

LTSP – Linux Terminal Server Project – v3.0

James McQuillan

jam@LTSP.org

Copyright © 2002 James A. McQuillan

Diario delle revisioni

Revisione 1.0.4

02-03-2002

Corretto da: jam

Il sistema operativo GNU/Linux e una piattaforma ideale per l'utilizzo di piccoli terminali senza disco fisso. L'obiettivo principale di questo documento è di mostrarti come installare tali clients usando LTSP ed inoltre altre utili informazioni sulle workstations senza disco fisso più in generale.

Indice

<u>Introduzione</u>	1
<u>1. Dichiarazioni</u>	1
<u>2. Copyright and Licenze</u>	2
<u>Capitolo 1. Teoria del funzionamento</u>	3
<u>Capitolo 2. Installazione di LTSP sul server</u>	7
<u>2.1. Installare la versione RPM</u>	7
<u>2.2. Installare la versione TGZ</u>	7
<u>2.3. Configurazione del server</u>	8
<u>2.4. Configurazione specifica delle workstations (clients senza disco)</u>	8
<u>2.4.1. /etc/dhcpd.conf</u>	8
<u>2.4.2. /etc/hosts</u>	10
<u>2.4.3. /opt/lts/i386/etc/lts.conf</u>	10
<u>Capitolo 3. Configurazione delle workstations</u>	12
<u>3.1. Creare il dischetto di partenza</u>	12
<u>Capitolo 4. Usare la workstation</u>	14
<u>Capitolo 5. Stampa</u>	15
<u>5.1. Settaggi sul client</u>	15
<u>5.2. Settaggi sul server</u>	15
<u>Capitolo 6. Soluzione dei problemi</u>	18
<u>6.1. Problemi con l'immagine del dischetto di partenza</u>	18
<u>6.2. Problemi con il DHCP</u>	18
<u>6.2.1. Controllare le connessioni</u>	19
<u>6.2.2. Il processo dhcpd e' attivo?</u>	19
<u>6.2.3. Controlla due volte la configurazione DHCP</u>	20
<u>6.2.4. E ipchains o iptables a bloccare le richieste?</u>	20
<u>6.2.5. La workstation manda le richieste?</u>	21
<u>6.3. Problemi con TFTP</u>	21
<u>6.3.1. Il processo tftpd non e' attivo</u>	21
<u>6.3.2. Il kernel non e' dove tftpd se l'aspetta</u>	21
<u>6.4. Problemi con il filesystem di root NFS</u>	21
<u>6.4.1. Messaggio No init found</u>	22
<u>6.4.2. Il server da un errore -13</u>	22
<u>6.4.3. Problemi con i processi NFS (portmap, nfsd & mountd)</u>	22
<u>6.5. Problemi con Xserver</u>	24
<u>6.6. Problemi con il Display manager</u>	24
<u>6.6.1. Schermo grigio con un grosso cursore a forma di X</u>	25
<u>Capitolo 7. I kernel</u>	28
<u>7.1. Kernels standard distribuiti con LTSP</u>	28
<u>7.2. Costruire il tuo kernel</u>	28
<u>7.2.1. Scaricare i sorgenti del kernel</u>	28
<u>7.2.2. Patches per il kernel</u>	29

Indice

<u>Capitolo 7. I kernel</u>	
7.2.3. Configurare le opzioni del kernel	30
7.2.4. Compilare il kernel	32
7.2.5. Taggare il kernel per Etherboot	32
<u>Capitolo 8. Il file lts.conf</u>	33
8.1. Esempio di lts.conf	33
8.2. Parametri disponibili in lts.conf	33
8.2.1. Parametri generali	33
8.2.2. Parametri del sistema X–Windows	35
8.2.3. Parametri per il touch screen	37
8.2.4. Parametri per le applicazioni locali	37
8.2.5. Parametri per la tastiera	38
8.2.6. Parametri per la configurazione della stampante	38
<u>Capitolo 9. Local Applications</u>	40
9.1. Vantaggi delle applicazioni che girano in locale	40
9.2. Svantaggi delle applicazioni che girano in locale	40
9.3. Configurazione del server per le applicazioni locali	41
9.3.1. Parametri in lts.conf	41
9.3.2. Network Information Service – NIS	41
9.4. Configurazione delle applicazioni	41
9.5. Lanciare le applicazioni locali	43
<u>Capitolo 10. Esempi di configurazioni</u>	44
10.1. Mouse seriale	44
10.2. Mouse PS/2 Wheel (con rotellina)	44
10.3. Stampante USB collegata ad un ThinkNic	44
10.4. Forzare il caricamento di XFree86 3.3.6 sulla workstation	44
<u>Capitolo 11. Altre fonti di informazioni</u>	45
11.1. Riferimenti online	45
11.2. Pubblicazioni su carta	45

Introduzione

LTSP fornisce un modo semplice per riutilizzare computers a basso costo come terminali grafici o in modo testo collegandoli ad un server GNU/Linux.

In un moderno ufficio, generalmente I PC utilizzati sono relativamente potenti basati su processori Intel o compatibili distribuiti su ogni scrivania. Ognuno di essi con alcuni gigabyte di disco fisso. Gli utenti memorizzano i loro dati su questi dischi fissi su cui raramente viene effettuato il back-up.

Siamo veramente certi che abbia senso avere un PC completo su ogni scrivania?

Noi crediamo di no.

Fortunatamente ce unalternativa. Usando LTSP, puoi acquistare PC datati, rimuovere il disco fisso, il floppy ed il cdrom, ed aggiungere una scheda di rete con supporto per la partenza da rete. Molte schede di rete hanno questa caratteristica ed aspettano solo una eeprom per partire.

Durante la fase di partenza, la workstation (il pc senza disco fisso) ottiene il suo indirizzo IP ed il suo kernel dal server e poi monta un filesystem tramite la rete ed il protocollo NFS.

Questa workstation puo essere configurata in tre modi:

- **Grafica usando X windows**

Usando X window, la workstation puo essere usata per lanciare qualsiasi applicazione sul server di partenza, o su qualunque altro server sulla rete.

- **Testo basato su sessioni telnet**

La workstation puo collegarsi in telnet sul server. Ogni sessione sara messa sun un differente schermo virtuale, selezionato tramite la pressione dei tasti ALT+F1 fino a ALT+F9 per un totale di 9 sessioni separate.

- **Prompt dei comandi**

La workstation ti lascia alla console per debuggare problemi con NFS o con il sistema grafico X windows. Questa modalita e motlo utile durante la fase di configurazione per risolvere vari problemi.

La cosa veramente bella e che puoi avere molte workstations che funzionano con un unico server GNU/Linux. Quante? Beh, questo dipende dalle dimensioni del server e dal tipo di applicazioni usate.

Non e raro avere 40 workstations, tutte con Netscape e StarOfficeu caricato, che girano con un server equipaggiato con doppio Pentium3 a 650Mhz ed 1 Gb di ram. Noi sappiamo bene che questa configurazione funziona egregiamente, ed il carico medio va raramente sopra 1.0!

1. Dichiarazioni

Ne lautore, ne il distributore, ne altre persone che contribuiscono a questa documentazione sono in alcun modo responsabili per alcun tipo di danno fisico, finanziario, morale o in altri tipi di danneggiamento che

seguiranno la lettura di questo testo.

2. Copyright and License

Questo documento è copyright 2001 di James McQuillan, e viene rilasciato sotto i termini della licenza GNU Free Documentation, che viene quindi incorporata dalla citazione.

Capitolo 1. Teoria del funzionamento

Far partire una workstation senza disco include molte diverse fasi. Capire cosa sta accadendo mentre si procede rendera piu facile la risoluzione di eventuali problemi, nel caso dovessero sorgere.

Questo esempio si basa sulla seguente configurazione:

- Una workstation standard basata su processore x86
- Scheda di rete Linksys LNE100TX con una eeprom Etherboot installata
- Chipset grafico Intel i810
- Server con sistema operativo Redhat 7.2
- DHCP
- Sottorete IP 192.168.0.0/24

Supponendo che il server abbia il pacchetto LTSP installato, questo e quello che succede:

1. Quando accendi la workstation, questa inizia il suo test hardware, chiamato POST (Power On Self Test).
2. Durante il POST, il bios cerca le eventuali rom di espansione. La scheda di rete ne contiene una, la eeprom Etherboot. Il bios quindi ne prendera atto.
3. Una volta che il POST sara terminato, verra eseguito il codice Etherboot contenuto nella eeprom.
4. Il codice Etherboot cerchera una scheda di rete. Una volta trovata, questa verra inizializzata.
5. Il codice Etherboot fara quindi una richiesta DHCP sulla rete locale. La richiesta includera il MAC address della scheda di rete.
6. Il processo dhcpd sul server leggerà la richiesta e cercherà nel suo file di configurazione se e presente un MAC address corrispondente.
7. Il processo dhcpd costruirà un pacchetto di risposta contenente molte informazioni. Questo pacchetto sara inviato indietro alla workstation ed includera:
 - ◆ Indirizzo IP per la workstation
 - ◆ Settaggio NETMASK per la rete locale
 - ◆ Directory e nome del kernel da scaricare (dal server)
 - ◆ Directory e nome del filesystem NFS da montare come root
 - ◆ Parametri opzionali da passare al kernel, tramite linterfaccia comandi standard del kernel di Linux.
8. Il codice Etherboot riceverà la risposta del server, e configurerà linterfaccia TCP/IP nella scheda di rete con i parametri ricevuti.
9. Usando il protocollo TFTP (Trivial File Transfer Protocol), il codice Etherboot contatterà il server e scaricherà il kernel.
10. Una volta che il kernel sara stato completamente scaricato sulla workstation, il codice Etherboot lo mettera nella corretta locazione di memoria.

11. Il controllo viene quindi passato al kernel, che inizializzera lintero sistema e tutte le periferiche che potra riconoscere.
12. Qui comincia il divertimento. Attaccata alla fine del kernel, ce unimmagine di un filesystem. Questa viene caricata in memoria come se fosse un normale ramdisk, e temporaneamente montata come filesystem di root. Un parametro del kernel (`root=/dev/ram0`) gli dice di montare limmagine come filesystem di root.
13. Normalmente, quando il kernel finisce loperazione di boot, viene lanciato il programma **init**. Ma in questo caso, abbiamo istruito il kernel per fare altrimenti, e lanciare un nostro script, tramite un altro parametro (**init=/linuxrc**).
14. Lo script **/linuxrc** esegue una scansione del bus PCI cercando una scheda di rete. Ogni volta che ne trova una, esso guarda nel file `/etc/niclist` per vedere se corrisponde. Trovata una corrispondenza, il nome del modulo driver viene stabilito e caricato. Nel caso di schede di rete ISA, questo driver deve essere specificato come parametro del kernel, assieme a qualunque parametro aggiuntivo richiesto, come indirizzi IRQ ecc.
15. Il programma **dhclient** verra quindi eseguito per fare unaltra richiesta al server DHCP. Siamo costretti a fare questa ulteriore richiesta perche la prima viene utilizzata dal kernel, ed inoltre il kernel ignorerà qualsiasi server NFS specificato dal DHCP. Questo e particolarmente importante nel caso to voglia usare un server NFS diverso dal server TFTP.
16. Quando **dhclient** riceve una risposta dal server, il file **/etc/dhclient-script** verra eseguito, le informazioni verranno recuperate e la scheda eth0 verra configurata usando quelle informazioni.
17. Fino ad ora, il filesystem di root e stato un ramdisk. Ora, lo script `/linuxrc` montera un nuovo filesystem su root attraverso NFS. La directory che viene esportata per questo scopo e generalmente **/opt/ltsp/i386**. Ma non puo montare semplicemente il nuovo filesystem su /. Deve infatti prima montarlo su `/mnt`. Eseguira quindi unoperazione chiamata **pivot_root**. `pivot_root` scambiera il root filesystem corrente con un nuovo filesystem. Quando loperazione sara ultimata, il filesystem NFS sara montato su /, ed il vecchio ramdisk su `/oldroot`.
18. Ona volta che il filesystem NFS sara montato, il compito di `/linuxrc` sara finito, e potra essere eseguito il vero programma **init**.
19. Init leggerà il file `/etc/inittab` ed iniziera a configurare lambiente della workstation.
20. Una caratteristica di init e quella della presenza dei cosiddetti *runlevel*. Ogni runlevel fa partire diversi set di servizi per la workstation. Le workstation LTSP partono con runlevel '2'. Questa opzione viene settata dalla riga *initdefault* nel file `inittab`.
21. Una delle prime cose che `inittab` fa e lanciare il comando **rc.local** quando la workstation e nello stato '**sysinit**'.
22. Lo script `rc.local` creera un ramdisk di 1mb che dovra contenere tutti I files che necessitano di essere modificati durante il normale funzionamento.
23. Questo ramdisk verra montato sulla directory `/tmp`. Ogni file che deve essere scritto sara messo qui, ed alcuni links provvederanno a far si che questi files si trovino anche nella giusta collocazione della root directory.

24. Viene montato il filesystem `/proc`.
 25. Se la workstation e stata configurata per eseguire lo swap della ram su NFS, la directory `/var/opt/ltsp/swapfiles` verra montata su `/tmp/swapfiles`. Quindi, se qui non e presente nessun file di swap, ne verra automaticamente creato uno. La grandezza di questo file viene stabilita all'interno del file `lts.conf`.
- Il file di swap verra infine abilitato tramite il comando **swapon**.
26. L'interfaccia di **loopback** per la rete viene configurata. Questa e la scheda di rete virtuale che ha `127.0.0.1` come suo indirizzo IP.
 27. Se le applicazioni locali sono abilitate, la directory **/home** verra montata, in modo che le varie applicazioni possano accedere alle directory degli utenti.
 28. Alcune directory vengono create nel filesystem `/tmp` per contenere alcuni files temporanei che sono necessari quando la workstation lavora. Ad esempio:

- a. `/tmp/compiled`
- b. `/tmp/var`
- c. `/tmp/var/run`
- d. `/tmp/var/log`
- e. `/tmp/var/lock`
- f. `/tmp/var/lock/subsys`

verranno tutte create.

29. Il sistema X Windows verra ora configurato. Nel file **lts.conf**, ce un parametro chiamato **XSERVER**. Se questo parametro e mancante oppure settato su "**auto**", verra tentata la ricerca automatica della scheda video. Se questa scheda e PCI, otterremo il codice PCI Vendor e Device id, e vedremo se questa scheda e presente nel file **/etc/vidlist**.

Se la scheda e supportata da XFree86 4.X, la routine `pci_scan` ritornera il nome del modulo driver da caricare. Se la scheda e invece solo supportata da XFree86 3.3.6, `pci_scan` ritornera il nome del server X da usare. Lo script `rc.local` puo distinguere la differenza di questo output poiche i server 3.3.6 iniziavano tutti con 'XF86_'.

30. Se viene utilizzato XFree86 4.x, verra richiamato lo script **/etc/rc.setupx** che genera il file di configurazione `XF86Config` per X4. Se invece viene usato XFree86 3.3.6, verra richiamato lo script **/etc/rc.setupx3** per produrre lo stesso file, ma in formato X3.

Il file `XF86Config` verra creato in base alle informazioni contenute nel file di configurazione **/etc/lts.conf**.

31. Quando l'esecuzione dello script `rc.setupx` termina, il controllo tornera ad `rc.local`. Quindi lo script **/tmp/start_ws** sara creato. Questo script e responsabile della partenza corretta del server X.

32. Il file `/tmp/syslog.conf` verra creato. Questo file conterra informazioni che diranno, al processo **syslogd**, a quale host sulla rete dovra inviare i dati di log. Questo host viene specificato sempre all'interno del file `lts.conf`. Ce un link chiamato `/etc/syslog.conf` che punta al file `/tmp/syslog.conf`.
33. Il processo **syslogd** viene fatto partire, usando il file di configurazione di cui abbiamo parlato nel passo precedente.
34. Il controllo viene passato di nuovo ad **init**. Init guardera al parametro **initdefault** per determinare in quale **runlevel** entrare. A partire da `lts_core-2.08`, il valore di default e **2**.
35. Un runlevel di **2** fara partire lo script **set_runlevel** che leggerà il file `lts.conf` per determinare quale sarà il vero runlevel nel quale girerà la workstation.

I runlevels standard in LTSP sono **3, 4 e 5**.

- ◆ **3** – Partira una shell, utile per debuggare problemi vari alla workstation.
 - ◆ **4** – Partiranno una o piu sessioni telnet, utile nel caso si stiano sostituendo vecchi terminali testuali.
 - ◆ **5** – Modalita grafica (GUI). Verra lanciato X windows, e verra inviata una richiesta XDMCP al server, che portera una finestra di login per permetterti di entrare. Avrai bisogno di un display manager in ascolto sul server, come ad esempio **XDM, GDM o KDM**.
-

Capitolo 2. Installazione di LTSP sul server

Il pacchetto LTSP è disponibile nei formati **RPM** e **TGZ**. Scegli quello che preferisci e segui la sezione appropriata.

Se vuoi far girare X windows sulla workstation, sono 4 i pacchetti da scaricare. Tieni presente che, per lo scopo di questo documento, abbiamo una workstation con una scheda di rete Tulip e un chipset video Intel i810.

1. LTSP Core package
2. Kernel package
3. X Core package
4. X Fonts package

Il pacchetto X Fonts package non è proprio necessario, ma per le prime installazioni è raccomandato. Una volta che avrai fatto pratica sui settaggi di server e workstation, potrai settare l'X Font Server (XFS) sul server.

Dopo che i pacchetti sono installati, il sistema LTSP deve essere configurato. Questo vuol dire modificare i files di configurazione in modo da abilitare il server ad offrire i servizi necessari alle varie workstations.

2.1. Installare la versione RPM

Scarica l'ultima versione dei pacchetti ltsp ed installali tramite i seguenti comandi:

```
rpm -ivh ltsp_core-3.0.0-1.i386.rpm
rpm -ivh ltsp_kernel-3.0.1-1.i386.rpm
rpm -ivh ltsp_x_core-3.0.1-1.i386.rpm
rpm -ivh ltsp_x_fonts-3.0.0-0.i386.rpm
```

Questa procedura installerà i pacchetti nella directory `/opt/ltsp/i386`.

2.2. Installare la versione TGZ

Scarica l'ultima versione dei pacchetti ltsp ed installali tramite i seguenti comandi:

```
tar xzf ltsp_core-3.0.0-1-i386.tgz
cd ltsp_core
./install.sh
cd ..

tar xzf ltsp_kernel-3.0.1-1-i386.tgz
cd ltsp_kernel
./install.sh
cd ..

tar xzf ltsp_x_core-3.0.1-1-i386.tgz
cd ltsp_x_core
./install.sh
cd ..
```

```
tar xzf ltsp_x_fonts-3.0.0-0-i386.tgz
cd ltsp_x_fonts
./install.sh
cd ..
```

Questa procedura installerà i pacchetti nella directory `/opt/ltsp/i386`.

2.3. Configurazione del server

Una volta terminata l'installazione dei pacchetti, devi entrare nella directory `/opt/ltsp/templates`. Qui ci sono diversi files che configureranno i files di sistema all'interno del tuo server. Ognuno di questi è responsabile di un file di sistema. Da un'occhiata ad ognuno di questi files per essere sicuro che concordino con quello che vuoi ottenere. Questi files potrebbero rendere il tuo server vulnerabile ad attacchi esterni, quindi potresti voler modificare tutti i files a mano. Nel caso invece tu voglia far tutto in automatico, lancia, come utente root, il comando `ltsp_initialize`:

```
cd /opt/ltsp/templates
./ltsp_initialize
```

Questo comando ti chiederà alcune informazioni sui servizi che vuoi configurare. Esso inizierà i seguenti servizi:

- XDM – X Display Manager
- GDM – Gnome Display Manager
- Script di partenza del Display manager
- bootp
- Il file `/etc/exports` di NFS
- tcpwrappers
- Port mapper
- syslogd
- Script di partenza di TFTP

2.4. Configurazione specifica delle workstations (clients senza disco)

Questo è il momento di dire ad LTSP le caratteristiche delle tue workstations. Ci sono tre files che contengono queste informazioni.

1. `/etc/dhcpd.conf`
2. `/etc/hosts`
3. `/opt/ltsp/i386/etc/lts.conf`

2.4.1. `/etc/dhcpd.conf`

La workstation ha bisogno del suo indirizzo IP e di altre cose. Essa prenderà le seguenti informazioni dal server DHCP:

- indirizzo IP
- hostname
- indirizzo IP del server
- Default gateway

- directory e nome del kernel da scaricare
- server e directory da montare come filesystem di root

Per il nostro ambiente di esempio, abbiamo scelto DHCP per assegnare questi dati alle workstation.

Durante lo script `ltsp_initialize`, viene installato un file `dhcpd.conf` di esempio. Viene chiamato `/etc/dhcpd.conf.example` e può essere copiato sopra il file `/etc/dhcpd.conf` come base per la tua configurazione reale. Dovrai modificarne solo le parti che sono diverse nel tuo caso reale.

```
default-lease-time      21600;
max-lease-time          21600;

option subnet-mask      255.255.255.0;
option broadcast-address 192.168.0.255;
option routers          192.168.0.254;
option domain-name-servers 192.168.0.254;
option domain-name      "ltsp.org";
option root-path        "192.168.0.254:/opt/ltsp/i386";

shared-network WORKSTATIONS {
    subnet 192.168.0.0 netmask 255.255.255.0 {
    }
}

group {
    use-host-decl-names on;
    option log-servers 192.168.0.254;

    host ws001 {
        hardware ethernet 00:E0:18:E0:04:82;
        fixed-address 192.168.0.1;
        filename "/lts/vmlinuz.ltsp";
    }
}
```

Figura 2–1. /etc/dhcpd.conf

Dalla versione 2.09pre2 di LTSP, non dovrai più specificare un particolare kernel da scaricare. Il kernel specificato di default abilita già tutte le schede di rete supportate da Linux. Ci sono due kernels inclusi nel pacchetto LTSP: uno ha la patch Linux Progress Patch (LPP) applicata, l'altro no. I nomi sono:

```
vmlinuz-2.4.9-ltsp-5
vmlinuz-2.4.9-ltsp-lpp-5
```

Potrai aver notato che il kernel se ne sta nella directory `/tftpboot/lts`, ma che nel file `/etc/dhcpd.conf` la voce `filename` non presenta la parte iniziale `/tftpboot`. Questo perché dalla Redhat versione 7.1, il servizio TFTP viene lanciato con il parametro `-s`; questo fa sì che `tftpd` parta in modalità **sicura**, facendo un **chroot** sulla directory `/tftpboot` prima di partire. Quindi, i file disponibili al servizio `tftpd` sono solo quelli contenuti nella directory `/tftpboot`.

Altre distribuzioni Linux potrebbero non avere questa opzione settata, quindi avrai bisogno di editare il file `dhcpd.conf` di conseguenza.

2.4.2. /etc/hosts

I computers comunicano piuttosto bene usando gli indirizzi IP. Ma poi arriviamo noi umani, e mettiamo dei bei nomi sui di essi, perche non riusciamo a ricordare gli indirizzi. E qui dove il DNS o il file `/etc/hosts` entrano in gioco: questo passaggio da indirizzi IP a nomi non e normalmente rischiesto, tranne che in un ambiente LTSP. Questo perche senza i nomi, NFS darebbe errori di permesso negato quando la workstation tenta di montare il filesystem di root.

Oltre a cio, se la workstation non e elencata nel file `/etc/hosts`, potresti aver problemi anche con i display manager **GDM** e **KDM**.

2.4.3. /opt/ltsp/i386/etc/lts.conf

Ci sono molte voci che possono essere specificate in questo file.

Il file `lts.conf` ha una sintassi molto semplice ed e diviso in piu sezioni. Ce la sezione chiamata **[default]** e ci possono essere tante altre sezioni, una per ogni workstation, identificate attraverso l'hostname, l'indirizzo IP o il MAC address.

Un tipico `lts.conf`:

```
#
# Config file for the Linux Terminal Server Project (www.ltsp.org)
#

[Default]
SERVER                = 192.168.0.254
XSERVER               = auto
X_MOUSE_PROTOCOL     = "PS/2"
X_MOUSE_DEVICE       = "/dev/psaux"
X_MOUSE_RESOLUTION   = 400
X_MOUSE_BUTTONS      = 3
USE_XFS                = N
LOCAL_APPS            = N
RUNLEVEL              = 5

[ws001]
USE_NFS_SWAP          = Y
SWAPFILE_SIZE         = 48m
RUNLEVEL              = 5

[ws002]
XSERVER               = XF86_SVGA
LOCAL_APPS            = N
USE_NFS_SWAP          = Y
SWAPFILE_SIZE         = 64m
RUNLEVEL              = 3
```

Esempio 2–1. lts.conf file

Segue un breve elenco di alcune di queste voci:

XSERVER

Se la tua scheda video e PCI ed e supportata da Xfree86 4.1, avrai solo bisogno del pacchetto `lts_x_core`. Questo infatti contiene tutti i drivers disponibili per X4.

Ci sono diversi pacchetti di XFree86 3.3.6 disponibili per LTSP, giusto nel caso in cui la tua scheda non sia supportata da XFree86 4.1.

Puoi scrivere voci per ognuna delle workstations, o puoi modificare la sezione di default per interessare tutte le workstations assieme.

La nostra workstation di esempio ha un chipset video Intel i810 che puo essere riconosciuto automaticamente, quindi non abbiamo bisogno di nessuna voce `XSERVER` in questo file. Ad ogni modo, la voce `XSERVER` puo essere specificata ugualmente, se vuoi, o puo anche essere settata ad auto.

RUNLEVEL

Noi vogliamo la modalita grafica, quindi setteremo questa voce a 5.

Capitolo 3. Configurazione delle workstations

Una volta che il server è a posto, è tempo di focalizzare la nostra attenzione sulla workstation.

Il pacchetto LTSP riguarda tutto quello che succede dopo che il kernel è entrato in memoria. Ci sono molti modi per far entrare il kernel in memoria, come ad esempio Etherboot, Netboot, PXE e anche il normale floppy disk.

Per lo scopo di questo documento useremo un floppy disco contenente il codice del progetto **Etherboot**.

3.1. Creare il dischetto di partenza

Etherboot è un pacchetto software per creare immagini ROM che possono scaricare codice tramite un collegamento di rete per essere eseguito su computers x86. Molte schede di rete hanno uno zoccolo dove può essere installata una EEPROM o una ROM. Il codice Etherboot può essere caricato su di essa.

—Ken Yap

Anche Etherboot è Open Source e soggetta alla licenza GNU General Public License, Versione 2 (GPL2).

Puoi scaricare il pacchetto Etherboot e configurarlo per il tipo di ROM di cui hai bisogno. Poi, puoi compilarne i sorgenti per produrre un'immagine ROM che può essere scritta in una EEPROM o direttamente su un floppy disk per scopi di test.

Un'alternativa molto più semplice è andare al sito di Marty Connor www.Rom-O-Matic.net.

Marty ha fatto un lavoro eccellente mettendo a disposizione un sito web che compila in modo automatico i sorgenti di Etherboot e fornisce l'immagine della ROM già pronta. Su questo sito, tu selezioni il tipo di scheda di rete ed il tipo di immagine che vuoi; poi si possono modificare altre opzioni. Quindi, premendo sul bottone 'Get ROM' l'immagine potrà essere scaricata dopo pochi istanti.

La nostra workstation ha una scheda di rete Linksys LNE100TX, versione 4.1. Questa scheda ha un chipset ADMTek Centaur-P, quindi selezioniamo dal sito la scheda tipo *centaur-p*.

Non abbiamo bisogno di fare altri cambiamenti alla configurazione di default, quindi passiamo oltre.

Per quanto riguarda il tipo di formato, selezioneremo 'Floppy Bootable ROM Image'. Questo formato inizia con una testata di 512 byte contenente il codice per caricare Etherboot in memoria quando eseguito.

Premiamo ora il tasto 'Get ROM'. In pochi secondi l'immagine verrà generata e si aprirà la finestra di dialogo per salvare l'immagine sul nostro disco fisso.

Solitamente, io salvo il file nella directory `/tmp`. Quindi per questa immagine, specifico qualcosa come:
`/tmp/eb-5.0.2-centaur-p.lzdsk`

Una volta salvata, dovrai scrivere l'immagine sul floppy disk: inserisci quindi un dischetto vuoto nel drive e lancia il seguente comando:

```
cat /tmp/eb-5.0.2-centaur-p.lzdsk > /dev/fd0
```

Capitolo 4. Usare la workstation

Supponendo che il server e la workstation siano configurati correttamente, tutto quello che ce da fare sarebbe inserire il dischetto nel drive della workstation ed accenderla.

Il codice Etherboot sara letto dal floppy e trasferito in memoria, la scheda di rete verra riconosciuta ed inizializzata, la richiesta dhcp verra inviata sulla rete ed una risposta tornera dal server e il kernel verra scaricato alla workstation. Una volta che il kernel avra inizializzato lhardware della workstation, Xwindow partira e la finestra della login dovrebbe apparire sulla workstation come lesempio sottostante riporta.



Figura 4–1. Schermata di login

A questo punto puoi entrare. Una cosa importante da tenere a mente e che tu stai entrando tramite login nel server. Tutti i comandi che imposti vengono eseguiti sul server e mostrano il loro output sul monitor della workstation. Questa e la potenza di X windows.

Puoi eseguire qualunque programma supportato dal server.

Capitolo 5. Stampa

Oltre alla funzionalità di terminale grafico o testo, la workstation può funzionare anche come print server, permettendo all'utente di collegare fino a tre stampanti seriali o parallele.

Questa funzionalità è completamente trasparente, in termini di prestazioni, all'utente, che non noterà nemmeno il piccolo traffico di rete con il server.

5.1. Settaggi sul client

LTSP utilizza il programma `lp_server` sulla workstation per redirigere le stampe dal server alla stampante collegata alla workstation.

Per abilitare la stampante sulla workstation, ci sono un set di voci da inserire all'interno del file `lts.conf`.

```
[ws001]
PRINTER_0_DEVICE = /dev/lp0
PRINTER_0_TYPE   = P
```

L'esempio qui riportato farà partire il programma `lp_server` come servizio, il quale ascolterà sulla porta TCP/IP 9100, in attesa di una stampa dal server. I dati di stampa saranno convogliati sulla porta parallela `/dev/lp0`.

Ci sono molte altre opzioni disponibili. Controlla la sezione sul file `lts.conf` più avanti in questo documento per ulteriori informazioni sulla configurazione delle stampanti.

5.2. Settaggi sul server

Configurare la stampante sul server si riduce a definire la coda di stampa, utilizzando gli strumenti di configurazione già presenti sul server.

In Redhat 7.2, ci sono sia strumenti grafici che testuali. Lo strumento grafico è chiamato `printconf-gui`, mentre quello testuale `printconf-tui`. Versioni meno recenti di Redhat avevano un programma chiamato `printtool`. Questo esiste ancora anche in Redhat 7.2, ma serve solo a richiamare `printconf-gui`. Altre distribuzioni di Linux hanno altri software.

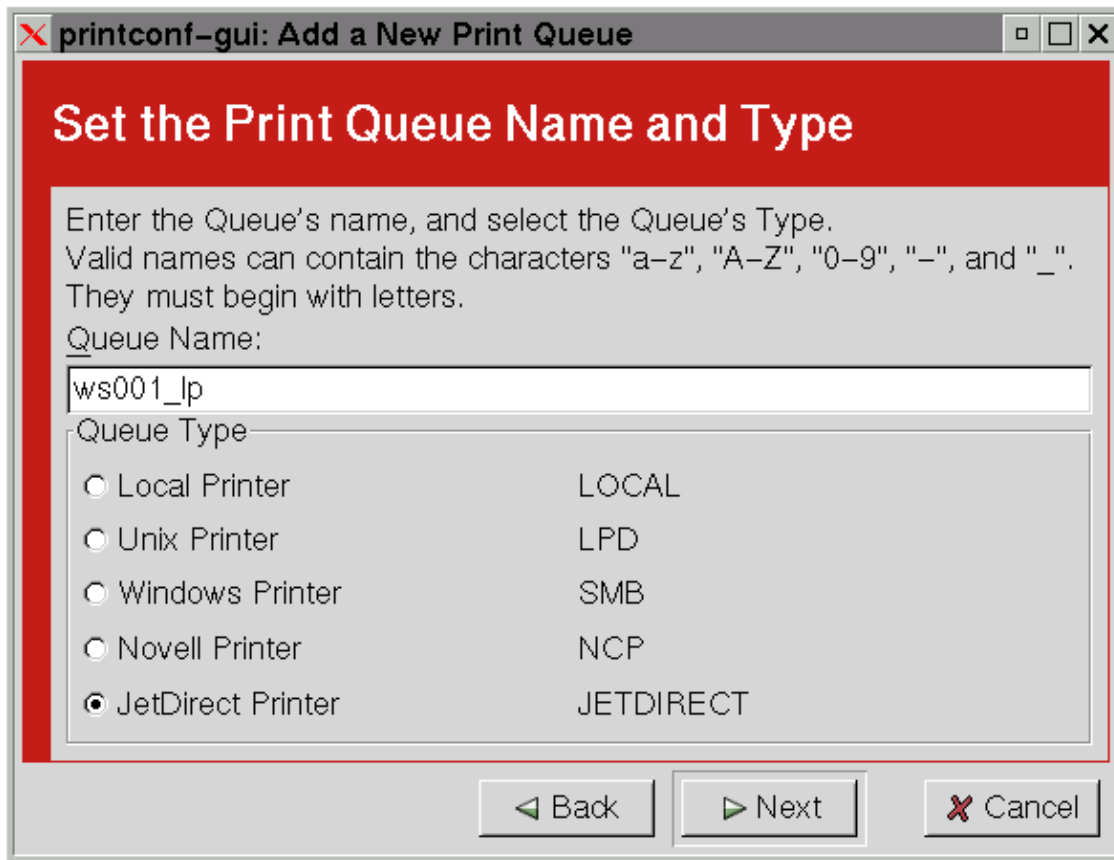


Figura 5–1. Il software printconf-gui mentre si aggiunge una stampante

Una volta lanciato il software di configurazione per la stampante, devi ovviamente aggiungerne una. Il programma `lp_server` permette alla workstation di emulare un server HP JetDirect. Tu dovrai quindi creare solamente una normale stampante **JetDirect**.

Inoltre, dovrai dare un nome alla coda di stampa; questo può essere un nome di pura fantasia, ma dandone uno sensato sarà più facile da gestire in futuro. In più il nome può contenere solo i seguenti caratteri:

- "a-z" lettere minuscole
- "A-Z" lettere maiuscole
- "0-9" numeri
- "-" simbolo di sottrazione
- "_" underscore

Il nome usato nel nostro esempio è **ws001_lp**. Questo nome permette di identificare semplicemente la stampante collegata alla workstation **ws001**.

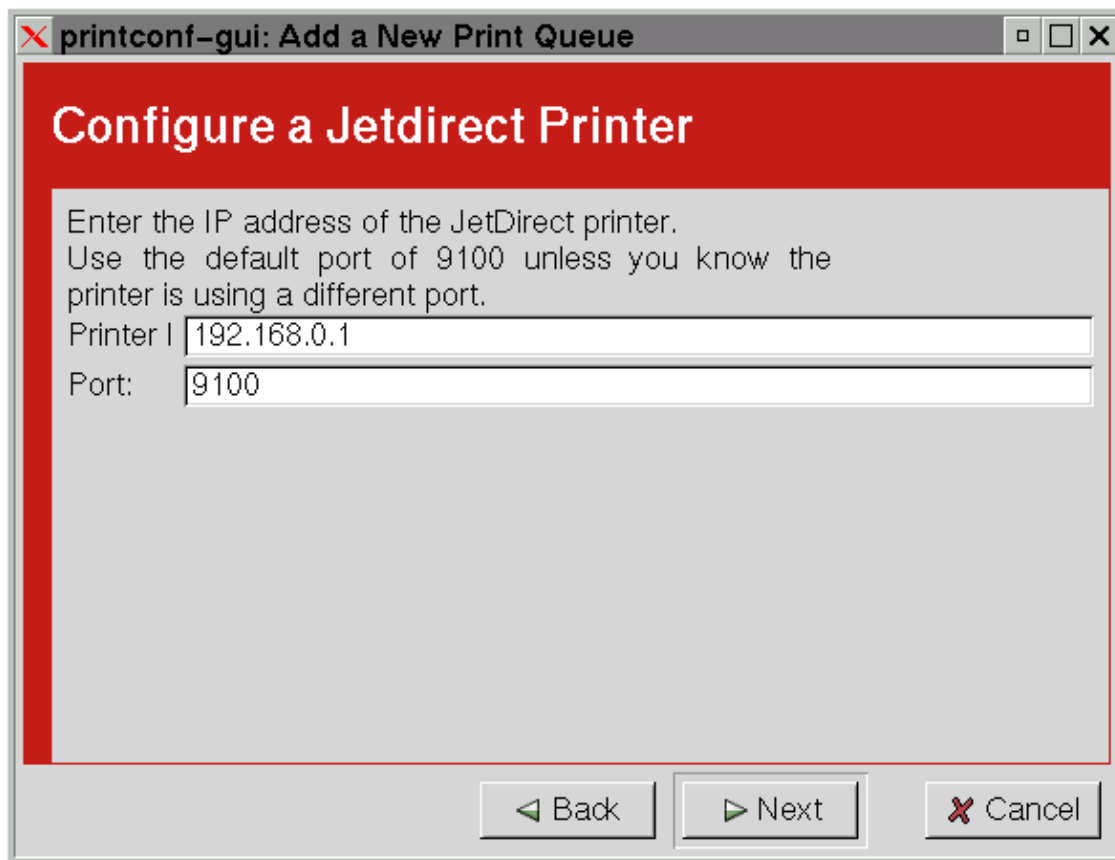


Figura 5–2. Informazioni di Printconf-gui

Ci sono due campi richiesti per comunicare con la stampante:

1. Indirizzo IP o hostname della workstation dove è collegata la stampante.
2. La porta TCP dove il servizio **lp_server** sta ascoltando per le stampe.

La prima stampante collegata alla workstation prende la porta **9100**, la seconda prende **9101**, la terza **9102**.

Capitolo 6. Soluzione dei problemi

Se seguendo tutte le istruzioni la workstation non dovesse partire, dovrai iniziare la fase di debug dell'installazione.

La prima cosa da fare è capire fino a che punto del processo di partenza è arrivata la tua workstation prima di incontrare problemi.

6.1. Problemi con l'immagine del dischetto di partenza

Quando esegui la partenza tramite il dischetto, dovresti vedere qualcosa come questo:

```
loaded ROM segment 0x0800 length 0x4000 reloc 0x9400
Etherboot 5.0.1 (GPL) Tagged ELF for [LANCE/PCI]
Found AMD Lance/PCI at 0x1000, ROM address 0x0000
Probing...[LANCE/PCI] PCnet/PCI-II 79C970A base 0x1000, addr 00:50:56:81:00:01
Searching for server (DHCP)...
<sleep>
```

L'esempio qui riportato mostra quello che dovresti aspettarti sullo schermo quando parti da un floppy disk. Se non vedi questi messaggi che segnalano la partenza di Etherboot, potresti avere un floppy difettoso, oppure l'immagine non è stata scritta correttamente.

Se invece vedi un messaggio simile al seguente, l'immagine che hai usato non è quella corretta per la tua scheda di rete.

```
ROM segment 0x0800 length 0x8000 reloc 0x9400
Etherboot 5.0.2 (GPL) Tagged ELF for [Tulip]
Probing...[Tulip]No adapter found
<sleep>
<abort>
```

Se invece arriviamo fino al punto in cui ci viene mostrato il MAC address della scheda di rete, il floppy è probabilmente a posto.

6.2. Problemi con il DHCP

Una volta che la scheda di rete è stata inizializzata, verranno inviate le richieste DHCP sulla rete, per cercare il server.

Se la workstation ottiene una risposta valida dal server DHCP, configurerà la scheda di rete di conseguenza. Puoi capire se tutto funziona quando vedi il settaggio di rete sul monitor. Qui c'è un esempio di quello che dovresti vedere:

```
ROM segment 0x0800 length 0x4000 reloc 0x9400
Etherboot 5.0.1 (GPL) Tagged ELF for [LANCE/PCI]
Found AMD Lance/PCI at 0x1000, ROM address 0x0000
Probing...[LANCE/PCI] PCnet/PCI-II 79C970A base 0x1000, addr 00:50:56:81:00:01
Searching for server (DHCP)...
<sleep>
Me: 192.168.0.1, Server: 192.168.0.254, Gateway 192.168.0.254
```

Se vedi una linea che inizia con 'Me:' seguito da un indirizzo IP, saprai che il sistema DHCP sta funzionando bene. Puoi procedere per vedere se funziona altrettanto bene il sistema TFTP.

Se invece vedi il messaggio seguente sulla workstation, seguito da molti <sleep>, qualcosa è andato storto. Potrebbe essere normale vedere uno o due <sleep>, ma non di più.

```
Searching for server (DHCP)...
```

Sapere cosa non sta funzionando non è sempre facile, ma qui ci sono un po' di cose da controllare per facilitarti il compito.

6.2.1. Controllare le connessioni

La workstation è fisicamente collegata alla stessa rete del server?

Quando la workstation è accesa, assicurati che le varie spie che indicano il collegamento fisico siano accese, sia sulla workstation stessa, sia su eventuali hubs.

Se workstation e server sono collegati direttamente, assicurati che il cavo utilizzato sia invertente (cross-over). Se invece usi hubs o switch, assicurati di usare cavi pin-to-pin, ossia non invertenti.

6.2.2. Il processo dhcpd è attivo?

Devi essere sicuro se il processo **dhcpd** sia attivo o meno sul server; possiamo saperlo in un paio di modi.

Normalmente, il processo **dhcpd** funziona in background, in ascolto sulla porta udp 67. Prova ad usare il comando **netstat** per vedere se qualcosa sta effettivamente ascoltando su quella porta:

```
netstat -an | grep ":67 "
```

Dovresti vedere qualcosa di simile a questo:

```
udp      0      0  0.0.0.0:67          0.0.0.0:*
```

La quarta colonna contiene l'indirizzo IP e la porta separati da un simbolo (:). Un indirizzo formato da quattro zeri ('0.0.0.0') indica che il processo ascolta su tutti gli indirizzi disponibili, cioè se si hanno 2 o più interfacce di rete, questo ascolta su tutte contemporaneamente.

Comunque, solo perché netstat dice che qualcosa ascolta sulla porta udp 67, non vuol dire che sia proprio il processo **dhcpd** a farlo. Potrebbe essere ad esempio il processo **bootpd**, anche se questo è piuttosto improbabile, poiché non viene più inserito nelle distribuzioni Linux da tempo.

Per essere sicuri che sia proprio **dhcpd** a girare, prova ad utilizzare il comando **ps**.

```
ps aux | grep dhcpd
```

Dovresti vedere qualcosa come:

```
root 23814 0.0 0.3 1676 820 ?        S 15:13 0:00 /usr/sbin/dhcpd
root 23834 0.0 0.2 1552 600 pts/0  S 15:52 0:00 grep dhcp
```

La prima riga indica che il processo **dhcpd** sta effettivamente girando, mentre la seconda rappresenta solamente il comando **grep** appena lanciato.

Se non vedi alcuna riga che dimostra che il `dhcpd` sta girando, devi controllare che il tuo server sia configurato per il runlevel 5, e che il **dhcpd** parta in questo runlevel. Su sistemi Redhat, puoi usare **ntsysv** per controllare che il **dhcpd** sia configurato per partire.

Puoi provare a far partire **dhcpd** a mano con questo comando:

```
service dhcpd start
```

Fa attenzione al suo output, in quanto potrebbe mostrare errori.

6.2.3. Controlla due volte la configurazione DHCP...

Nel file `/etc/dhcpd.conf` è presente una voce per la tua workstation?

Dovresti anche controllare bene il parametro 'fixed-address' nel file di configurazione, per essere sicuro che corrisponda al MAC address della tua scheda di rete.

6.2.4. E ipchains o iptables a bloccare le richieste?

6.2.4.1. Controllare ipchains

Lancia il seguente comando per vedere cosa esce:

```
ipchains -L -v
```

Se vedi qualcosa come:

```
Chain input (policy ACCEPT: 229714 packets, 115477216 bytes):
Chain forward (policy ACCEPT: 10 packets, 1794 bytes):
Chain output (policy ACCEPT: 188978 packets, 66087385 bytes):
```

significa che non è ipchains a bloccare le richieste.

6.2.4.2. Controllare iptables

Lancia il seguente comando:

```
iptables -L -v
```

Se vedi qualcosa come questo:

```
Chain INPUT (policy ACCEPT 18148 packets, 2623K bytes)
  pkts bytes target      prot opt in      out     source           destination
Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
  pkts bytes target      prot opt in      out     source           destination
Chain OUTPUT (policy ACCEPT 17721 packets, 2732K bytes)
  pkts bytes target      prot opt in      out     source           destination
```

significa che non è iptables a bloccare le richieste.

6.2.5. La workstation manda le richieste?

Prova a controllare il file `/var/log/messages` mentre la workstation esegue il boot; puoi farlo con questo comando:

```
tail -f /var/log/messages
```

Tail terra sottocchio il file di log mentre viene aggiornato in tempo reale.

```
server dhcpd: DHCPDISCOVER from 00:50:56:81:00:01 via eth0
server dhcpd: no free leases on subnet WORKSTATIONS
server dhcpd: DHCPDISCOVER from 00:50:56:81:00:01 via eth0
server dhcpd: no free leases on subnet WORKSTATIONS
```

Se vedi messaggi come questi, che dicono 'no free leases', significa che il **dhcpd** sta girando, ma non ha informazioni riguardo la workstation che richiede l'indirizzo IP.

6.3. Problemi con TFTP

Il software Etherboot utilizza TFTP per scaricare il kernel Linux dal server. Questo protocollo è semplice, ma alcune volte ci sono problemi a farlo funzionare.

Se vedi messaggi come questo:

```
Loading 192.168.0.254:/lts/vmlinuz-2.4.9-ltsp-5 |
```

con l'ultimo carattere sulla linea che cambia da `|` a `\` e ancora a `/`, formando quella che sembrerebbe una barra rotante, vuol dire che il kernel è in fase di scaricamento. Che a sua volta significa che TFTP sta funzionando bene.

Se invece non vedi la barra ruotare, ci sono dei problemi, che possono includere:

6.3.1. Il processo tftpd non è attivo

Su Redhat 7.1, tftp viene eseguito da **xinetd**. C'è uno script per la partenza chiamato `/etc/xinetd.d/tftp` che contiene informazioni per la partenza di **tftpd**

6.3.2. Il kernel non è dove tftpd se lo aspetta

Il kernel deve essere messo in un punto dove tftpd possa avere accesso. Se viene specificata l'opzione `-s` per il **tftpd**, qualunque file richiesto deve trovarsi all'interno della directory `/tftpboot`. Quindi, se l'opzione **filename** nel file `/etc/dhcpd.conf` è settata a `/lts/vmlinuz-2.4.9-ltsp-5`, il kernel deve trovarsi in `/tftpboot/lts/vmlinuz-2.4.9-ltsp-5`

6.4. Problemi con il filesystem di root NFS

Ci sono molte cose che potrebbero impedire al filesystem di root di essere montato; alcune sono:

6.4.1. Messaggio No init found

Se ottieni il seguente messaggio d'errore:

```
Kernel panic: No init found. Try passing init= option to kernel.
```

e probabile che tu stia montando la directory sbagliata, o che la directory `/opt/ltsp/i386` sia vuota.

6.4.2. Il server da un errore -13

Se ottieni:

```
Root-NFS: Server returned error -13 while mounting /opt/ltsp/i386
```

questo indica che la directory `/opt/ltsp/i386` non è inclusa nel file `/etc/exports`.

Prova a guardare nel file `/var/log/messages` per vedere se puoi scoprire qualcosa. Una linea come questa:

```
Jul 20 00:28:39 jamlap rpc.mountd: refused mount request from ws004
                for /opt/ltsp/i386 (/): no export entry
```

conferma i nostri sospetti che il file `/etc/exports` non sia corretto.

6.4.3. Problemi con i processi NFS (portmap, nfsd & mountd)

Il servizio NFS può essere difficile da correggere, ma capendo cosa deve essere regolato e quali strumenti abbiamo per farlo renderà la diagnosi più facile.

Ci sono tre processi che devono essere attivi per far sì che NFS funzioni correttamente: **portmap**, **nfsd** e **mountd**.

6.4.3.1. Il Portmapper (portmap)

Se ottieni un messaggio come questo:

```
Looking up port of RPC 100003/2 on 192.168.0.254
portmap: server 192.168.0.254 not responding, timed out
Root-NFS: Unable to get nfsd port number from server, using default
Looking up port of RPC 100005/2 on 192.168.0.254
portmap: server 192.168.0.254 not responding, timed out
Root-NFS: Unable to get mountd port number from server, using default
mount: server 192.168.0.254 not responding, timed out
Root-NFS: Server returned error -5 while mounting /opt/ltsp/i386
VFS: unable to mount root fs via NFS, trying floppy.
VFS: Cannot open root device "nfs" or 02:00
Please append a correct "root=" boot option
Kernel panic: VFS: Unable to mount root fs on 02:00
```

è molto probabile che il processo **portmap** non sia attivo. Puoi ottenere una conferma con il comando **ps**:

```
ps -e | grep portmap
```

Se il processo e attivo, dovresti vedere qualcosa come: like this:

```
30455 ?          00:00:00 portmap
```

Unaltro test consiste nellusare **netstat**. Il portmapper usa le porte TCP e UDP 111. Prova a lanciare:

```
netstat -an | grep ":111 "
```

Dovresti vedere qualcosa come:

```
tcp    0    0 0.0.0.0:111      0.0.0.0:*        LISTEN
udp    0    0 0.0.0.0:111      0.0.0.0:*
```

Se non vedi queste righe, il portmap non e attivo. Puoi farlo partire cosi:

```
/etc/rc.d/init.d/portmap start
```

Quindi, assicurati che il processo sia configurato per partire automaticamente allavvio del server, usando **ntsysv**.

6.4.3.2. I processi NFS e MOUNT (nfsd & mountd)

NFS ha 2 processi che devono essere attivi: **nfsd** e **mountd** e sono tutti e due lanciati dallo script `/etc/rc.d/init.d/nfs`.

Puoi utilizzare il comando **ps** per sapere se sono attivi.

```
ps -e | grep nfs
ps -e | grep mountd
```

Se questo mostra che uno o entrambi i processi non sono attivi, devi farli partire.

Solitamente, i processi vengono fatti partire richiamando il loro init script con lopzione **restart**, ma per qualche ragione lo script `/etc/rc.d/init.d/nfs` non fa ripartire **nfsd** se usato in questo modo (bug?). Quindi dovrai usare questa serie di comandi:

```
/etc/rc.d/init.d/nfs stop
/etc/rc.d/init.d/nfs start
```

Potresti vedere degli errori durante il comando di **stop**, ma e tutto normale. Il comando di **start** dovrebbe mostrare un **OK** come risultato.

Se i processi stanno girando ma NFS ancora non funziona, puoi verificare che essi si siano registrati correttamente con il portmapper usando il comando **rpcinfo**.

```
rpcinfo -p localhost
```

Dovresti vedere quanto segue:

```
program vers proto  port
100000    2    tcp    111  portmapper
100000    2    udp    111  portmapper
100011    1    udp    856  rquotad
100011    2    udp    856  rquotad
100005    1    udp    1104 mountd
100005    1    tcp    2531 mountd
100005    2    udp    1104 mountd
100005    2    tcp    2531 mountd
```

```
100003 2 udp 2049 nfs
```

Questo indica che **nfs** (nfsd) e **mountd** sono entrambi attivi ed hanno registrato se stessi con il portmapper.

6.5. Problemi con IXserver

Oh ragazzi, probabilmente la parte più difficile nel settare una workstation LTSP e riuscire a far funzionare correttamente IX server. Se la tua scheda video è relativamente nuova, ed è supportata da XFree86, e anche il tuo monitor è nuovo e può gestire molte risoluzioni anche grandi, settare X dovrebbe essere abbastanza semplice. Normalmente, se anche così X non funziona, probabilmente hai scelto IX server sbagliato per la tua scheda video.

Quando un X server non va con la tua scheda video, è abbastanza evidente: esso infatti può non partire proprio, oppure visualizzare cose non corrette.

Quando la workstation è pronta a far partire IX server, esegue lo script `/tmp/start_ws` che lo lancia in locale, con un'opzione **-query** puntata al server dove il display manager, come **XDM**, **GDM** o **KDM**, sta girando.

Siccome IX server viene fatto partire dallo script `start_ws`, che è a sua volta lanciato dal programma **init**, quando dovesse non funzionare, **init** cercherà di farlo ripartire. **init** ci riproverà per 10 volte e poi rinuncerà, perché penserà che questo processo riparte un po' troppo spesso per essere tutto normale. Dopo di che il messaggio d'errore dell'X server dovrebbe rimanere sul video.

Aspettare che IX server si chiuda e riparta per 10 volte può dare fastidio; un metodo semplice per evitarlo è quello di partire con un runlevel 3 che non fa partire X in automatico e una volta giunti al prompt **bash**, possiamo far partire X in manuale con il seguente comando:

```
sh /tmp/start_ws
```

LX server proverà a partire e, quando fallirà, ritornerai al prompt **bash**, dove potrai vedere per quale ragione si è fermato.

6.6. Problemi con il Display manager

Il display manager è il processo che gira sul server in attesa di un X server che si colleghi. Una volta stabilito il contatto, esso mostra una finestra di login sullo schermo dando la possibilità all'utente di entrare nel server.

I display managers più usati sono:

- **XDM** – È in giro da sempre ed è incluso con il sistema X standard.
- **GDM** – Il 'Gnome Display Manager'. Fa parte del pacchetto Gnome.
- **KDM** – Il 'KDE Display Manager'. Fa parte del sistema K Desktop.

La maggior parte delle recenti distribuzioni di GNU/Linux li includono tutti e tre.

6.6.1. Schermo grigio con un grosso cursore a forma di X

Questo significa che l'X server è attivo, ma non è stato in grado di contattare il display manager. Alcune ragioni possono essere:

1. Il display manager potrebbe non essere attivo

Su versioni recenti di Redhat (7.1 e superiori), il display manager è lanciato da **init**. Nel file `/etc/inittab` ce una linea che riporta:

```
x:5:respawn:/etc/X11/prefdm -nodaemon
```

Lo script **prefdm** determinerà quale display manager lanciare.

Il display manager di default dipende dai pacchetti installati. Se Gnome è installato, GDM sarà il display manager di default. Se invece Gnome non è installato, lo script `prefdm` controllerà se KDE è installato, lanciando in caso positivo KDM. Se nemmeno KDE non è presente, XDM sarà il display manager di default.

Usando il comando **netstat**, dovresti essere in grado di vedere il display manager in uso. Sul server, lancia il comando:

```
netstat -ap | grep xdmcp
```

Dovresti vedere un processo che ascolta sulla porta `xdmcp` (177).

```
udp        0      0  *:xdmcp          *: *              1493/gdm
```

Questo mostra chiaramente che **gdm** sta girando con un numero di processo 1493, e che sta ascoltando sulla porta `xdmcp`.

Se vedi una riga come quella, che dimostra che sicuramente il display manager è in ascolto, ed hai ancora problemi, devi essere sicuro che la workstation invii le richieste XDMCP in modo corretto.

Nel file `lts.conf`, devi avere una voce che indica l'indirizzo IP del server dove gira il display manager:

```
XDM_SERVER = 192.168.0.254
```

Naturalmente, questo è un esempio. Il tuo server potrebbe infatti avere un indirizzo diverso.

Se questa voce non è presente, verrà utilizzato l'indirizzo nella voce 'SERVER', se presente. Altrimenti, verrà usato l'indirizzo fisso **192.168.0.254**.

In qualunque modo esso sia specificato, devi assicurarti che l'indirizzo IP del server sia corretto e che corrisponda al server dove gira il display manager.

2. Il display manager potrebbe essere configurato per ignorare richieste da hosts remoti.

Se sei sicuro che il display manager sia attivo, è possibile che questo sia configurato per ignorare le richieste XDMCP provenienti da hosts remoti. Dovrai quindi controllare i file di configurazione del tuo display manager.

◆ XDM

La configurazione di default su Redhat e di disabilitare il login remoto tramite XDM. Lo script **Itsp_initialize** lo abiliterà per te, ma se non dovesse funzionare, dovresti controllare il file `/etc/X11/xdm/xdm-config`. Cerca una riga che assomigli a questa:

```
DisplayManager.requestPort: 0
```

Questa riga DEVE essere commentata per far in modo che XDM possa accogliere richieste esterne sulla porta 177.

Unaltro file altrettanto importante per XDM è `/etc/X11/xdm/Xaccess`, che deve contenere una linea che inizia con un asterisco *. Il file normalmente contiene già una riga con l'asterisco, ma viene commentata per motivi di sicurezza. Al solito, lo script **Itsp_initialize** dovrebbe scommentare la riga per te, ma se XDM non funziona, controlla anche questo file. La riga dovrebbe essere così:

```
* #any host can get a login window
```

◆ KDM

Le nuove versioni di KDM hanno un file chiamato **kdmrc**. A seconda della distribuzione di Linux, questo file può trovarsi in vari posti; Per Redhat 7.2, è in `/etc/kde/kdm/kdmrc`. Per le altre distribuzioni, usa il comando **locate** per trovarlo.

La voce che controlla l'accesso remoto è nella sezione **[Xdmcp]**. Devi essere sicuro che la voce **Enable** sia settata su **true**.

Versioni meno recenti di KDM usano i files di configurazione di XDM, situati in `/etc/X11/xdm`.

◆ GDM

GDM usa uno schema di files diverso; sono tutti nella directory `/etc/X11/gdm`.

Il file principale dove guardare è `gdm.conf`; cerca la sezione **[xdmcp]** dove dovresti vedere una voce chiamata **Enable**. Questa deve essere settata a 1 o true, a seconda della versione di GDM. Ecco un esempio:

```
[xdmcp]
Enable=true
HonorIndirect=0
MaxPending=4
MaxPendingIndirect=4
MaxSessions=16
MaxWait=30
MaxWaitIndirect=30
Port=177
```

Nota la linea 'Enable=true'. Vecchie versioni di GDM usano 0 e 1 per Disable e Enable.

3. Se il Display manager sta sicuramente girando, e sta ascoltando per le richieste delle workstations, potrebbe semplicemente non essere in grado di mappare gli indirizzi IP con i nomi degli host. La

workstation deve infatti essere elencata o nel file `/etc/hosts` oppure nella tavola del server DNS.

Capitolo 7. I kernel

Ci sono alcune scelte da fare riguardo al kernel che dovrà girare sulla workstation. Devi infatti decidere se vuoi usare uno di quelli standard distribuiti da LTSP.org o uno compilato da te. E devi anche decidere se vuoi includere anche la schermata grafica con barra di caricamento fornita dalla **Linux Progress Patch (LPP)**.

7.1. Kernels standard distribuiti con LTSP

Il pacchetto dei kernel distribuito con LTSP include due kernels: uno ha la Linux Progress Patch già applicata, l'altro no.

Entrambi invece hanno la patch per lo swap su NFS applicata.

7.2. Costruire il tuo kernel

Ci sono due vie per configurare un kernel per LTSP; il metodo di default e quello di usare l'Initial Ram Disk', o `initrd`. L'immagine `initrd` è un piccolo filesystem appeso alla fine del kernel. Esso viene caricato in memoria e una volta che il kernel è partito, montera questo filesystem come root. Ci sono dei vantaggi nell'utilizzo di **initrd**: per prima cosa possiamo compilare i drivers delle schede di rete come moduli e caricare il modulo corretto durante la partenza, permettendo di creare un solo kernel per tutte le schede di rete; l'altro vantaggio è che possiamo contattare il server DHCP usando un programma a se stante, e non il kernel in modo diretto; questo ci permette di avere maggior controllo sulle opzioni ricevute dal server, mantenendo anche il kernel un po' più snello. L'altro modo di configurare il kernel è ovviamente fare a meno di `initrd`; questo implica che i drivers per le schede di rete siano inclusi dentro il kernel e inoltre che le opzioni IP-Autoconfig e "Root filesystem on NFS" siano attivate. Il vantaggio di questa soluzione è che l'immagine da scaricare da rete è più piccola, data l'assenza del filesystem `initrd`, e l'operazione di boot è un po' più veloce. Dopo la fase di boot, non ci sarà differenza nel funzionamento delle workstations.

The standard kernel for LTSP includes an Initial Ramdisk (`initrd`) that takes care of detecting the network card, and making a user-space DHCP request. A major goal for the image was to make it as small as possible. So, we chose the uClinux libc replacement library, and busybox for the utilities that we need during the boot.

Il kernel standard per LTSP include un Initial Ramdisk (`initrd`) che riconosce automaticamente la scheda di rete e il programma per le richieste DHCP. Uno dei maggiori scopi di `initrd` era di essere il più piccolo possibile, quindi abbiamo scelto la libreria libc uClinux libc, e busybox per le utility che ci servono per fare il boot.

7.2.1. Scaricare i sorgenti del kernel

Quando si costruisce un proprio kernel, è bene partire dai sorgenti originali scaricati da **ftp.kernel.org**. Il motivo di questa scelta è che le distribuzioni, come RedHat, applicano molte patch ai loro kernel, che fa sì che questi non siano più uguali a quelli ufficiali.

Scarica i sorgenti della versione che preferisci e salvali nella directory `/usr/src`. I kernel sono situati nella directory `/pub/linux/kernel` del server `ftp.kernel.org`. Avrai bisogno di una versione recente della serie 2.4.x series del kernel, perché hai bisogno del supporto per il filesystem **devfs**.

Inoltre, se vuoi avvalerti dei vantaggi forniti dallo swap su NFS o dalla Linux Progress Patch (LPP), devi essere sicuro di scaricare le versioni corrette di queste patch per il tuo kernel. Mentre viene scritto questo

documento, il kernel 2.4.9 e il più recente a supportare queste caratteristiche.

Nel nostro esempio, useremo il kernel 2.4.9. Il path completo è

```
ftp://ftp.kernel.org/pub/linux/kernel/v2.4/linux-2.4.9.tar.bz2
```

Scompatta i sorgenti nella directory `/usr/src`; devi fare attenzione quando scompatti, perché verrà creata una directory chiamata `linux`. Potresti avere già una directory con questo nome che contiene un'altra serie di sorgenti, perciò se non vuoi rovinarli, rinomina la tua directory prima di procedere.

I sorgenti che abbiamo scaricato sono stati compressi con l'utilità **bzip2**; quindi, dobbiamo scompattarli prima di mandarli al programma **tar**. Puoi usare il seguente comando:

```
bunzip2 <linux-2.4.9.tar.bz2 | tar xf -
```

Quando l'operazione termina, avrai ottenuto una directory chiamata `linux` contenente l'intero set di sorgenti. A questo punto, mi piace rinominare la directory in qualcosa di più significativo.

```
mv linux linux-2.4.9
```

Una volta rinominata, entriamo nella directory:

```
cd linux-2.4.9
```

Mi piace anche modificare il `Makefile` prima di configurare il nuovo kernel: vicino all'inizio del file, c'è una variabile chiamata **EXTRAVERSION**. Io setto questa variabile a `'ltsp-1'`, in modo tale che la versione del kernel risulti essere `'2.4.9-ltsp-1'`; in questo modo il kernel è più riconoscibile. La parte iniziale del file `Makefile` dovrebbe apparire così:

```
VERSION = 2
PATCHLEVEL = 4
SUBLEVEL = 9
EXTRAVERSION = -ltsp-1

KERNELRELEASE=$(VERSION) . $(PATCHLEVEL) . $(SUBLEVEL) $(EXTRAVERSION)
```

7.2.2. Patches per il kernel

Dopo aver scompattato il kernel, potresti voler applicare alcune patch, come ad esempio `INFS Swap patch` o la `Linux Progress Patch`; queste DEVONO essere applicate prima di configurare il kernel.

7.2.2.1. NFS Swap patch

LNFS Swap patch permetterà al kernel della workstation di usare il server tramite NFS per lo swap file. Nonostante sia consigliabile avere sufficiente RAM per il funzionamento senza ricorrere allo swap, può essere in alcuni casi impossibile aggiungere RAM, come su PC vecchi. Quindi questa caratteristica può rendere alcuni PC obsoleti in grado di funzionare.

Se la directory corrente è `/usr/src/linux-2.4.9`, e la patch si trova in `/usr/src`, devi dare questo comando per verificare la patch:

```
patch -p1 --dry-run <../linux-2.4.9-nfs-swap.diff
```


Questo testerà la patch, per essere sicuri che possa essere applicata. Se non ci sono errori, puoi applicarla tramite lo stesso comando senza l'opzione **--dry-run**.

```
patch -p1 <../linux-2.4.9-nfs-swap.diff
```

7.2.2.2. Linux Progress Patch (LPP)

La Linux Progress Patch (LPP) ti permetterà di configurare un logo grafico da visualizzare durante la partenza; i normali messaggi del kernel sono rediretti ad un altro schermo tty, e alcune istruzioni speciali sono aggiunte allo script di boot per far muovere correttamente la barra di avanzamento.

Come per NFS Swap patch, puoi testare la patch prima di applicarla:

```
patch -p1 --dry-run <../lpp-2.4.9
```

Se il test non rileva errori, puoi realmente applicare la patch:

```
patch -p1 <../lpp-2.4.9
```

7.2.3. Configurare le opzioni del kernel

Puoi ora lanciare il programma di configurazione che preferisci:

- make xconfig

Versione X Windows del configuratore (grafico).

- make menuconfig

Versione basata sulle librerie curses del configuratore (testo con menu).

- make config

Versione su singola riga del configuratore.

7.2.3.1. Configurazione del kernel con initrd

La configurazione del kernel per utilizzare initrd richiede le seguenti opzioni:

- File systems -> /dev filesystem support

Il supporto per il filesystem /dev deve essere abilitato. NON attivare 'Automatically mount at boot', in quanto il mount viene eseguito dallo script /linuxrc.

- Block devices -> RAM disk support

Le workstations LTSP richiedono che il kernel supporti un RAM disk. Questa opzione è nella sezione 'Block devices'.

- Block devices -> Initial RAM disk (initrd) support

Anche questo deve essere abilitato.

- Processor type and features → Processor family

Devi essere sicuro che il kernel che stai costruendo sia in grado di essere eseguito sul processore della tua workstation: puoi scegliere l'architettura con questo parametro. Dovresti anche disattivare il supporto SMP, a meno che tu non abbia una macchina multiprocessore.

- File systems → Network file systems → NFS Client support

La workstation monterà il filesystem di root tramite NFS, quindi il supporto NFS client è richiesto.

Questo dovrebbe essere sufficiente. Puoi anche disattivare altre opzioni del kernel in modo da ridurre le dimensioni.

7.2.3.2. Configurazione del kernel senza initrd

La configurazione del kernel senza initrd differisce da quella precedente in alcuni punti:

- Block devices → RAM disk support

Le workstation LTSP richiedono il supporto del RAM disk.

- Block devices → Initial RAM disk (initrd) support

Questo deve essere disabilitato.

- Networking options → IP:kernel level autoconfiguration

Questo deve essere abilitato: dirà infatti al kernel di configurare automaticamente la scheda di rete eth0, in base ai parametri specificati tramite la riga di comandi del kernel.

Non è necessario specificare le opzioni DHCP, BOOTP o RARP perché la bootrom Etherboot ha già fatto una richiesta al DHCP o al BOOTP, e passa i dati ottenuti sulla riga di comandi del kernel. Questo risparmia al kernel la fatica di eseguire un'ulteriore sua richiesta.

- Network device support → Ethernet (10 or 100Mbit)

Quando non usi initrd, devi scegliere una driver per la scheda di rete che utilizzi; esso DEVE essere compilato nel kernel, NON COME MODULO, perché la scheda di rete deve essere funzionante prima che sia montato il filesystem di root. Questa è una differenza grande nel funzionamento tra questa configurazione e quella con initrd.

- File systems → /dev filesystem support

A partire dalla versione 2.09pre2 di LTSP, il supporto per **devfs** è richiesto, indipendentemente dall'uso o meno di initrd.

- File systems → Automatically mount at boot

Quando non si usa `initrd`, il filesystem `/dev` deve essere montato in automatico dal kernel, durante l'operazione di boot. Rispondi Y qui.

- File systems -> Network file systems -> NFS Client support

La workstation monterà il proprio filesystem di root tramite NFS, quindi NFS client support deve essere abilitato.

7.2.4. Compilare il kernel

Per rendere le cose un po' più semplici, una copia del file `.config` viene inclusa nel pacchetto `ltsp_initrd_kit`; puoi copiarlo nella directory `/usr/src/linux-2.4.9`.

Una volta selezionate le opzioni del kernel, lo devi compilare: i seguenti comandi devono essere eseguiti:

```
make dep
make clean
make bzImage
make modules
make modules_install
```

Puoi metterli tutti su una riga in questo modo:

```
make dep && make clean && make bzImage && make modules && make modules_install
```

La doppia e commerciale (&) significa che i comandi devono essere eseguiti uno dopo l'altro se quello precedente viene completato con successo.

Quando la compilazione è terminata, il nuovo kernel si troverà in `/usr/src/linux-2.4.7/arch/i386/boot/bzImage`.

7.2.5. Taggare il kernel per Etherboot

Per fare in modo che Etherboot possa gestire il kernel, quest'ultimo deve essere preparato: questa operazione è chiamata "Taggare il kernel". Questa operazione aggiungerà del codice al kernel e sarà eseguita prima che il controllo sia passato al kernel stesso. Lo strumento per taggare il kernel è chiamato **'mknbi-linux'**.

Il pacchetto `ltsp_initrd_kit` include uno script chiamato **buildk** che include tutti i comandi necessari a preparare un kernel per il boot da rete.

Capitolo 8. Il file `lts.conf`

Quando abbiamo disegnato LTSP, uno dei problemi che sapevamo di dover affrontare era la differenza di hardware tra le varie workstations. Sicuramente, qualsiasi combinazione di processore, scheda di rete e scheda video disponibile oggi, non lo sarà più tra 3 mesi, quando vorremo aggiungere altre workstations.

Quindi, abbiamo pensato ad un sistema per specificare la configurazione della singola workstation; questa configurazione è memorizzata nel file `lts.conf` che risiede nella directory `/opt/ltsp/i386/etc`.

Il formato di questo file precede settaggi di default e settaggi specifici per ogni workstation. Se tutte le tue workstation fossero uguali, potresti usare solo la sezione di default.

8.1. Esempio di `lts.conf`

Ecco un esempio del file `lts.conf`:

```
[Default]
SERVER          = 192.168.0.254
X_MOUSE_PROTOCOL = "PS/2"
X_MOUSE_DEVICE  = "/dev/psaux"
X_MOUSE_RESOLUTION = 400
X_MOUSE_BUTTONS = 3
USE_XFS         = N
RUNLEVEL        = 5

[ws001]
XSERVER         = auto
X_MOUSE_PROTOCOL = "Microsoft"
X_MOUSE_DEVICE  = "/dev/ttyS1"
X_MOUSE_RESOLUTION = 50
X_MOUSE_BUTTONS = 3
X_MOUSE_BAUD    = 1200

[ws002]
XSERVER         = XF86_Mach64

[ws003]
RUNLEVEL        = 3
```

8.2. Parametri disponibili in `lts.conf`

8.2.1. Parametri generali

Commenti

I commenti partono tutti con il simbolo del cancelletto `#` e continuano fino al termine della riga.

LTSP_BASEDIR

Questo indica dove è situato il filesystem di root di LTSP; il default è `/opt/ltsp`

SERVER

Questo è il server usato come `XDM_SERVER`, `TELNET_HOST`, `XFS_SERVER` e `SYSLOG_HOST`, se nessuno di questi è a sua volta specificato esplicitamente. Se hai un solo server per tutti questi

servizi, non dovrai fare altro che specificarne l'indirizzo in questo parametro. Se questo parametro non è specificato, verrà utilizzato il valore **192.168.0.254**.

SYSLOG_HOST

Se vuoi inviare i messaggi di log ad una macchina differente dal server, puoi specificare il suo indirizzo qui. Se questo parametro non viene specificato, viene utilizzato il valore del parametro 'SERVER' descritto sopra.

NFS_SERVER

Specifica l'indirizzo IP del server NFS usato per montare la directory /home. Il default è lo stesso indirizzo indicato nel parametro **SERVER**.

USE_NFS_SWAP

Setta questo a **Y** se vuoi usare lo swap su NFS; il default è **N**

SWAPFILE_SIZE

Qui puoi controllare la dimensione del file di swap; il default è **64m**.

SWAP_SERVER

Il file di swap può essere messo su qualsiasi server sulla rete che sia in grado di gestirlo: puoi specificare qui il suo indirizzo. Il default è lo stesso valore del parametro **NFS_SERVER**.

NFS_SWAPDIR

La directory sul server dello swap che viene esportata tramite NFS; il default è /var/opt/ltsp/swapfiles. Assicurati che la directory sia esportata nel file /etc/exports.

TELNET_HOST

Se la workstation è configurata per avere un'interfaccia testuale, questo parametro definisce l'host a cui connettersi tramite telnet. Se questo parametro non viene settato, verrà usato l'indirizzo nel parametro **SERVER**.

DNS_SERVER

Viene usato per costruire il file resolv.conf.

SEARCH_DOMAIN

Viene usato per costruire il file resolv.conf.

MODULE_01* fino a *MODULE_10

Possono essere specificati fino a 10 moduli da caricare tramite questi parametri; quindi inserire l'intera riga di comando che useresti per caricare i moduli con insmod. Per esempio:

```
MODULE_01 = uart401.o
MODULE_02 = sb.o io=0x220 irq=5 dma=1
MODULE_03 = opl3.o
```

Se il valore settato è un nome di file completo di path, verrà usato **insmod** per caricare il modulo. Altrimenti, verrà usato **modprobe**.

RAMDISK_SIZE

Quando la workstation parte, crea un ramdisk e lo monta sulla directory /tmp. Puoi controllare la grandezza di questo filesystem con questo parametro, specificando la dimensione in kbytes (1024

bytes). Per creare un ramdisk di 1 megabyte, usa **RAMDISK_SIZE = 1024**

Se cambi questo valore, dovrai cambiare la dimensione del ramdisk anche nel kernel: questa può essere scelta in fase di compilazione, oppure specificata quando tagghi il kernel con `mknbi–linux`.

Il valore di default è 1024 (1 mb)

RCFILE_01 fino a RCFILE_10

Altri script possono essere eseguiti dallo script principale `rc.local`; dovrai solo metterli nella directory `/etc/rc.d`, e specificarne il nome in uno di questi parametri.

SOUND

Se il pacchetto LTSP Sound è installato, devi mettere **Y** qui in modo che lo script **rc.sound** venga eseguito per configurare la scheda sonora e il processo che la controlla. Il default è **N**.

8.2.2. Parametri del sistema X–Windows

XDM_SERVER

Se vuoi puntare XDM ad una macchina diversa del server di default, specifica il suo indirizzo in questo parametro. Se non specifichi nulla, verrà usato il parametro 'SERVER' descritto precedentemente.

XSERVER

Questo parametro definisce quale X server deve partire sulla workstation; per le schede video PCI o AGP, questo parametro non dovrebbe essere richiesto, poiché lo script `rc.local` dovrebbe riconoscere in automatico la scheda. Puoi anche settare questo parametro ad **auto** per indicare che la scheda video dovrebbe essere riconosciuta automaticamente.

Per le schede video ISA, o per specificare un Xserver, puoi usare il nome specifico del driver o dell'Xserver da usare.

Se questo valore inizia con **XF86_**, verrà utilizzato XFree86 3.3.6. Altrimenti verrà lanciato XFree86 4.1.x. Il default è **auto**.

X_MODE_0 fino a X_MODE_2

Qui possono essere configurate fino a 3 risoluzioni video per la workstation. Questo parametro accetta due tipi di valori: una risoluzione oppure una modeline completa.

```
X_MODE_0 = 800x600
OR
X_MODE_0 = 800x600 60.75 800 864 928 1088 600 616 621 657 -HSync -VSync
```

Se non viene specificata alcuna voce `X_MODE_x`, verranno usate quelle interne e verranno abilitate le risoluzioni 1024x768, 800x600 e 640x480.

Se una o più linee `X_MODE_x` vengono configurate, andranno a sovrascrivere completamente quelle interne.

X_MOUSE_PROTOCOL

Qui puo essere messo un qualunque valore che funzioni con il protocollo XFree86 Pointer Protocol; tipicamente questi valori sono "Microsoft" e "PS/2". Il default e settato a "**PS/2**".

X_MOUSE_DEVICE

Questo e il nodo dove viene collegato il mouse (nodo = file in /dev). Se il mouse e seriale, qui dovrebbe essere riportata una porta seriale, come **/dev/ttyS0** o **/dev/ttyS1**. Se invece e PS/2, questo valore sara **/dev/psaux**. Il default e **/dev/psaux**.

X_MOUSE_RESOLUTION

Questo parametro corrisponde al valore 'Resolution' nel file **XF86Config**. Un valore tipico per un mouse seriale e **50**, mentre per uno PS/2 e **400**. Il default e **400**.

X_BUTTONS

Qui possiamo dire al sistema quanti pulsanti ha il nostro mouse; normalmente viene settato a **2** o **3**. Il default e **3**.

X_MOUSE_EMULATE3BTN

Questo dice allX server di emulare il terzo tasto del mouse su un mouse s 2 pulsanti, cliccando assieme il tasto di sinistra piu quello di destra. Il valore di default e **N**.

X_MOUSE_BAUD

Per mouse seriali, e la velocita di comunicazione con la seriale. Il default e **1200** (baud).

X_COLOR_DEPTH

Questo e il numero di bit da usare per la profondita del colore. Valori accettati sono **8, 15, 16, 24** e **32**. 8 bits dara 256 colori, 16 ne dara 65536, 24 ne dara 16 milioni e 32 bits ne dara 4.2 miliardi! Non tutti gli X servers supportano tutti questi valori. Il default e **16**

USE_XFS

Hai la possibilita di usare lX Font Server (XFS) o di caricare le fonts attraverso il filesystem NFS. Il font server da la possibilita di mantenere tutte le fonts in un unico punto sul server, ma ci sono stati problemi quando le workstations aumentano di numero fino ad arrivare ad una quarantina circa. I due valori accettati qui sono **Y** e **N**. Il default e **N**. Se vuoi usare il font server, puoi specificarne la macchina server usando il parametro **XFS_SERVER**.

XFS_SERVER

Se usi lX Font Server per distribuire le fonts, puoi usare questo parametro per specificare lindirizzo IP della macchina che funge da font server. Se non la specifichi, verra usato lindirizzo specificato nel parametro **SERVER**.

X_HORZSYNC

Questo setta il parametro **HorizSync** di XFree86. Il default e "**31-62**".

X_VERTREFRESH

Questo setta il parametro **VertRefresh** di XFree86. Il default e "**55-90**".

XF86CONFIG_FILE

Se vuoi crearti un file XF86Config completo, puoi metterlo nella directory **/opt/ltsp/i386/etc**. Poi, metti il nome di questo file in questo parametro. Ad esempio:

```
XF86CONFIG_FILE = XF86Config.ws004
```

8.2.3. Parametri per il touch screen

USE_TOUCH

Se collegherai un touch screen alla workstation, puoi abilitarlo mettendo **Y**: in questo caso, altri parametri verranno esaminati per configurare altri particolari del tuo touch screen. Il default è **N**.

X_TOUCH_DEVICE

Un touch screen lavora come un mouse, e normalmente viene collegato ad una porta seriale: qui puoi specificare quale porta usare; ad esempio potresti mettere **/dev/ttyS0**. Non ce un valore predefinito.

X_TOUCH_MINX

Valore di calibrazione per i touch screen EloTouch. Default a **433**.

X_TOUCH_MAXX

Valore di calibrazione per i touch screen EloTouch. Default a **3588**.

X_TOUCH_MINY

Valore di calibrazione per i touch screen EloTouch. Default a **569**.

X_TOUCH_MAXY

Valore di calibrazione per i touch screen EloTouch. Default a **3526**.

X_TOUCH_UNDELAY

Valore di calibrazione per i touch screen EloTouch. Default a **10**.

X_TOUCH_RPTDELAY

Valore di calibrazione per i touch screen EloTouch. Default a **10**.

8.2.4. Parametri per le applicazioni locali

LOCAL_APPS

Se vuoi abilitare le applicazioni locali sulla workstation, setta questa variabile a **Y**; dovrai comunque configurare molti altri aspetti del tuo server per far funzionare le applicazioni in locale: per ulteriori informazioni, da unocchiata alla sezione Applicazioni locali. Il valore di default è **N**.

NIS_DOMAIN

Se decidi di abilitare *LOCAL_APPS*, avrai bisogno di un server NIS sulla rete. Questo parametro serve a specificare il nome del dominio NIS, e deve ovviamente essere uguale a quello del server; inoltre, questo dominio **NON** ha niente a che vedere con quello DNS. Il default è **ltsp**.

NIS_SERVER

Metti qui l'indirizzo IP del server NIS, se non vuoi che la workstation mandi una richiesta broadcast per scoprirlo.

8.2.5. Parametri per la tastiera

Tutti i files di supporto per la tastiera sono inclusi nella directory `/opt/ltsp/i386`, quindi bisogna solo configurare correttamente XFree86 per avere una tastiera funzionante in modo corretto; sono disponibili diverse opzioni per fare questo.

I valori per i seguenti parametri sono presi dalla documentazione di XFree86, quindi qualunque valore valido per XFree86 puo essere utilizzato.

Vorremmo anche aggiungere informazioni piu precise per ogni configurazione internazionale: se riesci a configurare la tua tastiera in modo ottimale, manda due righe al gruppo principale di ltsp: saranno molto apprezzate!

XkbTypes

Il default e la parola '**default**'.

XkbCompat

Il default e la parola '**default**'.

XkbSymbols

The default value for this is '**us(pc101)**'.

XkbModel

Il default e' '**pc101**'.

XkbLayout

Il default e' '**us**'.

8.2.6. Parametri per la configurazione della stampante

Possono essere connesse alla workstation fino a tre stampanti; qualsiasi combinazione di stampanti parallele e seriali puo essere configurata tramite il file **Its.conf**:

PRINTER_0_DEVICE

Il dispositivo su cui e collegata la stampante, come `/dev/lp0`, `/dev/ttyS0` o `/dev/ttyS1`.

PRINTER_0_TYPE

Il tipo di collegamento della stampante: '**P**' per parallela e '**S**' per seriale.

PRINTER_0_PORT

La porta TCP/IP da utilizzare; il default e '**9100**'

PRINTER_0_SPEED

Se la stampante e seriale, puoi settarne qui la velocita in baud; il default e '**9600**'.

PRINTER_0_FLOWCTRL

Sempre per le stampanti seriali, stabilisce il tipo di controllo di flusso: '**S**' per quello software (XON/XOFF), o '**H**' per quello hardware (CTS/RTS). Se non viene specificato, verra usato '**S**'.

PRINTER_0_PARITY

Per le stampanti seriali, stabilisce la parità da usare: 'E'–Even, 'O'–Odd o 'N'–Nessuna. Se non specificato, verrà usata 'N'.

PRINTER_0_DATABITS

Per le stampanti seriali, stabilisce il numero di bits da usare: '5', '6', '7' e '8' sono le scelte possibili. Se non specificato, verrà usato '8'.

PRINTER_1_DEVICE

Dispositivo a cui è collegata la seconda stampante

PRINTER_1_TYPE

Tipo della seconda stampante

PRINTER_1_PORT

La seconda porta TCP/IP da usare

PRINTER_1_SPEED

Velocità della seconda stampante (seriale)

PRINTER_1_FLOWCTRL

Second printer flow control (serial)

PRINTER_1_PARITY

Controllo di flusso della seconda stampante (seriale)

PRINTER_1_DATABITS

Bits della seconda stampante (serial)

PRINTER_2_DEVICE

Dispositivo a cui è collegata la terza stampante

PRINTER_2_TYPE

Tipo della terza stampante

PRINTER_2_PORT

Porta TCP/IP della terza stampante

PRINTER_2_SPEED

Velocità della terza stampante (seriale)

PRINTER_2_FLOWCTRL

Controllo di flusso della terza stampante (seriale)

PRINTER_2_PARITY

Third printer parity (seriale)

PRINTER_2_DATABITS

Bits della terza stampante (seriale)

Capitolo 9. Local Applications

In un ambiente LTSP, hai la possibilità di scegliere se far girare le applicazioni localmente sulla workstation, o sul server.

Il modo più semplice, e di molto, è quello di far girare le applicazioni sul server, cioè usando la CPU e la memoria del server, e il monitor, tastiera e mouse della workstation.

Questa è una delle principali caratteristiche del sistema X Windows, dove la workstation lavora come un terminale grafico.

Per far girare le applicazioni in locale, la workstation deve conoscere alcune informazioni sull'utente:

- Id dell'utente
- Gruppo primario di appartenenza dell'utente
- Home directory dell'utente

LTSP si affida al Network Information Service – NIS, (un tempo chiamato *Yellow Pages*) per rendere disponibili queste informazioni alle workstation tramite la rete.

9.1. Vantaggi delle applicazioni che girano in locale

Ci sono alcuni vantaggi facendo girare le applicazioni in locale.

- Si riduce il carico sul server: in configurazioni di rete grandi, dove si usano applicazioni voraci di memoria come Netscape, far girare le applicazioni sulla workstation dà prestazioni migliori, ammesso che la workstation sia abbastanza performante.
- Applicazioni che dovessero bloccarsi non influenzano gli altri utenti.
- Il supporto sonoro è molto più semplice da configurare, quando l'applicazione che lo produce gira sulla stessa macchina dove lo si ascolta.

9.2. Svantaggi delle applicazioni che girano in locale

Configurare le applicazioni locali richiede molto di più.

- Requisiti più pesanti sulla workstation, che deve avere più RAM e una CPU più performante: un buon punto di partenza sono 64MB di memoria.
- NIS Per far girare le applicazioni in locale, prima devi farti riconoscere dalla workstation; questo richiede una qualche forma di autenticazione delle password. È stato scelto il metodo NIS.
- Ulteriori directory devono essere esportate dal server attraverso NFS.
- Le applicazioni partono più lentamente, poiché vengono lette via rete tramite NFS, accrescendo il traffico di rete. Oltretutto, visto che ogni applicazione gira su una CPU diversa, non puoi avvalerti della capacità di Linux, e più in generale di Unix, o *nix, di condividere parti di codice tra più sessioni dello stesso programma, che riduce notevolmente i tempi di attesa della seconda e delle successive invocazioni del programma stesso.

9.3. Configurazione del server per le applicazioni locali

9.3.1. Parametri in `lts.conf`

Alcuni parametri devono essere settati nel file `lts.conf`:

LOCAL_APPS

Questo deve essere settato a **Y**: questo porterà la workstation a fare le seguenti cose durante la partenza:

1. Viene montata la directory `/home` del server tramite NFS.
2. Viene creato il file `/var/yp/nicknames` sulla workstation.
3. Viene lanciato il processo **portmapper** sulla workstation.
4. **xinetd** viene fatto partire sulla workstation.
5. Viene creato il file `/etc/yp.conf` sulla workstation.
6. Viene lanciato il comando **domainname** con il valore **NIS_DOMAIN** dal file `lts.conf`.
7. Viene fatto partire il processo **ypbind** sulla workstation.

NIS_DOMAIN

Con il NIS, tutti i nodi della rete che vogliono essere associati ad un server NIS, devono appartenere allo stesso dominio NIS (che non è in alcun modo collegato al dominio DNS). Puoi usare questa voce per specificare il nome del dominio NIS a cui apparterrà la workstation.

NIS_SERVER

Il NIS cercherà o di collegarsi con un server NIS specifico, oppure invierà sulla rete una richiesta broadcast cercando un server disponibile. Se vuoi usare un server specifico, metti qui il suo indirizzo IP.

9.3.2. Network Information Service – NIS

Il NIS è un servizio di tipo Client/Server. Sul server, c'è un servizio chiamato **ypserv** che accetta richieste dalle workstation.

Sulle workstation, c'è un processo chiamato **ypbind**; quando questa ha bisogno di informazioni sull'utente, come una password o la sua home directory, userà **ypbind** per stabilire una connessione con **ypserv** sul server.

Se hai già un server NIS nella tua rete, è inutile configurare anche il server LTSP come NIS server; potrai semplicemente configurare le voci `NIS_DOMAINNAME` e `NIS_SERVER` nel file `lts.conf` per rispecchiare la tua configurazione.

Se invece non hai già un server NIS, devi configurarne uno.

Per avere informazioni più dettagliate su come settare un server NIS, leggi *IHOWTO* al sito LDP (Linux Documentation Project) chiamato *The Linux NIS(YP)/NIS/NIS+ HOWTO*. Fai riferimento alla sezione altre risorse di informazioni alla fine di questo documento.

9.4. Configurazione delle applicazioni

Per poter far funzionare un'applicazione sulla workstation, devi metterne tutti i componenti in un punto in cui la workstation possa vederli.

Con le versioni precedenti di LTSP (2.08 e precedenti), molte directory venivano esportate dal server e montate dalla workstation, come /bin, /usr/bin, /lib e /usr.

Il problema, in questo caso, era che la workstation e il server dovevano avere per forza la stessa architettura, perché i files condivisi potevano essere compilati per un processore diverso da quello della workstation; anche piccole differenze, come un server Pentium II (i686) e una workstation Pentium (i586), potevano essere un problema, poiché alcune librerie i686 non funzionavano con architetture i386, i486 o i586.

Quindi, il modo migliore per procedere è quello di avere una serie completa di files binari e librerie di cui la workstation avrà bisogno, indipendentemente dai files del server.

Configurare un'applicazione per l'esecuzione in locale vuol dire mettere tutti i pezzi nel posto giusto. Uno dei pacchetti LTSP scaricabili, il Local netscape package, è un bellesempio di applicazione locale, che mette molti files in /opt/ltsp/i386/usr/local/netscape. Cose come le classi java, i files di help e gli eseguibili sono messi qui.

Netscape non richiede alcuna libreria di sistema addizionale, quindi non bisogna aggiungere nulla in /opt/ltsp/i386/lib. Ma molte altre applicazioni ne hanno bisogno.

Quindi, come puoi fare a sapere quali librerie sono necessarie? Ecco l'utilità del comando **ldd**.

Poniamo che tu voglia far girare una certa applicazione in locale; prenderemo il programma **gaim** come esempio. **gaim** è un client per la rete AOL Instant Messenger, che permette di comunicare con altri utenti del sistema AOL.

La prima cosa che bisogna fare è trovare l'eseguibile di **gaim**. Su Redhat 7.2 si trova in /usr/bin.

Una volta trovato, possiamo far girare **ldd** su gaim:

```
[jam@server /]$ ldd /usr/bin/gaim
        libaudiofile.so.0    => /usr/lib/libaudiofile.so.0 (0x40033000)
        libm.so.6           => /lib/i686/libm.so.6 (0x40051000)
        libnsl.so.1         => /lib/libnsl.so.1 (0x40074000)
        libgnomeui.so.32    => /usr/lib/libgnomeui.so.32 (0x4008a000)
        libart_lgpl.so.2    => /usr/lib/libart_lgpl.so.2 (0x4015d000)
        libgdk_imlib.so.1   => /usr/lib/libgdk_imlib.so.1 (0x4016c000)
        libSM.so.6          => /usr/X11R6/lib/libSM.so.6 (0x40191000)
        libICE.so.6         => /usr/X11R6/lib/libICE.so.6 (0x4019a000)
        libgtk-1.2.so.0     => /usr/lib/libgtk-1.2.so.0 (0x401b1000)
        libdl.so.2          => /lib/libdl.so.2 (0x402df000)
        libgdk-1.2.so.0     => /usr/lib/libgdk-1.2.so.0 (0x402e3000)
        libgmodule-1.2.so.0 => /usr/lib/libgmodule-1.2.so.0 (0x40319000)
        libXi.so.6          => /usr/X11R6/lib/libXi.so.6 (0x4031d000)
        libXext.so.6        => /usr/X11R6/lib/libXext.so.6 (0x40325000)
        libX11.so.6         => /usr/X11R6/lib/libX11.so.6 (0x40333000)
        libgnome.so.32      => /usr/lib/libgnome.so.32 (0x40411000)
        libgnomesupport.so.0 => /usr/lib/libgnomesupport.so.0 (0x40429000)
        libesd.so.0         => /usr/lib/libesd.so.0 (0x4042e000)
        libdb.so.2          => /usr/lib/libdb.so.2 (0x40436000)
        libglib-1.2.so.0    => /usr/lib/libglib-1.2.so.0 (0x40444000)
        libcrypt.so.1       => /lib/libcrypt.so.1 (0x40468000)
        libc.so.6           => /lib/i686/libc.so.6 (0x40495000)
        libz.so.1           => /usr/lib/libz.so.1 (0x405d1000)
        /lib/ld-linux.so.2  => /lib/ld-linux.so.2 (0x40000000)
```

Il listato elenca tutte le librerie richieste dal programma **gaim**.

Molti programmi che necessitano di librerie condivise, si basano sul caricatore dinamico **ld-linux** per trovare tutte le librerie richieste. Altri programmi, invece, caricano le librerie manualmente usando la funzione **dlopen()**; per queste applicazioni, **ldd** non riuscirà a mostrarci le librerie richieste: **strace** può allora essere usato come alternativa, tracciando l'esecuzione del programma, e tu potrai vedere le chiamate **dlopen()**, che contengono i nomi delle librerie.

Una volta prese le librerie, devono essere copiate nel posto giusto della directory `/opt/ltsp/i386`.

9.5. Lanciare le applicazioni locali

In X Windows, i programmi normalmente girano relativamente a dove gira il window manager, cioè se il window manager gira sul server, mostrando la propria grafica sullo schermo della workstation, allora qualunque programma lanciato girerà sul server, mandando il proprio output sulla workstation.

Il trucco sta nel far in modo che il server dica alla workstation di lanciare il programma; questo viene fatto tramite il comando **rsh**.

Ecco un esempio di come far partire **gaim** sulla workstation:

```
HOST=`echo $DISPLAY | awk -F: '{ print $1 }'`
rsh ${HOST} /usr/bin/gaim -display ${DISPLAY}
```

L'esempio sopra riportato può venire inserito all'interno di una finestra **xterm**, oppure può essere inserito in uno script ed essere lanciato tramite un'icona sul desktop.

Lanciare una sessione di Netscape in locale e simile, ma dovremo anche settare delle variabili d'ambiente particolari di cui il programma ha bisogno.

```
HOST=`echo $DISPLAY | awk -F: '{ print $1 }'`
rsh ${HOST} MOZILLA_HOME=/usr/local/netscape \
    /usr/local/netscape/netscape -display ${DISPLAY}
```

Capitolo 10. Esempi di configurazioni

Praticamente, qualunque aspetto della workstation puo' essere configurato tramite le voci del file `lts.conf`, che si trova nella directory `/opt/ltsp/i386/etc`.

10.1. Mouse seriale

Ecco un'esempio delle voci in `lts.conf` che servono a configurare un mouse seriale standard a 2 pulsanti:

```
X_MOUSE_PROTOCOL = "Microsoft"
X_MOUSE_DEVICE   = "/dev/ttyS0"
X_MOUSE_RESOLUTION = 400
X_MOUSE_BUTTONS  = 2
X_MOUSE_EMULATE3BTN = Y
```

10.2. Mouse PS/2 Wheel (con rotellina)

Questo e' un esempio di voci per un Intellimouse:

```
X_MOUSE_PROTOCOL = "IMPS/2"
X_MOUSE_DEVICE   = "/dev/psaux"
X_MOUSE_RESOLUTION = 400
X_MOUSE_BUTTONS  = 5
X_ZAxisMapping   = "4 5"
```

10.3. Stampante USB collegata ad un ThinkNic

Il ThinkNIC ha una porta USB che puo' essere usata per collegare una stampante. Eccone la configurazione:

```
MODULE_01 = usb-ohci
MODULE_02 = printer
PRINTER_0_DEVICE = /dev/usb/lp0
PRINTER_0_TYPE = S
```

10.4. Forzare il caricamento di XFree86 3.3.6 sulla workstation

Per default viene sempre caricata la versione 4.1.0 di XFree. Se vuoi invece forzare la workstation a caricare la vecchia release 3.3.6, devi prima installare il pacchetto Xserver 3.3.6 corretto. Poi, dovrai inserire delle voci nel file `lts.conf`, come in questo esempio:

```
XSERVER = XF86_SVGA
```

Capitolo 11. Altre fonti di informazioni

11.1. Riferimenti online

1. Il sito di LTSP

www.LTSP.org

2. Documento Diskless–Nodes HOW–TO per Linux

www.linuxdoc.org/HOWTO/Diskless–HOWTO.html

3. Il sito di Etherboot

etherboot.sourceforge.net

4. Il sito Rom–O–Matic

www.Rom–O–Matic.net

5. XFree86–Video–Timings–HOWTO

www.linuxdoc.org/HOWTO/XFree86–Video–Timings–HOWTO.html

6. La versione Linux di NIS(YP)/NYS/NIS+ HOWTO

www.linuxdoc.org/HOWTO/NIS–HOWTO.html

11.2. Pubblicazioni su carta

- 1.

Managing NFS and NIS
Hal Stern
O'Reilly & Associates, Inc.
1991
ISBN 0–937175–75–7

- 2.

TCP/IP Illustrated, Volume 1
W. Richard Stevens
Addison–Wesley
1994
ISBN 0–201–63346–9

- 3.

X Window System Administrator's Guide
Linda Mui and Eric Pearce

O'Reilly & Associates, Inc.

1993

ISBN 0-937175-83-8

(Volume 8 of the The Definitive Guides to the X Window System)