

Minix

Procurarsi il software	1987
Pacchetti essenziali	1988
Preparazione all'installazione	1988
Nomi di dispositivo riferiti alle partizioni	1988
Avvio	1989
Installazione	1990
Setup	1991
Avvio del sistema copiato nel disco fisso	1993
Installazione delle serie di dischetti	1993
Dischetto di avvio	1993
Conclusione	1993
Ricompilazione del kernel	1993
/usr/include/minix/config.h	1994
File di dispositivo	1995
Parametri di avvio	1995
Scheda di rete	1995
Configurazione della rete	1996
/etc/hosts	1996
File «/etc/hostname.file»	1996
File «/etc/resolv.conf»	1996
File «/etc/rc.net»	1996
File «/etc/rc»	1996
Personalizzazione	1997
File «/etc/passwd», «/etc/group» e «/etc/shadow»	1997
Tastiera	1997
Modifica della mappa	1998
Altri programmi	1999
Pacchetto «HTTPD.TAZ»	1999
Pacchetto «FROG.TAZ»	2000
Copie di sicurezza	2000
Archiviazione	2000
Recupero	2001
Convivenza tra Minix e GNU/Linux	2001
LILO	2001
Riferimenti	2001

Minix ¹ è nato originariamente come sistema didattico. Minix non ha avuto il successo e la diffusione che avrebbe potuto avere a causa delle limitazioni della sua licenza iniziale. In seguito le cose sono cambiate, fortunatamente, perché Minix resta probabilmente l'unica possibilità reale per chi vuole utilizzare elaboratori con architettura i286 o inferiore.

Lo scopo di questo capitolo è introdurre all'uso di Minix per poter riutilizzare i vecchi elaboratori i286, in particolare, collegandoli a una rete locale TCP/IP. Le informazioni seguenti si riferiscono alla versione 2.0.0.

Procurarsi il software

Minix è ottenibile dalla rete, precisamente a partire dalla sua pagina di presentazione ufficiale, quella del suo primo autore (La sigla «AST» rappresenta le iniziali di Andrew S. Tanenbaum), oltre che dai vari siti speculari del relativo FTP.

<http://www.cs.vu.nl/~ast/minix.html>

Minix è nato assieme a un libro che tuttora dovrebbe essere accompagnato da un CD-ROM contenente il sistema operativo:

- Andrew S. Tanenbaum, Alber S. Woodhull, *Operating Systems: Design and Implementation*, 2/e, Prentice-Hall

Pacchetti essenziali

Minix è un sistema molto piccolo e composto da pochi pacchetti. Questi hanno alcune particolarità: i nomi sono composti con lettere maiuscole e gli archivi compressi utilizzano la combinazione `'tar'+compress'` e sono evidenziati dall'uso dell'estensione `' .TAR'`.

Per prima cosa è necessario riprodurre la coppia di dischetti `'ROOT'` e `'USR'`, a partire dai file omonimi. Il primo è in grado di avviarsi e contiene un file system minimo che si installa in un disco RAM, il secondo contiene una piccola serie di programmi da innestare nella directory `'/usr/'`, che servono per poter installare Minix nel disco fisso. Se si ha poca memoria a disposizione (i classici 640 Kibyte sono il minimo in assoluto per poter fare funzionare Minix), si può evitare l'utilizzo del disco RAM fondendo i due file in un solo dischetto. Data l'intenzione di questo capitolo viene descritta l'ultima di queste modalità di installazione.

Il mini sistema che si ottiene attraverso i due file appena citati, permette di installare, più o meno automaticamente, un insieme minimo di programmi contenuto nell'archivio `'USR.TAR'`

Esistono due versioni di questi tre file: una per architettura i386 o superiore e l'altra per i microprocessori inferiori. Date le intenzioni, si devono utilizzare i file della versione denominata `'i86'`.

Preparazione all'installazione

Il procedimento che viene descritto è valido sia per dischetti da 1440 Kibyte che da 1200 Kibyte. I due file `'ROOT'` e `'USR'` vanno copiati uno di seguito all'altro. Utilizzando GNU/Linux, si può fare nel modo seguente:

```
# cat ROOT USR > /dev/fd0 [Invio]
```

Dal punto di vista di Minix, il dischetto inserito nella prima unità a dischetti corrisponde al dispositivo `'/dev/fd0'`, come per GNU/Linux, ma risulta diviso in partizioni: l'immagine `'ROOT'` risulta essere `'/dev/fd0a'` e l'immagine `'USR'` è `'/dev/fd0c'` (la partizione `'b'` è vuota).²

L'archivio `'USR.TAR'` deve essere suddiviso in diversi dischetti, con un procedimento un po' insolito: viene semplicemente tagliato a fette della dimensione massima contenibile dal tipo di dischetti che si utilizza. Nel caso si tratti di dischetti da 1440 Kibyte, si può utilizzare GNU/Linux, o un altro sistema Unix, nel modo seguente:

```
# dd if=/USR.TAR of=/dev/fd0 bs=1440k count=1 skip=0 [Invio]
```

```
# dd if=/USR.TAR of=/dev/fd0 bs=1440k count=1 skip=1 [Invio]
```

```
# dd if=/USR.TAR of=/dev/fd0 bs=1440k count=1 skip=2 [Invio]
```

Se si trattasse di dischetti da 1200 Kibyte, occorrerebbe modificare la dimensione del blocco, come nell'esempio seguente:

```
# dd if=/USR.TAR of=/dev/fd0 bs=1200k count=1 skip=0 [Invio]
```

```
# dd if=/USR.TAR of=/dev/fd0 bs=1200k count=1 skip=1 [Invio]
```

```
# dd if=/USR.TAR of=/dev/fd0 bs=1200k count=1 skip=2 [Invio]
```

Nomi di dispositivo riferiti alle partizioni

Minix viene installato normalmente all'interno di una partizione primaria suddivisa in almeno due partizioni secondarie. La prima partizione serve a contenere il file system principale ed è di piccole dimensioni: 1440 Kibyte. La seconda serve per tutto il resto e viene innestata in corrispondenza della directory `'/usr/'`. Inizialmente le partizioni erano tre e `'/usr/'` era la terza. Attualmente, `'/usr/'` continua a essere la terza partizione e si finge che esista una seconda partizione senza alcuno spazio a disposizione.

Il primo disco fisso viene identificato dal dispositivo `'/dev/hd0'`, il secondo da `'/dev/hd5'`. Le partizioni primarie del primo disco fisso vanno da `'/dev/hd1'` a `'/dev/hd4'`; quelle del secondo disco fisso da `'/dev/hd6'` a `'/dev/hd9'`. Le partizioni secondarie corrispondono al nome del dispositivo della partizione primaria con l'aggiunta di una lettera alfabetica che ne indica l'ordine.

- `/dev/hd0`
 - `/dev/hd1`
 - * `/dev/hd1a`
 - * `/dev/hd1b`
 - * `/dev/hd1c`
 - * ...
 - `/dev/hd2`
 - `/dev/hd3`
 - `/dev/hd4`
- `/dev/hd5`
 - `/dev/hd6`
 - `/dev/hd7`
 - `/dev/hd8`
 - `/dev/hd9`

Minix può essere installato in una partizione primaria qualunque, purché ci siano almeno 40 Mibyte a disposizione. Utilizzando la versione di Minix `'i86'`, non conviene tentare di superare i 128 Mibyte.

Avvio

Minix utilizza un sistema di avvio piuttosto sofisticato; per fare un paragone con GNU/Linux, si tratta di qualcosa che compie le stesse funzioni di LILO, o di un cosiddetto *bootloader*.

Il sistema che svolge questa funzione in Minix si chiama *boot monitor* ed è importante capire subito come si utilizza se non si ha molta memoria RAM a disposizione, quanta ne richiederebbe un disco RAM per l'immagine `'ROOT'`.

Per cominciare, dopo aver preparato il dischetto `'ROOT'+USR'`, lo si inserisce senza la protezione contro la scrittura e si avvia l'elaboratore. Questo è ciò che appare.

```
Minix boot monitor 2.5
Press ESC to enter the monitor
Hit a key as follows:
  = Start Minix
```

Premendo il tasto `[Esc]` si attiva il *boot monitor*, mentre premendo `[=]` (si fa riferimento alla tastiera americana e questo simbolo si trova in corrispondenza della nostra lettera `<i>`) si avvia Minix con le impostazioni predefinite.

Dal momento che si immagina di avere a disposizione poca memoria (solo 1 Mibyte), non si può avviare Minix così, perché il contenuto dell'immagine `'ROOT'` verrebbe caricato come disco RAM. È necessario utilizzare subito il *boot monitor*.

```
[Esc]
```

```
[ESC]
```

```
fd0>
```

Si ottiene un invito (*prompt*), attraverso il quale possono essere utilizzati alcuni comandi importanti per predisporre l'avvio del sistema Minix. Per ottenere aiuto si può utilizzare il comando `'help'`.

```
fd0> help [Invio]
```

Si ottiene un riepilogo dei comandi e del modo con cui possono essere utilizzati. Le cose più importanti che si possono fare con il *boot monitor* sono: l'avvio a partire da una partizione differente da quella predefinita; l'assegnamento o il ripristino al valore predefinito di una variabile.

Queste variabili sono solo entità riferite al sistema di avvio e la loro modifica permette di cambiare il modo con cui si avvia il kernel. Il comando **'set'** permette di elencare il contenuto di queste variabili.

```
fd0> set [Invio]
```

```
rootdev = (ram)
ramimagedev = (bootdev)
ramsize = (0)
processor = (286)
bus = (at)
memsize = (640)
emsize = (330)
video = (vga)
chrome = (mono)
image = (minix)
main() = (menu)
```

I valori appaiono tutti tra parentesi tonde perché rappresentano le impostazioni predefinite. Quando si cambia qualche valore, questo appare senza le parentesi.

La prima cosa da cambiare è il dispositivo di avvio, **'rootdev'**. Si deve assegnare il nome di dispositivo riferito all'immagine **'ROOT'** su dischetto. Si tratta di **'/dev/fd0a'**, come dire, la prima partizione secondaria del dischetto. In questo caso, il nome del dispositivo può anche essere indicato senza la parte iniziale, limitandolo al solo **'fd0a'**.

```
fd0> rootdev=fd0a [Invio]
```

```
fd0> set [Invio]
```

```
rootdev = fd0a
ramimagedev = (bootdev)
ramsize = (0)
processor = (286)
bus = (at)
memsize = (640)
emsize = (330)
video = (vga)
chrome = (mono)
image = (minix)
main() = (menu)
```

Per avviare il sistema, basta utilizzare il comando **'boot'** senza argomenti.

```
fd0> boot [Invio]
```

In questo modo si lascia il *boot monitor* e si avvia il kernel. Una volta avviato il sistema, viene richiesto immediatamente l'innesto della seconda immagine, **'USR'**, contenente gli strumenti necessari all'installazione. Avendo avviato senza disco RAM, il dischetto contenente l'immagine **'ROOT'** non può essere tolto e questo è il motivo per il quale deve essere contenuta nello stesso dischetto insieme a **'USR'**.³

```
Minix 2.0.0 Copyright 1997 Prentice-Hall, Inc.

Executing in 16-bit protected mode

Memory size = 970K MINIX = 206K RAM disk = 0K Available = 765K

Mon Nov 3 15:24:15 MET 1997
Finish the name of device to mount as /usr: /dev/
```

Date le premesse, occorre specificare il nome del dispositivo corrispondente all'immagine **'USR'**: si tratta di **'fd0c'**.

```
fd0c [Invio]
```

```
/dev/fd0c is read-write mounted on /usr
Starting standard daemons: update.

Login as root and run 'setup' to install Minix.

Minix Release 2.0 Version 0

noname login:
```

```
root [Invio]
```

```
#
```

Installazione

L'installazione di Minix avviene in tre fasi:

1. preparazione della partizione di destinazione;

2. trasferimento del contenuto del dischetto, ovvero delle immagini **'ROOT'** e **'USR'**;

3. dopo il riavvio, trasferimento dell'archivio **'USR.TAZ'** e possibilmente, se si dispone di una partizione di almeno 40 Mibyte, anche di **'SYS.TAZ'** e **'CMD.TAZ'**.

Setup

Per iniziare l'installazione, dopo aver avviato il sistema Minix dal dischetto, si utilizza lo script **'setup'**.

```
# setup [Invio]
```

```
This is the Minix installation script.

Note 1: If the screen blanks suddenly then hit F3 to select "software scrolling".

Note 2: If things go wrong then hit DEL and start over.

Note 3: The installation procedure is described in the manual page usage(8). It will be hard without it.

Note 4: Some questions have default answers, like this: [y]
Simply hit RETURN (or ENTER) if you want to choose that answer.

Note 5: If you see a colon (:) then you should hit RETURN to continue.
:
```

```
[Invio]
```

Dopo la breve spiegazione, avendo premuto il tasto **[Invio]** si passa all'indicazione del tipo di tastiera. La scelta è ovvia, **'italian'**, anche se non corrisponde esattamente: la barra verticale (quella per rappresentare i condotti) si trova al posto della lettera **<i>** (i accentata). Durante questa fase di installazione conviene utilizzare la tastiera nazionale (**'italian'**) per evitare spiacevoli incidenti quando si utilizza il programma di gestione delle partizioni.

```
What type of keyboard do you have? You can choose one of:

french italian latin-am scandinav uk us-swap
german japanese olivetti spanish us-std

Keyboard type? [us-std]
```

```
italian [Invio]
```

```
Minix needs one primary partition of at least 30 Mb (it fits in 20 Mb, but it needs 30 Mb if fully recompiled. Add more space to taste.)

If there is no free space on your disk then you have to back up one of the other partitions, shrink, and reinstall. See the appropriate manuals of the operating systems currently installed. Restart your Minix installation after you have made space.

To make this partition you will be put in the editor 'part'. Follow the advice under the '!' key to make a new partition of type MINIX. Do not touch an existing partition unless you know precisely what you are doing! Please note the name of the partition (hd1, hd2, ..., hd9, sd1, sd2, ... sd9) you make. (See the devices section in usage(8) on Minix device names.)
:
```

Il programma di Minix che permette di accedere alla tabella delle partizioni è **'part'** ed è ciò che sta per essere avviato. Come sempre, l'uso di un programma di questo genere è molto delicato: un piccolo errore mette fuori uso tutti i dati eventualmente contenuti in altre partizioni.

```
[Invio]
```

Select device	---first---	--geom/last--	-----sectors-----						
Device	Cyl	Head	Sec	Cyl	Head	Sec	Base	Size	Kb
/dev/hd0	?	?	?	?	?	?	?	?	?
Num	Sort	Type							
?	?	?	?	?	?	?	?	?	?
?	?	?	?	?	?	?	?	?	?
?	?	?	?	?	?	?	?	?	?
?	?	?	?	?	?	?	?	?	?

Prima di utilizzare questo programma conviene leggere la sua guida interna, ottenibile con la pressione del tasto **[?]**. Il cursore si presenta inizialmente sull'indicazione del disco, **'/dev/hd0'**, e può essere cambiato semplicemente premendo i tasti **[+]** o **[-]**. Una volta raggiunto il disco desiderato (in questo caso il primo disco va bene), si deve leggere la sua tabella delle partizioni, in modo da rimpiazzare tutti i punti interrogativi che riempiono lo schermo.

```
[r]
```

Select device	Device	----first----	---geom/last---	-----sectors-----	Base	Size	Kb
	/dev/hd0	Cyl Head Sec	Cyl Head Sec				
		0 0 0	615 8 17		0	83640	41820
Num Sort Type							
1*	hd1	86 DOS-BIG	0 1 0	613 7 16	17	83487	41743
2	hd2	00 None	0 0 0	0 0 -1	0	0	0
3	hd3	00 None	0 0 0	0 0 -1	0	0	0
4	hd4	00 None	0 0 0	0 0 -1	0	0	0

Con questo esempio si suppone di avere solo un vecchio disco fisso MFM di circa 40 Mibyte, nel quale la prima partizione primaria risulta inizialmente dedicata al Dos. Così, basta cambiare il numero che identifica il tipo di partizione. Per farlo, vi si posiziona sopra il cursore, spostandolo con i tasti freccia, quindi si usano i tasti [+] o [-] fino a fare apparire il numero 81₁₆. Al primo intervento per cambiare un valore qualsiasi, viene richiesto esplicitamente se si intende modificare effettivamente i dati della tabella delle partizioni.

Do you wish to modify existing partitions (y/n) [y]

Una volta modificato il tipo, la prima partizione dovrebbe apparire così come segue:

Num Sort Type	Device	----first----	---geom/last---	-----sectors-----	Base	Size	Kb
1*	hd1	81 MINIX	0 1 0	613 7 16	17	83487	41743

Quindi si conclude.

[q]

Save partition table? (y/n) [y]

Lo script di configurazione e installazione riprende richiedendo quale sia la partizione su cui installare Minix. In questo caso si tratta della prima, cioè '/dev/hd1'.

```
Please finish the name of the primary partition you have created:
(Just type RETURN if you want to rerun "part") /dev/
```

hd1 [Invio]

```
You have created a partition named: /dev/hd1
The following subpartitions are about to be created on /dev/hd3:

Root subpartition: /dev/hdla 1440 kb
/usr subpartition: /dev/hdlc rest of hd1

Hit return if everything looks fine, or hit DEL to bail out if you want to
think it over. The next step will destroy /dev/hd1.
:
```

Come accennato in precedenza, Minix viene installato in due partizioni secondarie: la prima serve a contenere il file system principale, la seconda per il resto. In seguito si possono innestare anche partizioni successive.

```
Migrating from floppy to disk...

Scanning /dev/hdlc for bad blocks. (Hit DEL to stop the scan if are absolutely
sure that there can not be any bad blocks. Otherwise just wait.)
```

La scansione del disco fisso è necessaria se si utilizza un vecchio disco MFM, come si suppone di avere in questo esempio, mentre può essere inutile con un disco ATA. È importante fare attenzione: se ci sono settori inutilizzabili, vengono creati alcuni file '/usr/.Bad_*' che **non vanno cancellati!** Alla fine, lo script procede a copiare il contenuto del dischetto nel disco fisso.

```
What is the memory size of this system in kilobytes? [4096 or more]
```

La dimensione di memoria RAM disponibile effettivamente è solo di 970 Kibyte, quindi si inserisce questo valore; se la memoria a disposizione fosse maggiore o uguale a 4 Mibyte, non occorrerebbe indicare alcunché, basterebbe solo confermare.

970 [Invio]

```
Second level file system block cache set to 0 kb.
```

A questo punto, termina l'installazione del dischetto nel disco fisso e si può passare a riavviare il sistema da lì.

```
Please insert the installation ROOT floppy and type 'halt' to exit Minix.
You can type 'boot hd1' to try the newly installed Minix system. See
"TESTING" in the usage manual.
```

halt [Invio]

System Halted

Avvio del sistema copiato nel disco fisso

Una volta conclusa l'esecuzione dello script di configurazione e installazione, si ritorna sotto il controllo del *boot monitor*, attraverso il quale è possibile avviare il sistema dalla prima partizione del disco fisso.

fd0> boot /dev/hd1 [Invio]

```
Minix 2.0.0 Copyright 1997 Prentice-Hall, Inc.

Executing in 16-bit protected mode

at-hd0: 615x8x17

Memory size = 970K MINIX = 206K RAM disk = 0K Available = 765K

Mon Nov 3 16:01:27 MET 1997
Starting standard daemons: update.

Login as root and run 'setup /usr' to install floppy sets.

Minix Release 2.0 Version 0

noname login:
# root [Invio]
```

Il suggerimento dato all'avvio ricorda che è possibile installare altre serie di dischetti, a cominciare da 'USR.TAZ', utilizzando il comando 'setup /usr'.

Installazione delle serie di dischetti

Tra i pacchetti di Minix, 'USR.TAZ' è essenziale e cambia a seconda del tipo di architettura (i86 o i386). Però, dal momento che c'è spazio sufficiente nel disco fisso, conviene installare anche 'SYS.TAZ' per poter ricompilare il kernel e 'CMD.TAZ' che contiene i sorgenti dei vari programmi di servizio.

Tutti questi pacchetti devono essere suddivisi in dischetti nel modo visto in precedenza per il caso di 'USR.TAZ'.

setup /usr [Invio]

Lo script 'setup' chiede una serie di conferme.

```
What is the size of the images on the diskettes? [all]
```

Premendo semplicemente [Invio] si intende che i dischetti vadano letti nella loro interezza.

[Invio]

```
What floppy drive to use? [0]
```

Premendo semplicemente [Invio] si fa riferimento alla prima unità, '/dev/fd0'.

[Invio]

```
Please insert input volume 1 and hit return
```

Si inserisce il primo dischetto e si conferma

[Invio]

Inizia la fase di estrazione di quanto contenuto nel primo dischetto, a partire dalla directory '/usr/'. Quando termina l'estrazione del primo dischetto, viene richiesto il successivo, fino alla conclusione.

Conviene ripetere la procedura fino a quando sono stati installati anche gli archivi 'SYS.TAZ' e 'CMD.TAZ'.

Dischetto di avvio

Minix è molto semplice e non è necessario un dischetto di avvio realizzato appositamente. È sufficiente il dischetto utilizzato per iniziare l'installazione. Se si hanno difficoltà con l'avviamento di Minix dal disco fisso, si può avviare il *boot monitor* dal dischetto e con quello utilizzare il comando 'boot /dev/hd1'.

Conclusione

Per chiudere l'attività di Minix, si può fare nel solito modo comune a quasi tutti i sistemi Unix.

shutdown -h now [Invio]

Ricompilazione del kernel

« Anche Minix, nella sua semplicità, richiede una ricompilazione del kernel per la sua ottimizzazione. In particolare, per poter attivare la gestione del TCP/IP occorre passare per la configurazione e ricompilazione.

Il file del kernel, secondo la tradizione di Minix, dovrebbe trovarsi nella directory radice e avere il nome 'minix'. Se però, invece di trattarsi di un file, si tratta di una directory, nella fase di avvio viene eseguito il file più recente contenuto in tale directory. Il kernel normale, cioè quello che si trova dopo l'installazione, dovrebbe essere '/minix/2.0.0'.

Per poter ricompilare il kernel occorre avere installato il pacchetto 'SYS.TAZ'. Si procede come segue:

1. si modifica il file '/usr/include/minix/config.h';
2. ci si posiziona nella directory '/usr/src/tools/';
3. si avvia la compilazione con il comando 'make'.

Al termine si ottiene il file del kernel (o immagine) corrispondente a '/usr/src/tools/image' che si può copiare e rinominare come si ritiene più opportuno.

/usr/include/minix/config.h

« La configurazione che viene proposta deriva dagli esempi precedenti, in cui si ha una particolare penuria di memoria. Seguono solo alcuni pezzi.

```
/* If ROBUST is set to 1, writes of i-node, directory, and indirect blocks
 * from the cache happen as soon as the blocks are modified. This gives a more
 * robust, but slower, file system. If it is set to 0, these blocks are not
 * given any special treatment, which may cause problems if the system crashes.
 */
#define ROBUST 1 /* 0 for speed, 1 for robustness */
```

La macro 'ROBUST' permette di sincronizzare le operazioni di accesso al disco. Nell'esempio mostrato si attiva questa opzione, in modo da poter utilizzare il sistema con tranquillità (e ovviamente con maggiore lentezza).

```
/* Number of slots in the process table for user processes. */
#define NR_PROCS 32
```

Il numero massimo dei processi eseguibili può essere una seria limitazione all'uso simultaneo dell'elaboratore da parte di più utenti, ma la scarsa memoria a disposizione consiglia di mantenere basso questo valore.

```
/* Enable or disable the second level file system cache on the RAM disk. */
#define ENABLE_CACHE2 0
```

Sempre a causa della carenza di memoria, è opportuno disabilitare la memoria cache.

```
/* Include or exclude device drivers. Set to 1 to include, 0 to exclude. */
#define ENABLE_NETWORKING 1 /* enable TCP/IP code */
#define ENABLE_AT_WINI 1 /* enable AT winchester driver */
#define ENABLE_BIOS_WINI 1 /* enable BIOS winchester driver */
#define ENABLE_ESDI_WINI 1 /* enable ESDI winchester driver */
#define ENABLE_XT_WINI 0 /* enable XT winchester driver */
#define ENABLE_ADAPTEC_SCSI 0 /* enable ADAPTEC SCSI driver */
#define ENABLE_MITSUMI_CDROM 0 /* enable Mitsumi CD-ROM driver */
#define ENABLE_SB_AUDIO 0 /* enable Soundblaster audio driver */
```

In questa sezione è importante abilitare ciò che serve ed eliminare il resto. In particolare, è qui che si attiva la connettività TCP/IP, che non risulta attivata in modo predefinito.

```
/* NR_CONS, NR_RS_LINES, and NR_PTYS determine the number of terminals the
 * system can handle.
 */
#define NR_CONS 2 /* # system consoles (1 to 8) */
#define NR_RS_LINES 1 /* # rs232 terminals (0, 1, or 2) */
#define NR_PTYS 2 /* # pseudo terminals (0 to 64) */
```

Il numero predefinito di console virtuali è due, ma può essere espanso, sempre che ciò possa avere senso date le limitazioni del sistema. Invece è importante attivare gli pseudoterminali, cioè il numero massimo di connessioni remote. Volendo gestire la rete, è il caso di indicare almeno uno pseudoterminale.

Per modificare il file '/usr/include/minix/config.h' si può utilizzare 'vi', che è un collegamento a 'elvis', oppure 'elle'.⁴

Si procede con la compilazione.

```
# cd /usr/src/tools [Invio]
```

```
# make [Invio]
```

Al termine della compilazione, se non sono occorsi incidenti, si ottiene il file 'image'.

```
# cp image /minix/rete.0.1 [Invio]
```

Questo dovrebbe bastare, trattandosi del file più recente nella directory '/minix/', è anche quello che viene avviato la volta successiva.

```
# shutdown -h [Invio]
```

File di dispositivo

« Quando si ricompila il kernel è probabile che si renda necessaria la creazione di file di dispositivo che in altre situazioni non sarebbero necessari. In caso della gestione della rete, sono necessari i file seguenti:

- '/dev/eth';
- '/dev/ip';
- '/dev/tcp';
- '/dev/udp';
- '/dev/tty0' e successivi;
- '/dev/pty0' e successivi.

Questo ragionamento vale anche per le console virtuali: se si vogliono molte console, forse è necessario aggiungere i file relativi.

Probabilmente c'è già tutto ciò di cui si può avere bisogno, ma se manca si può creare con lo script 'MAKEDEV'.

```
MAKEDEV dispositivo
```

Per esempio, trovandosi già nella directory '/dev/', si può creare il dispositivo '/dev/tcp' nel modo seguente:

```
# MAKEDEV tcp [Invio]
```

Parametri di avvio

« Anche Minix richiede alcuni parametri di avvio in presenza di hardware particolare. La gestione di questi avviene in modo molto semplice attraverso il *boot monitor*: basta definire una nuova variabile, assegnandole il valore corretto.

Scheda di rete

« Per gestire una rete occorre una scheda di rete Ethernet. Nell'esempio seguente si immagina di disporre di una scheda compatibile con il modello NE2000 configurata con indirizzo di I/O 300₁₆ e IRQ 11.

Il parametro di avvio per ottenere il riconoscimento della scheda Ethernet è 'DPETHn', dove n è il numero della scheda, a partire da zero.

```
DPETHn=indirizzo_i/o:irq:indirizzo_di_memoria
```

La scheda NE2000 non utilizza alcun indirizzo di memoria, quindi, per il nostro esempio occorre il parametro seguente:

```
DPETH0=300:11
```

Come si vede, l'indirizzo di I/O è espresso implicitamente in esadecimale e l'IRQ in decimale, mentre l'indirizzo di memoria viene ommesso trattandosi di una NE2000. Per inserire tale parametro si utilizza il *boot monitor* nel modo seguente:

```
hd0> DPETH0=300:11 [Invio]
```

```
hd0> save [Invio]
```

L'ultima istruzione, 'save', salva questo parametro che altrimenti dovrebbe essere indicato ogni volta che si avvia il sistema.

Se la scheda di rete viene riconosciuta, all'avvio appare il messaggio seguente:

```
Minix 2.0.0 Copyright 1997 Prentice-Hall, Inc.

Executing in 16-bit protected mode

ne2000: NE2000 at 300:11
```

Configurazione della rete

La configurazione della rete va fatta con cura, in modo da non avere bisogno di alcuni demoni che permettono una sorta di autoconfigurazione. Negli esempi seguenti si configura il nuovo sistema Minix tenendo conto di questa situazione:

- *dinkel.brot.dg*, IP 192.168.1.1, servizio DNS e router predefinito;
- *minix.brot.dg*, IP 192.168.1.25, elaboratore Minix.

Per quanto possibile, si fa in modo di non avere bisogno del DNS.

`/etc/hosts`

Volendo attivare localmente la risoluzione dei nomi e degli indirizzi è necessario il file `/etc/hosts`, che va configurato come al solito, esattamente come si fa con GNU/Linux.

```
127.0.0.1 localhost
192.168.1.1 dinkel.brot.dg
192.168.1.25 minix.brot.dg
```

File `«/etc/hostname.file»`

Il file `/etc/hostname.file` serve solo a definire il nome dell'elaboratore locale, in senso generale. Non ha niente a che vedere con le interfacce di rete.

```
# echo "minix.brot.dg" > /etc/hostname.file [Invio]
```

File `«/etc/resolv.conf»`

Il file `/etc/resolv.conf` permette di indicare gli indirizzi dei nodi che forniscono un servizio DNS. Nell'esempio proposto si vuole fare in modo che il sistema di risoluzione dei nomi avvenga localmente, per mezzo di quanto contenuto nel file `/etc/hosts`. Per questo viene indicato come server DNS anche l'indirizzo locale (*loopback*).

```
nameserver 127.0.0.1
nameserver 192.168.1.1
```

File `«/etc/rc.net»`

Lo script `/etc/rc.net` viene utilizzato da `/etc/rc` per attivare la rete. Lo si può utilizzare per attivare l'interfaccia di rete e per definire l'instradamento verso il router (l'instradamento verso la rete connessa all'interfaccia è predefinito).

```
# Attiva l'interfaccia e l'instradamento verso la sua rete.
ifconfig -h 192.168.1.25

# Definisce l'instradamento predefinito verso il router
add_route -g 192.168.1.1
```

File `«/etc/rc»`

Probabilmente, è utile ritoccare il file `/etc/rc`, per eliminare l'avvio automatico di alcuni demoni inutili dal momento che la rete è configurata. Quello che segue è il pezzo che attiva la gestione della rete.

```
# Network initialization.
(</dev/eth </dev/tcp) 2>/dev/null && net=true # Is there a TCP/IP server?

if [ "$net" -a -f /etc/rc.net ]
then
    # There is a customized TCP/IP initialization script; run it.
    . /etc/rc.net
elif [ "$net" ] && [ "hostaddr -e" = 0:0:0:0:0 ]
then
    # No network hardware, configure a fixed address to run TCP/IP alone.
    ifconfig -h 192.9.200.1
fi

if [ "$net" ]
then
    echo -n "Starting network daemons: "
```

1996

```
for daemon in rarpd nonamed irdpd talkd
do
    if [ -f /usr/bin/$daemon ]
    then
        echo -n " $daemon"
        $daemon &
    fi
done
echo .

# Get the nodename from the DNS and set it.
hostaddr -a >/etc/hostname.file || echo noname >/etc/hostname.file

echo -n "Starting network services:"
for pair in 'shell in.rshd' 'login in.rld' \
            'telnet in.telnetd' 'ftp in.ftpd'
do
    set $pair
    if [ -f /usr/bin/$2 ]
    then
        echo -n " $1"
        tcpd $1 /usr/bin/$2 &
    fi
done
echo .

fi
```

Vale la pena di modificare quanto segue:

```
if [ "$net" ]
then
    echo -n "Starting network daemons: "
    for daemon in nonamed talkd ## rarpd nonamed irdpd talkd
    do
        ...
    done
fi
```

Nel pezzo precedente non vengono avviati i demoni `'rarpd'` e `'irdpd'`, che sono necessari rispettivamente per ottenere l'indirizzo IP in base all'indirizzo hardware della scheda Ethernet e a definire gli instradamenti verso i router. Eventualmente, si potrebbe anche evitare di avviare `'talkd'` se non si intende utilizzare `'talk'`. Il demone `'nonamed'` è necessario se non si vuole essere obbligati ad avere un servizio DNS esterno; in pratica è necessario perché venga interpretato il contenuto del file `/etc/hosts`.

Personalizzazione

Il sistema risulta configurato in maniera piuttosto disordinata, a cominciare dal fatto che la directory personale dell'utente `'root'` corrisponde alla directory radice; così, al suo interno si trovano i file di configurazione dell'amministratore. Probabilmente, la prima cosa da fare è quella di creare una directory `'/root/'`, porvi al suo interno i file di configurazione (dovrebbe trattarsi di `'.ellepro.b1'`, `'.exrc'` e `'.profile'`), modificando anche il file `/etc/passwd` in modo da assegnare all'utente `'root'` questa nuova directory.

File `«/etc/passwd»`, `«/etc/group»` e `«/etc/shadow»`

Minix, nonostante la sua semplicità, utilizza le parole d'ordine oscure (*shadow password*). Pertanto, se si tenta di inserire un utente manualmente, occorre intervenire anche su questo file, `/etc/shadow`, altrimenti l'utente non può accedere.

Il file `/etc/group`, se non va bene com'è, deve essere modificato manualmente, mentre per gli utenti conviene affidarsi allo script `'adduser'`.

```
adduser utente gruppo directory_home
```

Dopo aver creato un utente, come al solito è opportuno utilizzare il programma `'passwd'` per assegnare la parola d'ordine.⁵

Tastiera

La mappa della tastiera viene definita attraverso il programma `'loadkeys'` e il file contenente la mappa desiderata. Per cui,

```
# loadkeys ./tastiera.map [Invio]
```

permette di caricare la mappa del file `'tastiera.map'` contenuto nella directory corrente.

La mappa della tastiera, secondo la scelta fatta durante l'installazione di Minix, avviene per mezzo del file `'/etc/keymap'`: se lo

1997

script `/etc/rc` lo trova durante la fase di avvio, lo carica attraverso `'loadkeys'`.

Modifica della mappa

La configurazione della tastiera italiana, per quanto riguarda la versione 2.0 di Minix, non è perfetta. Per modificare la mappa occorre intervenire sul file `/usr/src/kernel/keymaps/italian.src`. Dopo la modifica si deve compilare il sorgente in modo da ottenere il file `/usr/src/kernel/keymaps/italian.map`. Al termine, questo file va copiato e rinominato in modo da sostituire `/etc/keymap`.

Il sorgente corretto potrebbe apparire come nell'esempio seguente, in particolare, per ottenere la tilde (`'~'`) si deve usare la combinazione `[AltGr i]`, mentre per ottenere l'apostrofo inverso (`'`'`) si deve usare la combinazione `[AltGr `]`. I caratteri che si trovano oltre il settimo bit, vengono rappresentati in ottale.

```

/* Modified by Daniele Giacomini daniele@swlibero.org 1998.12.22 */
/* Keymap for Italian standard keyboard, similar to Linux layout. */

ui6_t keymap[NR_SCAN_CODES * MAP_COLS] = {

/* scan-code      !Shift Shift  Alt    AltGr  Alt+Sh Ctrl  */
/* ===== */
/* 00 - none      */ 0,    0,    0,    0,    0,    0,
/* 01 - ESC       */ C('['), C('['), CA('['), C('['), C('['), C('['),
/* 02 - '1'       */ '1',  '1',  A('1'), '1',  '1',  C('A'),
/* 03 - '2'       */ '2',  '2',  A('2'), '2',  '@',  C('@'),
/* 04 - '3'       */ '3',  0234, A('3'), '3',  0234, C('C'),
/* 05 - '4'       */ '4',  '$',  A('4'), '4',  '$',  C('D'),
/* 06 - '5'       */ '5',  '&', A('5'), '5',  '&', C('E'),
/* 07 - '6'       */ '6',  '&', A('6'), '6',  '&', C('F'),
/* 08 - '7'       */ '7',  '/', A('7'), '/',  '/',  C('G'),
/* 09 - '8'       */ '8',  '(', A('8'), '(',  '(',  C('H'),
/* 10 - '9'       */ '9',  ')', A('9'), ')',  ')',  C('I'),
/* 11 - '0'       */ '0',  '=', A('0'), '=',  '=',  C('@'),
/* 12 - '~'       */ '~',  '?', A('~'), '~',  '?',  C('@'),
/* 13 - '='       */ 0215, '^', 0215, '~',  '^',  C('^'),
/* 14 - BS        */ C('H'), C('H'), CA('H'), C('H'), C('H'), 0177,
/* 15 - TAB       */ C('I'), C('I'), CA('I'), C('I'), C('I'), C('I'),
/* 16 - 'q'       */ L('q'), 'Q', A('q'), 'q',  'Q',  C('Q'),
/* 17 - 'w'       */ L('w'), 'W', A('w'), 'w',  'W',  C('W'),
/* 18 - 'e'       */ L('e'), 'E', A('e'), 'e',  'E',  C('E'),
/* 19 - 'r'       */ L('r'), 'R', A('r'), 'r',  'R',  C('R'),
/* 20 - 't'       */ L('t'), 'T', A('t'), 't',  'T',  C('T'),
/* 21 - 'y'       */ L('y'), 'Y', A('y'), 'y',  'Y',  C('Y'),
/* 22 - 'u'       */ L('u'), 'U', A('u'), 'u',  'U',  C('U'),
/* 23 - 'i'       */ L('i'), 'I', A('i'), 'i',  'I',  C('I'),
/* 24 - 'o'       */ L('o'), 'O', A('o'), 'o',  'O',  C('O'),
/* 25 - 'p'       */ L('p'), 'P', A('p'), 'p',  'P',  C('P'),
/* 26 - '['       */ 0212, 0202, 0212, '[',  '[',  C('['),
/* 27 - ']'       */ '+',  '+', A('+'), '+',  '+',  C(')'],
/* 28 - CR/LF     */ C('M'), C('M'), CA('M'), C('M'), C('M'), C('J'),
/* 29 - Ctrl      */ CTRL, CTRL, CTRL, CTRL, CTRL, CTRL,
/* 30 - 'a'       */ L('a'), 'A', A('a'), 'a',  'A',  C('A'),
/* 31 - 's'       */ L('s'), 'S', A('s'), 's',  'S',  C('S'),
/* 32 - 'd'       */ L('d'), 'D', A('d'), 'd',  'D',  C('D'),
/* 33 - 'f'       */ L('f'), 'F', A('f'), 'f',  'F',  C('F'),
/* 34 - 'g'       */ L('g'), 'G', A('g'), 'g',  'G',  C('G'),
/* 35 - 'h'       */ L('h'), 'H', A('h'), 'h',  'H',  C('H'),
/* 36 - 'j'       */ L('j'), 'J', A('j'), 'j',  'J',  C('J'),
/* 37 - 'k'       */ L('k'), 'K', A('k'), 'k',  'K',  C('K'),
/* 38 - 'l'       */ L('l'), 'L', A('l'), 'l',  'L',  C('L'),
/* 39 - ';'       */ 0225, 0207, 0225, '@',  '@',  C('@'),
/* 40 - '`'       */ 0205, 0370, 0205, '#',  '#',  C('@'),
/* 41 - '~'       */ '\\', '|', '\\', '\\', '\\',  C('\\'),
/* 42 - 1. SHIFT+SHIFT  SHIFT, SHIFT, SHIFT, SHIFT, SHIFT, SHIFT,
/* 43 - '\\\'      */ 0227, '|', 0227, 0227, '|',  C('@'),
/* 44 - 'z'       */ L('z'), 'Z', A('z'), 'z',  'Z',  C('Z'),
/* 45 - 'x'       */ L('x'), 'X', A('x'), 'x',  'X',  C('X'),
/* 46 - 'c'       */ L('c'), 'C', A('c'), 'c',  'C',  C('C'),
/* 47 - 'v'       */ L('v'), 'V', A('v'), 'v',  'V',  C('V'),
/* 48 - 'b'       */ L('b'), 'B', A('b'), 'b',  'B',  C('B'),
/* 49 - 'n'       */ L('n'), 'N', A('n'), 'n',  'N',  C('N'),
/* 50 - 'm'       */ L('m'), 'M', A('m'), 'm',  'M',  C('M'),
/* 51 - ','       */ ',',  ':', A(','), ',',  ':',  C('@'),
/* 52 - '.'       */ '.',  ':', A('.'), '.',  ':',  C('@'),
/* 53 - '/'       */ '/',  '_', A('/'), '/',  '_',  C('@'),
/* 54 - r. SHIFT+SHIFT  SHIFT, SHIFT, SHIFT, SHIFT, SHIFT, SHIFT,
/* 55 - '*'       */ '*',  '*', A('*'), '*',  '*',  C('M'),
/* 56 - ALT       */ ALT,  ALT, ALT,  ALT, ALT,  ALT,
/* 57 - ' '       */ ' ',  ' ', A(' '), ' ',  ' ',  C('@'),
/* 58 - CapsLock  */ CALOCK, CALOCK, CALOCK, CALOCK, CALOCK, CALOCK,
/* 59 - F1        */ F1,  SF1, AF1,  AF1,  ASF1, CF1,
/* 60 - F2        */ F2,  SF2, AF2,  AF2,  ASF2, CF2,
/* 61 - F3        */ F3,  SF3, AF3,  AF3,  ASF3, CF3,
/* 62 - F4        */ F4,  SF4, AF4,  AF4,  ASF4, CF4,
/* 63 - F5        */ F5,  SF5, AF5,  AF5,  ASF5, CF5,
/* 64 - F6        */ F6,  SF6, AF6,  AF6,  ASF6, CF6,
/* 65 - F7        */ F7,  SF7, AF7,  AF7,  ASF7, CF7,
/* 66 - F8        */ F8,  SF8, AF8,  AF8,  ASF8, CF8,
/* 67 - F9        */ F9,  SF9, AF9,  AF9,  ASF9, CF9,
/* 68 - F10       */ F10, SF10, AF10, AF10, ASF10, CF10,
/* 69 - NumLock   */ NLOCK, NLOCK, NLOCK, NLOCK, NLOCK, NLOCK,
/* 70 - ScrLock   */ SLOCK, SLOCK, SLOCK, SLOCK, SLOCK, SLOCK,
/* 71 - Home      */ HOME, '7', AHOME, AHOME, '7',  CHOME,

```

```

/* 72 - CurUp    */ UP,  '8',  AUP,  AUP,  '8',  CUP,
/* 73 - PgUp     */ PGUP, '9',  APGUP, APGUP, '9',  CPGUP,
/* 74 - '-'      */ NMN,  '-',  ANMIN, ANMIN, '-',  CNMIN,
/* 75 - Left     */ LEFT, '4',  ALEFT, ALEFT, '4',  CLEFT,
/* 76 - MID      */ MID,  '5',  AMID,  AMID,  '5',  CMID,
/* 77 - Right    */ RIGHT, '6',  ARIGHT, ARIGHT, '6',  CRIGHT,
/* 78 - '+'      */ PLUS, '+',  APLUS, APLUS, '+',  CPLUS,
/* 79 - End      */ END,  '1',  AEND,  AEND,  '1',  CEND,
/* 80 - Down     */ DOWN, '2',  ADOWN, ADOWN, '2',  CDOWN,
/* 81 - PgDown   */ PGDN, '3',  APGDN, APGDN, '3',  CPGDN,
/* 82 - Insert   */ INSRT, '0',  AINSRT, AINSRT, '0',  CINSRT,
/* 83 - Delete   */ 0177, '?',  A(0177), 0177, '?',  0177,
/* 84 - Enter    */ C('M'), C('M'), CA('M'), C('M'), C('M'), C('J'),
/* 85 - ???     */ 0,    0,    0,    0,    0,    0,
/* 86 - ???     */ '<', '>', A('<'), '|', '>', C('@'),
/* 87 - F11      */ F11, SF11, AF11, AF11, ASF11, CF11,
/* 88 - F12      */ F12, SF12, AF12, AF12, ASF12, CF12,
/* 89 - ???     */ 0,    0,    0,    0,    0,    0,
/* 90 - ???     */ 0,    0,    0,    0,    0,    0,
/* 91 - ???     */ 0,    0,    0,    0,    0,    0,
/* 92 - ???     */ 0,    0,    0,    0,    0,    0,
/* 93 - ???     */ 0,    0,    0,    0,    0,    0,
/* 94 - ???     */ 0,    0,    0,    0,    0,    0,
/* 95 - ???     */ 0,    0,    0,    0,    0,    0,
/* 96 - EXT_KEY  */ EXTKEY, EXTKEY, EXTKEY, EXTKEY, EXTKEY, EXTKEY,
/* 97 - ???     */ 0,    0,    0,    0,    0,    0,
/* 98 - ???     */ 0,    0,    0,    0,    0,    0,
/* 99 - ???     */ 0,    0,    0,    0,    0,    0,
/* 100 - ???    */ 0,    0,    0,    0,    0,    0,
/* 101 - ???    */ 0,    0,    0,    0,    0,    0,
/* 102 - ???    */ 0,    0,    0,    0,    0,    0,
/* 103 - ???    */ 0,    0,    0,    0,    0,    0,
/* 104 - ???    */ 0,    0,    0,    0,    0,    0,
/* 105 - ???    */ 0,    0,    0,    0,    0,    0,
/* 106 - ???    */ 0,    0,    0,    0,    0,    0,
/* 107 - ???    */ 0,    0,    0,    0,    0,    0,
/* 108 - ???    */ 0,    0,    0,    0,    0,    0,
/* 109 - ???    */ 0,    0,    0,    0,    0,    0,
/* 110 - ???    */ 0,    0,    0,    0,    0,    0,
/* 111 - ???    */ 0,    0,    0,    0,    0,    0,
/* 112 - ???    */ 0,    0,    0,    0,    0,    0,
/* 113 - ???    */ 0,    0,    0,    0,    0,    0,
/* 114 - ???    */ 0,    0,    0,    0,    0,    0,
/* 115 - ???    */ 0,    0,    0,    0,    0,    0,
/* 116 - ???    */ 0,    0,    0,    0,    0,    0,
/* 117 - ???    */ 0,    0,    0,    0,    0,    0,
/* 118 - ???    */ 0,    0,    0,    0,    0,    0,
/* 119 - ???    */ 0,    0,    0,    0,    0,    0,
/* 120 - ???    */ 0,    0,    0,    0,    0,    0,
/* 121 - ???    */ 0,    0,    0,    0,    0,    0,
/* 122 - ???    */ 0,    0,    0,    0,    0,    0,
/* 123 - ???    */ 0,    0,    0,    0,    0,    0,
/* 124 - ???    */ 0,    0,    0,    0,    0,    0,
/* 125 - ???    */ 0,    0,    0,    0,    0,    0,
/* 126 - ???    */ 0,    0,    0,    0,    0,    0,
/* 127 - ???    */ 0,    0,    0,    0,    0,    0
};

```

Dopo la modifica, si avvia la compilazione.

```
# cd /usr/src/kernel/keymaps/ [Invio]
```

```
# make [Invio]
```

Vengono generati tutti i file di configurazione che non siano già presenti (se si vuole ripetere la compilazione occorre prima rimuovere il file `italian.map`).

```
# cp italian.map /etc/keymap [Invio]
```

Al prossimo riavvio si utilizza così la nuova mappa.

Altri programmi

Il software a disposizione per Minix non è molto e può essere trovato negli stessi FTP da cui si accede ai file del sistema operativo citati qui. In particolare, tra i programmi riferiti alla rete, vale la pena di ricordare i pacchetti `'HTTPD.TAZ'` e `'FROG.TAZ'`. Il primo è un server HTTP molto semplice e il secondo è un programma per il tracciamento dell'instradamento (simile a Traceroute). Sono entrambi molto utili e compilabili facilmente.

Pacchetto «HTTPD.TAZ»

Minix dispone di un server HTTP elementare e lo si trova distribuito nel pacchetto `'HTTPD.TAZ'`. I pacchetti supplementari, come questo, vanno installati a partire dalla directory `'/usr/local/'` e i sorgenti vanno collocati in `'/usr/local/src/'`.

```
# cd /usr/local/src [Invio]
```

Supponendo che il file `'HTTPD.TAZ'` si trovi nel dischetto, innestato nella directory `'/mnt/'`, si può agire come segue:

```
# cat /mnt/HTTPD.TAZ | compress -d | tar xvf - [Invio]
```

Si ottiene la creazione della directory 'httpd/' a partire dalla posizione corrente, cioè '/usr/local/src/'.

```
# cd httpd [Invio]
```

Si procede con la compilazione.

```
# make [Invio]
```

Quindi si installa il programma.

```
# make install [Invio]
```

L'installazione non si occupa di copiare i file di configurazione: bisogna farlo manualmente.

```
# cp httpd.conf /etc [Invio]
```

```
# cp httpd.mtype /etc [Invio]
```

Il demone 'httpd', installato in '/usr/local/bin/', ha bisogno di un utente 'www', in base alla configurazione predefinita del file '/etc/httpd.conf' che non serve modificare.

```
# adduser www operator /usr/home/www [Invio]
```

Naturalmente la scelta della directory personale di questo utente fitizio è solo un fatto di gusto personale. Sempre in base alla configurazione predefinita, occorre aggiungere alla directory personale la directory 'exec/' e all'interno di questa si devono collocare un paio di file.

```
# mkdir /usr/home/www/exec [Invio]
```

```
# cp dir2html /usr/home/www/exec [Invio]
```

```
# cp dir2html.sh /usr/home/www/exec [Invio]
```

Per ultimo, occorre avviare il demone. Per questo conviene ritoccare il file '/etc/rc.net' in modo da aggiungere la riga seguente:

```
tcpd http /usr/local/bin/httpd &
```

Pacchetto «FROG.TAZ»

'frog' è un programma per il tracciamento dell'instradamento. È semplice, ma anche molto importante. L'estrazione dell'archivio avviene nel modo solito, così come la compilazione e l'installazione.

```
# cd /usr/local/src [Invio]
```

```
# cat /mnt/FROG.TAZ | compress -d | tar xvf - [Invio]
```

```
# cd frog [Invio]
```

```
# make [Invio]
```

```
# make install [Invio]
```

Tutto qui.

Copie di sicurezza

Le copie di sicurezza possono essere fatte soltanto utilizzando 'tar' e 'compress'. Dal momento che il sistema è organizzato in modo piuttosto rigido, con una partizione principale molto piccola e una partizione '/usr/', normalmente conviene preoccuparsi solo di questa seconda partizione. Per la prima converrebbe realizzare un dischetto di avvio e installazione con gli stessi file di configurazione, compresi '/etc/passwd', '/etc/group' e '/etc/shadow'.

Archiviazione

Se si dispone di abbastanza spazio libero nella partizione '/usr/', se ne può fare la copia di sicurezza in un file collocato all'interno della stessa partizione. Successivamente si può scaricare su dischetti. Si può procedere nel modo seguente:

```
# cd /usr [Invio]
```

```
# tar cf - . | compress -c > .BKP.TAZ [Invio]
```

Si ottiene il file '/usr/.BKP.TAZ' contenente la copia di quanto contenuto nella directory corrente '/usr/'. Successivamente si può copiare il file ottenuto, a pezzi, su una serie di dischetti inizializzati in precedenza. Si comincia con l'inizializzazione e si suppone di disporre di dischetti da 1440 Kibyte.

```
# format /dev/fd0 1440 [Invio]
```

...

Una volta preparati i dischetti, si può scaricare il file nei dischetti.

```
# dd if=/usr/.BKP.TAZ of=/dev/fd0 bs=1440k count=1 skip=0 [Invio]
```

```
# dd if=/usr/.BKP.TAZ of=/dev/fd0 bs=1440k count=1 skip=2 [Invio]
```

...

Recupero

Per recuperare un sistema archiviato nel modo mostrato nella sezione precedente, si deve cominciare dall'installazione con il dischetto iniziale. La cosa migliore sarebbe l'utilizzo di un dischetto modificato opportunamente in modo che i file di configurazione corrispondano a quanto utilizzato nel proprio sistema.

Dopo l'installazione iniziale che consiste nel trasferimento di quanto contenuto nel dischetto iniziale nel disco fisso, si procede con l'installazione della copia (preparata in precedenza) della partizione collocata a partire da '/usr/'.

```
# setup /usr [Invio]
```

Uno dopo l'altro vengono richiesti tutti i dischetti.

Convivenza tra Minix e GNU/Linux

Se lo si desidera, si può fare convivere Minix assieme a GNU/Linux, nello stesso disco fisso, su partizioni distinte. Ma installare Minix in una partizione libera di un disco in cui GNU/Linux è già stato installato richiede prudenza e attenzione.

- L'installazione di Minix provoca l'alterazione dell'MBR del disco fisso, di conseguenza, al termine si avvia solo Minix. Quindi, prima di installare Minix occorre preparare uno o più dischi di avvio di GNU/Linux, in modo da poter in seguito ripristinare il sistema di avvio attraverso LILO.
- Quando si conclude il lavoro con Minix e si esegue un riavvio con un semplice [Ctrl Alt Canc], si ottiene un avvio a caldo (*warm boot*), ma se dopo si vuole avviare un kernel Linux, **questo non può essere caricato**. Pertanto, è necessario un riavvio a freddo, al limite attraverso lo spegnimento e la riaccensione dell'elaboratore.

LILO

Una volta che si è riusciti a fare riavviare il sistema GNU/Linux, con i dischetti di avvio a cui si faceva riferimento in precedenza, conviene modificare il file '/etc/lilo.conf' in modo che si possa scegliere tra l'avvio di GNU/Linux, Minix ed eventualmente altro.

Supponendo di avere installato Minix nella seconda partizione del primo disco fisso, le righe necessarie nel file '/etc/lilo.conf' sono quelle seguenti:

```
other=/dev/hda2
label=minix
table=/dev/hda
```

Volendo supporre che Minix sia stato installato nel secondo disco fisso, sempre nella seconda partizione, le righe sarebbero quelle seguenti:

```
other=/dev/hdb2
label=minix
table=/dev/hdb
loader=/boot/chain.b
```

In pratica, è esattamente ciò che si fa quando si vuole controllare l'avvio del Dos.

Riferimenti

<

Minix è un sistema operativo molto limitato rispetto a GNU/Linux. Resta comunque l'unica opportunità, almeno per ora, di fronte a vecchi elaboratori i286 o inferiori.

Molti particolari importanti non sono stati descritti, ma le informazioni relative sono comunque accessibili dai siti FTP di distribuzione di Minix e dalla documentazione interna costituita dalle pagine di manuale (`man`).

- Andrew S. Tanenbaum, Alber S. Woodhull, *Operating Systems: Design and Implementation*, 2/e, Prentice-Hall
- <http://www.cs.vu.nl/~ast/minix.html>

¹ **Minix** licenza simile a BSD

² Se si trattasse della seconda unità a dischetti, si parlerebbe di `‘/dev/fd1’`, `‘/dev/fd1a’` e `‘/dev/fd1c’`.

³ Una cosa da sapere subito è che Minix non utilizza la sequenza [`Ctrl C`] per interrompere un programma. Per questo si usa il tasto [`Canc`] da solo.

⁴ Il programma `‘elvis’` in particolare, è molto tradizionale: i tasti freccia generano proprio le lettere corrispondenti e quindi non possono essere usati durante la fase di inserimento. `‘elle’` è un Emacs ridotto al minimo.

⁵ Lo script `‘adduser’` si avvale della directory personale dell'utente `‘ast’` per inserire i file di configurazione iniziali. Questa directory, corrispondente a `‘/usr/ast/’`, svolge il ruolo di scheletro delle directory personali da creare. Volendo si può realizzare un proprio script per rendere la cosa più elegante.