :gw=192.168.1.19:\
        :ht=ethernet:\
        :bf=linux:
machine1:hd=/export/root/machine1:tc=global.prof:ha=0000c0863d7a:ip=192.168.1.140:
machine2:hd=/export/root/machine2:tc=global.prof:ha=0800110244e1:ip=192.168.1.141:
machine3:hd=/export/root/machine3:tc=global.prof:ha=0800110244de:ip=192.168.1.142:
```

global.prof è una generica traccia per le voci di host, in cui:

- il campo sm contiene la maschera di sottorete
- il campo ds contiene l'indirizzo del Domain Name Server
- il campo gw contiene l'indirizzo predefinito del gateway
- il campo ht contiene il tipo di hardware del supporto di rete
- il campo bf contiene il nome del file di boot

Dopo di che, ogni macchina deve avere una riga in cui:

- il primo campo contiene il nome dell'host
- il campo hd contiene la directory del file di boot
- la traccia globale può essere inclusa nel campo tc
- il campo ha contiene l'indirizzo hardware della scheda ethernet
- il campo ip contiene l'indirizzo ip assegnato

Ora si faccia il boot del DC col floppy e questi dovrebbe rilevare la propria scheda Ethernet e diffondere una richiesta BOOTP. Se tutto va bene il server dovrebbe rispondere al DC con l'informazione richiesta. Poiché /tftpboot/vmlinux.nb ancora non esiste, quando proverà a caricare il file fallirà. Ora occorre compilare un kernel speciale, uno in cui sia abilitata l'opzione per montare il filesystem principale dal NFS. Bisogna anche abilitare l'opzione per prendere l'indirizzo IP del kernel dalla risposta BOOTP originale. Si deve compilare il driver Linux per il proprio adattatore di rete dentro al kernel, invece che caricarlo come modulo. È possibile scaricare un ramdisk iniziale di modo che il caricamento dei moduli funzioni, ma questo è qualcosa che si può fare in seguito.

Non si può installare direttamente la zImage risultante dalla compilazione del kernel, deve essere prima convertita in una immagine marcata. Una immagine marcata è una normale immagine del kernel con un header speciale che dice al bootloader di rete dove vanno i byte nella memoria e da quale indirizzo iniziare il programma. Per creare tale immagine marcata va usato un programma chiamato mknbi-linux. Questa utility si trova nella distribuzione Etherboot. Dopo che è stata generata l'immagine la si metta nella directory /tftpboot col nome specificato in /etc/bootptab. Si renda tale file leggibile da tutti perché il server tftp non ha privilegi speciali.

7.3.2 Tftp

Riguardo TFTP, vedere tftp*.rpm sul cdrom Redhat Linux. Il TFTP (Trivial File Transfer Protocol) è un protocollo per il trasferimento di file, come ftp ma molto più semplice, il che è d'aiuto per codificarlo nelle EPROM. TFTP può essere usato in due modi:

- **tftp semplice:** significa che il client può accedere al vostro intero file system. È il più semplice ma è anche un grosso buco di sicurezza (chiunque può prendere il vostro file con le password via tftp).
- **tftp sicuro:** il server tftp usa una chiamata di sistema chroot.2 per cambiare la sua directory di root. Qualsiasi cosa al di fuori della nuova directory di root sarà completamente inaccessibile. Visto che la directory chroot diventa la nuova directory di root, l'hd indicato in bootptab deve riflettere la nuova situazione. Ad esempio: quando si usa il tftp insicuro, il campo hd conterrà il path completo per la directory di boot, cioè /export/root/machine1. Quando si usa il tftp sicuro con /export come directory di root, allora /export diventa / e il campo hd dovrà essere /root/machine1.

Tftpd viene di solito avviato da inetd con una riga in /etc/inetd.conf simile alla seguente:

```
tftp dgram udp wait root /usr/sbin/tcpd in.tftpd -s /tftpboot
#tftp   dgram   udp     wait    root    /usr/sbin/in.tftpd     tftpd /export
```

Di nuovo, si riavvii inetd con un segnale HUP e si riprovi il boot; stavolta dovrebbe scaricare l'immagine del kernel ed eseguirla. Si otterrà che il boot va avanti fino al punto in cui prova a montare un filesystem principale. A questo punto per procedere bisogna configurare ed esportare le partizioni NFS.

7.3.3 Filesystem principale NFS

Per vari motivi, non è una buona idea usare il filesystem principale del server come filesystem principale dei DC. Il primo semplicemente è che lì ci sono diversi file di configurazione e in quel modo il DC prenderà l'informazione sbagliata. Un altro è la sicurezza; è pericoloso permettere l'accesso in scrittura alla root del server (e, per diversi motivi, l'accesso in scrittura al filesystem principale è indispensabile). Comunque la buona notizia è che un filesystem principale per il DC non è molto grande, solo 30 MB circa, e buona parte di esso può essere condiviso fra più DC.

Idealmente, per costruire un filesystem principale bisogna sapere quali file si aspetta di vedere lì la vostra distribuzione del sistema operativo. Per il boot sono cruciali i file di device, quelli in /sbin e in /etc. È possibile evitare gran parte del lavoraccio copiando un filesystem principale esistente e modificando alcuni file per il DC. Nella distribuzione Etherboot c'è un tutorial ed alcuni link ad un paio di script di shell che servono per creare un tale filesystem principale per il DC partendo da un filesystem principale esistente su un server. Nella documentazione di Etherboot ci sono anche dei suggerimenti su come risolvere i problemi, visto che questa è spesso la parte più insidiosa della preparazione.

Il kernel Linux fatto su misura per il DC si aspetta di vedere il filesystem principale in /tftpboot/(indirizzo IP del DC), ad esempio: /tftpboot/192.168.1.100 nel suddetto caso. Ciò, volendo, può essere modificato durante la configurazione del kernel.

Ora si crei, o si modifichi, /etc/exports sul server e si aggiunga una riga così fatta:

```
/tftpboot/192.168.1.100 aldebaran.foo.com(rw,no_root_squash)
```

L'accesso rw è necessario per vari servizi di sistema. L'attributo no_root_squash impedisce al sistema NFS di mappare l'ID del root su un altro. Se questi non è specificato vari demoni e generatori di log non saranno contenti.

Si avviino, o riavviino, i servizi NFS (rpc.portmap e rpc.mountd) e si riprovi il boot senza dischi. Se va tutto bene il kernel dovrebbe essere in grado di montare un filesystem principale e proseguire il boot fino al prompt di login. Più probabilmente ci si accorgerà che ci sono diverse cose non configurate a dovere. La maggior parte delle distribuzioni Linux sono orientate verso le operazioni su disco e necessitano di un po' di modifiche per andar bene per un boot diskless. La più comune causa di fallimento è dovuta al fatto che il processo di boot fa affidamento ai file sotto /usr, che di solito viene importata da un server più tardi nel processo di boot. Due possibili soluzioni sono:

1. Fornire i pochi file richiesti sotto una piccola directory /usr sul filesystem principale, che verrà poi sovrascritta quando verrà importato /usr;
2. Modificare il path in modo che cerchi i file nel filesystem principale. I file da modificare sono sotto /tftpboot/192.168.1.100 (si ricordi che questa è la directory principale del DC).

Si potrebbero voler montare altre directory dal server, come /usr (che può essere esportata in sola lettura).

7.3.4 Scrivere la EPROM

Quando si è riusciti ad effettuare il boot dalla rete senza nessun problema, si potrebbe voler mettere il codice in una EPROM.

7.4 Utilizzi del boot via rete

I terminali X sono un naturale utilizzo del boot via rete. La mancanza di dischi sul terminale lo rende più silenzioso e contribuisce a rendere piacevole l'ambiente di lavoro. La macchina idealmente dovrebbe avere 16MB o più di memoria e la miglior scheda video che si riesce a trovargli. Questo è l'utilizzo ideale per un 486 di fascia alta, o un Pentium di fascia bassa, che è diventato obsoleto a causa dell'evoluzione dell'hardware. Altri usano il boot via rete per cluster di macchine in cui l'uso del DC è leggero e non giustifica la presenza di un disco; ad esempio un cluster di macchine in un'aula scolastica.

7.5 Per maggiori informazioni

Il primo posto in cui fermarsi è la home page di Etherboot: `<http://www.slug.org.au/etherboot>`

Lì si troveranno link ad altre risorse, inclusa una mailing list a cui iscriversi, in cui vengono discussi problemi e soluzioni.

Documenti correlati:

- NFS-root Mini Howto, in /usr/doc/HOWTO/mini o sul cdrom Linux.
- Linux Networking-HOWTO di Terry Dawson, in /usr/doc/HOWTO o sul cdrom Linux *94004531@postoffice.csu.edu.au* `<mailto:94004531@postoffice.csu.edu.au>`
- NET-3-Howto, in /usr/doc/HOWTO o sul cdrom Linux.
- /usr/src/linux/README riguardo la configurazione e la compilazione di nuovi kernel.

8 Configurazione del Linux Redhat

Il DC richiede di montare /tftpboot/<*indirizzo IP del DC*> (nella 2.1 e precedenti: /tftpboot/<*nome del DC in bootptab*>) come sua '/' tramite NFS dal server. Bisogna esportarlo dal server (rw, no_root_squash) perché il DC ci vuole scrivere (file di log, ecc).

La directory di root / deve contenere /sbin, /lib, /etc, /var, /tmp, /root, /dev e /proc.

/sbin, /bin e /lib possono essere una copia di un sistema Linux Redhat esistente. Possono essere condivise tra tutti i DC. Ma solo link fisici. A proposito, non si facciano link agli originali sul server.

/etc, /var e /dev devono essere copie non condivisibili. Si aggiustino ad hoc /etc/sysconfig/network, /etc/sysconfig/network-scripts/ifcfg-eth0, /etc/fstab, /etc/conf.modules e altri. Disabilitare tutti i servizi di rete che non servono. Togliere da /var tutta la roba di cui non si ha bisogno, ad es. i db RPM e i file lpd.

/root e /proc dovrebbero semplicemente esistere. /tmp dovrebbe esistere ed essere mode 1777.

Probabilmente si vorrà creare i mount point /usr e /home. /usr può essere montato ro (read only - a sola lettura).

Dovrebbero bastare circa 10 MB per DC più circa 15 MB per i file condivisi. A proposito: se i propri DC sono identici anche l'immagine del kernel può essere condivisa.

Ecco uno script illustrativo per creare il primo filesystem root:

```
#!/bin/sh
if [ $# != 1 ]
then
        echo Usage: $0 client-IP-addr
        exit 1
fi

cd /

umask 022

mkdir -p /tftpboot/$1

# facciamo così
for d in home mnt proc tmp usr
do
        mkdir /tftpboot/$1/$d
        done

        chmod 1777 /tftpboot/$1/tmp

        touch /tftpboot/$1/fastboot
        chattr +i /tftpboot/$1/fastboot

        # copiamo questi
        cp -a bin lib sbin dev etc root var /tftpboot/$1

cat <<EOF
Ora, in /tftpboot/$1/etc, editare
```

```
                sysconfig/network
                sysconfig/network-scripts/ifcfg-eth0
                fstab
                conf.modules

e configurare

                rc.d/rc3.d
EOF
```

Ecco uno script illustrativo per duplicare il filesystem root:

```
#!/bin/sh
if [ $# != 2 ]
then
        echo Usage: $0 olddir newdir
        exit 1
fi

cd /tftpboot

if [ ! -d $1 ]
then
        echo $1 is not a directory
        exit 1
fi

umask 022

mkdir -p $2

# facciamo così
for d in home mnt proc tmp usr
do
        mkdir $2/$d
done

chmod 1777 $2/tmp

touch $2/fastboot
chattr +i $2/fastboot

# linkiamo questi
for d in bin lib sbin
do
        (cd $1; find $d -print | cpio -pl ../$2)
done

# copiamo questi
for d in dev etc root var
do
```

```
        cp -a $1/$d $2
done

cat <<EOF
Ora, in /tftpboot/$2/etc, editare

        sysconfig/network
        sysconfig/network-scripts/ifcfg-eth0
        fstab (forse)
        conf.modules (forse)

e configurare

        rc.d/rc3.d
EOF
```

8.1 X-terminal

Sul server, ci si assicuri che il DC corrisponda ad un'apposita voce in /etc/X11/xdm/Xaccess e si commenti il :0 in /etc/X11/xdm/Xservers. Poi ci si assicuri che xdm venga eseguito dagli script di init.

Sul client, eseguire X -query server

Si otterrà la finestra di login di xdm, dopo di che tutti i propri client X gireranno sul server.

Riguardo l'uso di altre applicazioni, si potrebbe usare la tecnica diskless per router netboot, server di stampa (ma non devono essere server di stampa in spooling), applicazioni standalone, ecc.

9 PROM LanWorks BootWare

Questa informazione può far risparmiare del tempo. Per fare in modo che le PROM LanWorks BootWare(tm) facciano partire correttamente un'immagine del kernel Linux, la parte "bootsector" dell'immagine deve essere modificata in modo tale da permettere alle prom di boot di saltare dritto all'indirizzo iniziale dell'immagine. Il formato dell'immagine avviabile via rete creato dal programma 'mknbi-linux' di netboot/etherboot è diverso e non girerà se usato con le PROM BootWare.

Un bootsector modificato ed un Makefile utili per creare un'immagine avviabile da BootWare, fatta la compilazione del kernel, può essere trovata presso:

- Il pacchetto Bwimage `<ftp://ftp.ipp.mpg.de/pub/ipp/wls/linux/bwimage-0.1.tgz>`
- Vedere anche `<http://www.patoche.org/LTT/net/00000096.html>`
- ROM di boot LanWorks BootWare `<http://www.3com.com/lanworks>`

Si faccia riferimento al file README per i dettagli riguardanti l'installazione. Al momento sono supportati solo i kernel di tipo "zImage". Sfortunatamente i parametri del kernel vengono ignorati.

Per questa sezione si ringrazia Jochen Kmietsch; per qualsiasi domanda spedire una email a: *jochen.kmietsch@tu-clausthal.de* `<mailto:jochen.kmietsch@tu-clausthal.de>`

10 Etherboot

Etherboot è un pacchetto per creare immagini ROM che possono scaricare, attraverso la rete, codice che va eseguito su un computer x86. Tipicamente il computer sarà senza dischi e il codice sarà Linux, ma queste non sono le uniche possibilità.

Questo documento si trova presso la *Etherboot Home Page* `<http://www.slug.org.au/etherboot/>`. Spiega come installare, configurare e usare il pacchetto Etherboot.

11 Netboot

Netboot è stato scritto da Zurück zu Gero. Il sito principale è `<http://www.han.de/~gero/netboot.html>`.

11.1 Introduzione

Il seguente elenco mostra solo qualche esempio dei possibili usi di Netboot:

- Spooler di stampa
- Server per terminali
- Terminale X11
- Sistema per il log dei dati
- Network-Computer (NC)
- Altro ancora ...

Per trovare l'immagine del kernel la bootrom usa il protocollo BOOTP, così come definito nelle 15 (), per recuperare le informazioni necessarie per il boot e poi carica l'effettiva immagine usando il protocollo TFTP, definito nella 15 ().

Le specifiche dettagliate per il processo di boot via rete possono essere trovate in `<http://www.han.de/~gero/netboot/english/spec.html>`.

11.2 Mailing list

Esiste una mailing list dedicata al boot via rete. Per iscriversi basta semplicemente preparare una email contenente la riga

subscribe netboot

nel corpo del messaggio e spedirla a *majordomo@baghira.han.de* `<mailto:majordomo@baghira.han.de>`

L'oggetto della mail non è importante. Dopo la sottoscrizione, è possibile inviare messaggi alla lista scrivendo a netboot@baghira.han.de.

11.3 Netboot, link utili

L'archivio della mailing list di Netboot è presso `<http://www.han.de/~gero/netboot/archive/maillist.html>`

- I driver 3com sono presso `<http://support.3com.com/infodeli/tools/nic>`

- I driver Accton sono *qui* `<http://www.accton.com/accton/drivers/adapter.html>`
- *Artisoft* `<http://www.artisoft.com>`
- *CNET* `<http://www.cnet.com.tw>`
- *Compaq* `<http://www.compaq.com/support/networking>`
- *D-Link* `<http://www.dlink.com>`
- *Microdyne* `<http://www.mcdy.com/marketin/prodman/prodcat.htm>`
- Parecchie schede PCI NE2000 sono basate sui chipset Realtek. Si prendano i driver *qui* `<http://www.realtek.com.tw/cn/driver/driver.htm>`
- *Standard Microsystems Corp* `<http://www.smc.com/support.html>`
- *Surecom* `<http://www.sure-com.net>`
- *Thomas Conrad corp* `<http://www.compaq.com/support/networking/OutOfProduction.html>`
- *Winbond* `<http://www.com.tw>`
- *Xircom* `<http://www.xircom.com>`

- La *pagina Webopaedia* `<http://www.sandybay.com/pc-web/network_interface_card_NIC.htm>` sulle schede di rete.
- La *pagina dei driver* `<http://www.evitech.fi/~jarnomn/files/drivers/net_d.html>` di Jargon con parecchi driver per le vecchie schede di rete.
- *Etherboot* `<http://www.slug.org.au/etherboot/>`. Questo è un progetto simile al Netboot ma basato sul codice bootrom BSD.
- Come tirar fuori un *Terminale X Window* `<http://www.menet.umn.edu/~kaszeta/unix/xterminal/index.html>` dal vostro vecchio o obsoleto PC.
- Elenco delle *impostazioni dei ponticelli* `<http://www.slug.org.au/NIC/index.html>` per varie schede di rete. Questa pagina contiene anche parecchi altri buoni link.
- *Freefire* `<http://sites.inka.de/lina/freefire-l/tools.html>` è la home page del progetto Freefire, che elenca molte risorse riguardanti argomenti sulla sicurezza di rete.

12 URL correlati

- Vedere il 'Diskless-root-NFS-HOWTO' presso `<http://metalab.unc.edu/LDP/HOWTO/Diskless-root-NFS-HOWTO.html>`
- Linux goodies `<http://www.aldev.8m.com>` o `<http://www.aldev.webjump.com>`

13 Nota sul copyright

14 Altri formati di questo documento

Questo documento è pubblicato in 11 formati diversi, ossia: DVI, Postscript, Latex, Adobe Acrobat PDF, LyX, GNU-info, HTML, RTF (Rich Text Format), testo semplice, pagine di manuale Unix e SGML.

- È possibile ottenere questo HOWTO come singolo archivio tar nei formati HTML, DVI, Postscript o SGML da `<ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO/other-formats/>`
- Il formato testo semplice è in: `<ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO>`
- Le traduzioni in altre lingue come Francese, Tedesco, Spagnolo, Cinese e Giapponese sono in `<ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO>`. Qualsiasi aiuto da parte vostra per tradurlo in altre lingue è gradito.

Questo documento è stato scritto usando gli "SGML tools" che possono essere presi da: `<http://www.sgmltools.org>`.

Compilando i sorgenti otterrete dei comandi come:

- sgml2html CVS-HOWTO.sgml (per generare file html)
- sgml2rtf CVS-HOWTO.sgml (per generare file RTF)
- sgml2latex CVS-HOWTO.sgml (per generare file latex)

I documenti LaTeX possono essere convertiti in file PDF semplicemente generando una versione in Postscript usando **sgml2latex** (e dvips) e poi rielaborando il risultato col comando **distill** di Acrobat (`<http://www.adobe.com>`), come mostrato sotto:

```
bash$ man sgml2latex
bash$ sgml2latex nomefile.sgml
bash$ man dvips
bash$ dvips -o nomefile.ps nomefile.dvi
bash$ distill nomefile.ps
bash$ man ghostscript
bash$ man ps2pdf
bash$ ps2pdf input.ps output.pdf
bash$ acroread output.pdf &
```

Oppure si può usare il comando **ps2pdf** di Ghostscript. ps2pdf emula quasi tutte le funzionalità del prodotto Acrobat Distiller della Adobe: converte i file PostScript in file Portable Document Format (PDF). **ps2pdf** è realizzato tramite un brevissimo script (un file batch) che invoca Ghostscript selezionando uno speciale "output device" chiamato **pdfwrite**. Per poter usare ps2pdf, il device pdfwrite deve essere stato incluso nel makefile quando Ghostscript è stato compilato. Per i dettagli vedere la documentazione sulla compilazione di Ghostscript.

Questo documento si trova presso:

- <http://sunsite.unc.edu/LDP/HOWTO/CVS-HOWTO.html>

È possibile trovare questo documento anche presso i seguenti siti mirror:

- <http://www.caldera.com/LDP/HOWTO/CVS-HOWTO.html>
- <http://www.WGS.com/LDP/HOWTO/CVS-HOWTO.html>
- <http://www.cc.gatech.edu/linux/LDP/HOWTO/CVS-HOWTO.html>
- <http://www.redhat.com/linux-info/ldp/HOWTO/CVS-HOWTO.html>
- Altri siti mirror più vicini (dal punto di vista dell'indirizzo di rete) possono essere trovati presso <http://sunsite.unc.edu/LDP/hmirrors.html>. Scegliere un sito ed andare direttamente nella directory /LDP/HOWTO/CVS-HOWTO.html

Per poter vedere il documento in formato dvi si usi il programma xdvi. Il programma xdvi si trova nel pacchetto tetex-xdvi*.rpm del Linux RedHat che può essere rintracciato attraverso i pulsanti di menu ControlPanel | Applications | Publishing | TeX. Per leggere il documento dvi usare il comando:

```
xdvi -geometry 80x90 howto.dvi
man xdvi
```

e ridimensionare la finestra col mouse. Per navigare usare i tasti freccia, Pag Su, Pag Giù; è possibile usare anche i tasti 'f', 'd', 'u', 'c', 'l', 'r', 'p' e 'n' per muoversi su, giù, centro, pagina successiva, pagina precedente, ecc. Per eliminare il menu expert premere 'x'.

È possibile leggere i file postscript usando il programma 'gv' (ghostview) o 'ghostscript'. Il programma ghostscript è nel pacchetto ghostscript*.rpm e il programma gv è in quello gv*.rpm di Linux Redhat; possono essere rintracciati attraverso i pulsanti di menu ControlPanel | Applications | Graphics. Il programma gv è molto più amichevole di ghostscript. Ghostscript e gv sono disponibili anche per altre piattaforme come OS/2, Windows 95 e NT, questo documento è visibile anche su tali piattaforme.

- Ghostscript per Windows 95, OS/2 e per tutti i sistemi operativi è disponibile presso <http://www.cs.wisc.edu/~ghost>

Per leggere il documento postscript usare il comando:

```
gv howto.ps
ghostscript howto.ps
```

È possibile leggere il documento in formato HTML usando i browser Netscape Navigator, Microsoft Internet explorer, Redhat Baron Web o uno qualsiasi degli altri 10 web browser.

È possibile leggere il formato latex, l'output di LyX, usando LyX, un front-end di X-Windows per latex.

15 Argomenti per accademie ed università

Questa sezione è di interesse accademico, per università o istituti di ricerca. Potete leggerla, se avete del tempo da perdere. I link seguenti riguardano le RFC e la storia dei nodi diskless. Gli studenti potrebbero trovare interessante leggere in tali link la storia dello sviluppo delle stazioni diskless.

Attenzione: Le informazioni e i dati cui fanno riferimento questi URL potrebbero essere vecchi.

- Le istruzioni sull'installazione si trovano *qui* <http://www.aldev.8m.com/disklesshowto-install.html> - Mirror: <http://aldev.webjump.com/disklesshowto-install.html>
- RFC 951 <http://www.aldev.8m.com/disklesshowto-RFC-951.html> - Mirror: <http://aldev.webjump.com/disklesshowto-RFC-951.html>
- RFC 1350 <http://www.aldev.8m.com/disklesshowto-RFC-1350.html> - Mirror: <http://aldev.webjump.com/disklesshowto-RFC-1350.html>
- RFC 1533 <http://www.aldev.8m.com/disklesshowto-RFC-1533.html> - Mirror: <http://aldev.webjump.com/disklesshowto-RFC-1533.html>
- Risoluzione dei problemi <http://www.aldev.8m.com/disklesshowto-Troubleshoot.html> - Mirror: <http://aldev.webjump.com/disklesshowto-Troubleshoot.html>