

LTSP – Linux Terminal Server Project – Progetto per un sistema Server per Terminali con Linux v4.1 – Documentazione in Italiano

James McQuillan

<jam@LTSP.org>

Andrea Fascilla

<afascilla@mail.com>

Copyright © 2004 James A. McQuillan

Diario delle Revisioni

| | | |
|-----------------------|------------|---------------------|
| Revisione 4.1.3-en | 2004-06-20 | Revisionato da: jam |
| Revisione 4.1.3-0-it | 2005-03-09 | Revisionato da: af |
| Revisione 4.1.3-0c-it | 2005-03-21 | Revisionato da: eag |
| Revisione 4.1.3-0d-it | 2005-03-22 | Revisionato da: af |
| Revisione 4.1.3-0e-it | 2005-03-23 | Revisionato da: cp |
| Revisione 4.1.3-0f-it | 2005-04-18 | Revisionato da: af |

Il sistema GNU/Linux è una piattaforma ideale per sviluppare una rete leggera di terminali utilizzando computer privi di disco fisso (thin clients). Lo scopo principale di questo documento è di mostrare come installare ed utilizzare questi clients usando LTSP. Inoltre questo documento descrive molte delle problematiche che si incontrano con client senza disco fisso in generale.

Sommario

| | |
|---|-----------|
| <u>Introduzione</u> | 1 |
| <u>1. Avvertenze (Disclaimer)</u> | 2 |
| <u>2. Copyright e Licenza d'uso</u> | 2 |
| <u>3. Marchi registrati</u> | 2 |
| <u>Capitolo 1. Schema teorico di funzionameno</u> | 3 |
| <u>1.1. I vari passi per l'avvio della workstation</u> | 3 |
| <u>1.2. Il caricamento del kernel nella memoria</u> | 6 |
| <u>Capitolo 2. Installare LTSP sul server</u> | 9 |
| <u>2.1. Installazione delle utility LTSP</u> | 9 |
| <u>2.2. Installazione dei pacchetti client di LTSP</u> | 10 |
| <u>2.3. Configurazione dei servizi necessari a LTSP</u> | 14 |
| <u>2.4. Configurazione specifica delle workstation</u> | 17 |
| <u>2.5. Visualizzazione dela configurazione corrente</u> | 19 |
| <u>Capitolo 3. La Configurazione della workstation</u> | 21 |
| <u>3.1. Avviare il computer con PXE</u> | 21 |
| <u>3.2. Avviare il computer con Etherboot</u> | 22 |
| <u>Capitolo 4. La workstation in fuzione</u> | 24 |
| <u>Capitolo 5. La Stampa</u> | 25 |
| <u>5.1. La parte della configurazione della workstation</u> | 25 |
| <u>5.2. La parte della configurazione del server</u> | 25 |
| <u>Capitolo 6. Gli script per le sessioni dello schermo</u> | 28 |
| <u>Capitolo 7. Risoluzione dei problemi (Troubleshooting)</u> | 30 |
| <u>7.1. Risoluzione dei problemi nel caso dell'immagine Etherboot su dischetto floppy</u> | 30 |
| <u>7.2. Risoluzione dei problemi con DHCP</u> | 30 |
| <u>7.3. Risoluzione dei problemi con TFTP</u> | 33 |
| <u>7.4. Risoluzione dei problemi con il filesystem principale NFS</u> | 34 |
| <u>7.5. Risoluzione dei problemi con X</u> | 36 |
| <u>7.6. Risoluzione dei problemi con il Display manager</u> | 37 |
| <u>Capitolo 8. I Kernel</u> | 40 |
| <u>8.1. Kernel standard forniti con LTSP</u> | 40 |
| <u>8.2. Costuitevi il vostro kernel</u> | 40 |
| <u>Capitolo 9. Le voci del file di configurazione lts.conf</u> | 45 |
| <u>9.1. Esempio del file lts.conf</u> | 45 |
| <u>9.2. Elenco dei parametri disponibili per il file lts.conf</u> | 45 |
| <u>Capitolo 10. Applicazioni locali</u> | 53 |
| <u>10.1. Vantaggi di far eseguire le applicazioni sulla workstation</u> | 53 |
| <u>10.2. Svantaggi di far eseguire le applicazioni sulla workstation</u> | 53 |
| <u>10.3. Configurazione del server per l'esecuzione locale delle applicazioni</u> | 54 |

Sommario

Capitolo 10. Applicazioni locali

10.4. Configurazione delle Applicazioni.....55

10.5. Lanciare le applicazioni locali.....56

Capitolo 11. Esempi di configurazioni.....57

11.1. Mouse seriale.....57

11.2. Mouse PS/2 con rotella.....57

11.3. Stampante USB su una ThinkNic.....57

11.4. Come forzare la workstation a caricare XFree86 3.3.6 Xserver.....57

Capitolo 12. Altre fonti di informazioni.....58

12.1. Informazioni Online.....58

12.2. Pubblicazioni stampate (libri, riviste ecc) (in inglese).....59

Introduzione

Il progetto LTSP fornisce un modo semplice per (ri)utilizzare dei computer a basso costo come terminali, sia in modo grafico che in modo testo, collegandoli ad un server GNU/Linux.

In una tipica situazione d'ufficio su ogni scrivania vengono normalmente utilizzati computer abbastanza potenti, basati su processori Intel o compatibili, ognuno con molti gigabytes di spazio su disco fisso. Ciascun utente memorizza i propri dati sul disco rigido presente nel computer e molto raramente vengono effettuati dei backup.

Ha veramente un senso avere un computer completo su ciascuna scrivania?

Secondo noi no. Possiamo migliorare questa situazione.

. Fortunatamente c'è un'alternativa. Utilizzando LTSP, si possono usare anche PC piuttosto datati o comunque di scarse prestazioni. Basta, eventualmente rimuovere, il disco rigido, il floppy ed il CDROM, ed aggiungere una scheda di rete con supporto per il boot da rete. La maggior parte delle schede di rete ha un alloggiamento dove inserire una ROM che permette il boot da rete.

Durante la fase di accensione (la fase di boot), il computer senza dischi ottiene le informazioni necessarie per far funzionare la rete (come il proprio numero IP) ed il kernel Linux (la parte fondamentale di questo sistema operativo) dal server, quindi accede ai dischi (monta il root filesystem) tramite il servizio NFS.

Il computer della postazione di lavoro (workstation) può essere configurato nei 3 modi sotto descritti:

- **Interfaccia Grafica con Sistema X Window**

Si utilizza X Windows. La workstation può essere usata per accedere a qualsiasi applicazione presente sul server o su altri server eventualmente presenti all'interno della stessa rete.

- **Sessioni tramite telnet ad interfaccia in modalità testo**

La workstation può instaurare una o più sessioni telnet con il server. Ogni sessione telnet viene collocata in uno schermo virtuale separato. Sono possibili più sessioni in contemporanea fino ad un massimo di 9 sessioni. Premendo i tasti da Alt-F1 a Alt-F9 sarà possibile passare da una sessione ad un'altra.

- **Shell prompt (linea di comando)**

La workstation può essere configurata in modo da fornire direttamente e semplicemente una linea di comando (la bash shell) sulla console. Questo può essere molto utile in alcuni casi, ad esempio quando è necessario, in fase di configurazione correggere eventuali problemi che possono insorgere con il sistema X Window o con il collegamento NFS.

La cosa veramente interessante è che si possono avere una gran quantità di workstation (terminali) serviti da un solo server con sistema GNU/Linux. Quante potranno essere le workstation (i terminali)? Quante possano essere dipende dalla dimensione e dalle capacità del server e da quali e quante applicazioni verranno usate nello stesso momento.

Non è raro avere 50 workstations, tutte con Mozilla e OpenOffice aperti, utilizzando un server con due processori P4–2.4 con 4GB di ram. Sappiamo che questa configurazione funziona egregiamente ed il carico medio raramente va al di sopra 1.0!

1. Avvertenze (Disclaimer)

Né l'autore né il distributore né il traduttore di questo documento e neppure qualsiasi persona che ad esso abbia contribuito, potrà essere ritenuta in qualche modo responsabile per qualsiasi danno fisico, finanziario, morale o di qualsiasi altro tipo che possa verificarsi seguendo i suggerimenti presenti in questo testo.

2. Copyright e Licenza d'uso

Questo documento è coperto da copyright dall'anno 2004 da parte di James McQuillan, e viene fornito (rilasciato) sotto i termini della Licenza denominata GNU Free Documentation License, che è qui sotto inclusa per riferimento.

3. Marchi registrati

Tutti i marchi citati sono proprietà dei rispettivi proprietari.

Capitolo 1. Schema teorico di funzionameno

Far partire (fargli fare il boot) un computer da usare come terminale (workstation) che sia senza dischi richiede un certo numero di passi. La comprensione di cosa accade nei vari passi del procedimento, renderà più facile risolvere eventuali problemi, nel caso dovessero insorgere.

Ci sono quattro servizi di base che sono necessari per far partire una workstation (un terminale) con LTSP. Questi sono:

- DHCP
- TFTP
- NFS
- XDMCP

Il sistema LTSP è molto flessibile. Ognuno dei servizi qui sopra elencati può essere fornito dallo stesso server o da più server. Negli esempi descriveremo il caso di una semplice configurazione in cui c'è un solo server, che fornisce tutti i servizi qui sopra elencati.

1.1. I vari passi per l'avvio della workstation

1. Il kernel Linux viene caricato nella memoria della workstation. Questo può essere fatto in diversi modi, tra questi:
 - a. Bootrom (Etherboot, PXE, MBA, Netboot)
 - b. Dischetto floppy
 - c. Disco fisso (Harddisk)
 - d. CD-ROM
 - e. Dispositivo di Memoria USB (USB Memory device)Ognuno dei suddetti metodi per l'avvio sarà spiegato di seguito in questo capitolo.
2. Una volta che il kernel è stato caricato nella memoria, ne avverrà la sua esecuzione.
3. Il kernel inizializzerà l'intero sistema e tutte le periferiche che verranno identificate.
4. Qui entra in gioco una caratteristica interessante. Mentre viene caricato il kernel in memoria, viene anche creato un disco virtuale in memoria (RAM disk). Una linea di comando per il kernel del tipo **root=/dev/ram0** informerà il kernel di utilizzare il dico ram come disco principale (il ramdisk viene montato come cartella principale – root directory).
5. Normalmente, quando il kernel completa la fase di avvio, lancia il programma **init**. Invece in questo caso abbiamo istruito il kernel di caricare e di eseguire un piccolo script. Questo si ottiene passando alla linea di comando del kernel il parametro **init=/linuxrc**.
6. Lo script **/linuxrc** inizia effettuando una scansione del bus PCI alla ricerca di una scheda di rete. Per ogni scheda PCI trovata, effettua una ricerca nel file `/etc/niclist` per vedere se è qui elencata assieme al nome del modulo del kernel necessario. Una volta trovata una corrispondenza, viene fornito al kernel il nome del modulo che serve come driver per la scheda di rete trovata, e ne avverrà il suo caricamento. Nel caso di schede ISA, il nome del modulo driver DEVE essere specificato come parametro nella riga di comando del kernel, insieme ai valori di IRQ ed indirizzo e qualunque altro

parametro aggiuntivo eventualmente necessario per configurare la scheda.

7. Un piccolo client DHCP chiamato **dhclient** entrerà successivamente in esecuzione per fare un'altra richiesta al server DHCP. C'è bisogno di questa ulteriore richiesta fatta in user-space perchè abbiamo bisogno di maggiori informazioni di quelle che sono state ottenute con la prima richiesta DHCP fatta dal chip ROM.
8. Quando **dhclient** avrà ricevuto una risposta dal server, sarà eseguito il file **/etc/dhclient-script**, che leggerà le informazioni ricevute e configurerà la scheda ethernet (interfaccia di rete eth0). Quando **dhclient** avrà ricevuto una risposta dal server, eseguirà il file **/etc/dhclient-script**, che utilizzerà le informazioni precedentemente ottenute e configurerà l'interfaccia di rete eth0.
9. Non è possibile fare il mount di questa directory direttamente come /. Bisogna prima fare il mount come /mnt e poi fare un **pivot_root**. Il **pivot_root** farà cambiare il root filesystem corrente con un nuovo filesystem. Quando questo comando sarà completato, il filesystem importato tramite NFS sarà montato come /, e il vecchio filesystem sarà montato come /oldroot. Fino a questo punto, il filesystem principale (il root filesystem) è stato nel disco RAM. Ora lo script /linuxrc farà il mount di un nuovo filesystem principale tramite NFS. La directory che viene esportata dal server è in genere /opt/lts/i386. Lo script non può fare il mount di questa directory direttamente come /. Deve prima fare il mount come /mnt. Poi eseguirà il comando **pivot_root** che scambierà il root filesystem corrente con quello nuovo. Una volta completata l'operazione, il filesystem NFS risulterà montato come /, ed il vecchio filesystem risulterà montato come /oldroot.
10. Una volta completata la fase di mount e di pivot del nuovo filesystem principale, lo script /linuxrc ha completato il suo lavoro ed è ora necessario chiamare il vero programma **/sbin/init**.
11. Il programma **init** leggerà il file **/etc/inittab** e comincerà a configurare l'ambiente della workstation.
12. Uno dei primi elementi presenti in **inittab** è il comando **rc.sysinit**, che sarà eseguito mentre la workstation è nello stato di **'sysinit'**.
13. Lo script **rc.sysinit** creerà un disco ram di 1MB che conterrà tutti i file che per qualunque motivo necessitano essere scritti o modificati.
14. Questo disco RAM sarà montato come directory **/tmp**. Ogni file che sarà necessario scrivere, esisterà nella directory **/tmp** e vi saranno link simbolici che punteranno a questi file.
15. Verrà montato il filesystem **/proc**.
16. Il file **lts.conf** verrà letto ed analizzato e tutti i parametri che appartengono a questa workstation verranno impostati come variabili d'ambiente per poter essere utilizzati dallo script **rc.sysinit**.
17. Se la workstation è configurata in modo da utilizzare NFS per lo swap (dei file temporanei), allora la directory **/var/opt/lts/swapfiles** sarà montata come **/tmp/swapfiles**. A questo punto, se non c'è già un file di swap per questa workstation, ne viene creato uno automaticamente. La dimensione del file di swap è configurabile tramite il file **lts.conf**.

Il file di swap verrà attivato utilizzando il comando **swapon**.

18. L'interfaccia di rete di **loopback** viene configurata. Questa è l'interfaccia di rete che ha l'indirizzo IP *127.0.0.1*.
19. Se "Local apps" è abilitato, viene fatto il mount della directory **/home**, in modo che le applicazioni possano accedere alle home directory degli utenti.
20. Alcune directory vengono create nel filesystem sotto `/tmp` per contenere alcuni file temporanei che sono necessari mentre il sistema è in funzione. Tra queste verranno create le seguenti directory:
 - a. `/tmp/compiled`
 - b. `/tmp/var`
 - c. `/tmp/var/run`
 - d. `/tmp/var/log`
 - e. `/tmp/var/lock`
 - f. `/tmp/var/lock/subsys`
21. Il file `/tmp/syslog.conf` verrà creato. Questo file conterrà le informazioni che configureranno il demone **syslogd**, tra cui a quale host (computer) della rete devono essere inviate le informazioni che devono essere mantenute nei log. Quale debba essere l'host che riceve il log di **syslogd** è specificato nel file `lts.conf`. Verrà creato un link simbolico chiamato `/etc/syslog.conf` che punterà al file `/tmp/syslog.conf`.
22. Il demone **syslogd** viene fatto partire utilizzando il file di configurazione creato al passo precedente.
23. Una volta che lo script `rc.sysinit` è terminato, il controllo ritorna al programma `/sbin/init`, che cambierà il runlevel da **sysinit** a **5**.

Questo causerà l'esecuzione di alcune istruzioni elencate nel file `/etc/inittab`.

24. Come impostazione predefinita (cioè se non vengono fornite istruzioni diverse), le istruzioni presenti in `inittab` faranno eseguire lo script `/etc/screen_session` sulla `tty1`, sulla `tty2` e sulla `tty3`. Questo fa sì che ci saranno 3 sessioni contemporaneamente. Quale sia il tipo di ognuna delle sessioni viene controllato dai parametri **SCREEN_01**, **SCREEN_02** e **SCREEN_03** nel file `lts.conf`.

Si possono inserire ulteriori sessioni nel file `inittab`, se lo si desidera.

25. Se **SCREEN_01** ha il valore **startx**, allora sarà mandato in esecuzione lo script `/etc/screen.d/startx`, che lancerà il sistema X Window, creando l'interfaccia grafica.

Nel file `lts.conf` c'è un parametro chiamato **XSERVER**. Se questo parametro manca o ha il valore "**auto**", allora verrà fatto un rivelamento automatico della scheda video. Se la scheda video di tipo PCI o AGP, allora i codici PCI Vendor e Device Id verranno letti e verrà fatta una ricerca nel file `/etc/vidlist`.

Se la scheda video è supportata da Xorg 6.7 la routine `pci_scan` ritornerà il nome del modulo del driver. Se è supportata soltanto da XFree86 3.3.6, `pci_scan` ritornerà il nome del server X che deve

essere utilizzato. Lo script `startx` riconosce la differenza perché i nomi dei vecchi server di XFree 3.3.6 iniziano per 'XF86_', mentre i nomi dei nuovi moduli di X di Xorg sono tipicamente dei nomi scritti in minuscolo, come, ad esempio, *ati* o *trident*.

26. Se viene usato Xorg, allora viene eseguito lo script `/etc/build_x4_cfg` per creare il file di configurazione XF86Config. Se viene usato XFree86 3.3.6, allora viene eseguito `/etc/build_x3_cfg` per creare il file di configurazione XF86Config. Questi file vengono collocati nella directory `/tmp`, che come precedentemente detto è un disco RAM, visto solo dalla workstation.

Il file di configurazione XF86Config sarà creato in base ai parametri presenti nel file `/etc/lts.conf`.

27. Quando il file di configurazione XF86Config è stato completato, allora lo script `startx` lancerà il server X con la nuova configurazione.
28. Il server X lancerà una richiesta *XDMCP* al server LTSP, che fornirà una finestra di login.
29. A questo punto l'utente può effettuare il login ed otterrà una sessione sul server.

Questo causerà un po' di confusione agli inizi. Mentre si è seduti davanti alla workstation si avrà una sessione che è eseguita sul server. Tutti i comandi verranno eseguiti sul server, ma l'output sarà mostrato sullo schermo della workstation.

1.2. Il caricamento del kernel nella memoria

Il caricamento del kernel Linux dentro la memoria della workstation può essere ottenuto in più modi.

- Boot da memoria ROM
- Boot da dispositivo locale

1.2.1. Boot da memoria ROM

- Etherboot

Etherboot è un progetto open-source molto popolare per il bootrom (boot da rom). Contiene i driver per molte delle schede di rete più comuni e funziona molto bene con LTSP.

Il kernel linux deve contenere il tag **mknbi-linux**, che fa preparare il kernel per poter fare il boot da rete facendo eseguire prima del kernel del codice aggiuntivo e, una volta caricato ed eseguito il kernel, facendo utilizzare `initrd`. Il kernel linux deve contenere il tag **mknbi-linux**, che fa preparare il kernel per poter fare il boot da rete, premettendo al kernel una parte di codice aggiuntivo e collegando (lo script o il demone) `initrd` alla fine del kernel.

I kernel che vengono forniti con LTSP hanno già questo tag e sono pronti per fare il boot con Etherboot.

Etherboot può essere anche scritto su di un floppy. Questa cosa è molto comoda per fare dei test.

- PXE

La parte delle specifiche del 'Wired for Management' della fine del 1990 includono una specifica per la tecnologia del bootrom conosciuta come *Pre-boot Execution Environment* e comunemente abbreviata in **PXE**.

Un bootrom PXE può caricare al massimo un file di 32 kB. Un kernel Linux è molto più grande di questa dimensione. Pertanto dobbiamo configurare PXE per far sì che, come secondo passo, caricato un boot-loader chiamato *pxelinux*. PxeLinux è sufficientemente piccolo da poter essere caricato ed è a conoscenza di come caricare file più grandi, come il kernel Linux.

- **MBA**

Managed Boot Agent (MBA) è un bootrom della società *emBoot*. emBoot era la divisione Lanworks della 3Com. MBA in realtà è quattro bootrom in uno. Può gestire PXE, TCP/IP, RPL e Netware.

L'implementazione MBA di PXE funziona molto bene. Può essere utilizzato con pxelinux per far fare il boot di un kernel Linux.

Anche il metodo *TCP/IP* può essere utilizzato, ma il kernel deve prima essere preparato con un'utility chiamata **imggen**.

- **Netboot**

Netboot, come Etherboot, è un progetto software gratuito che fornisce immagini per il boot ROM gratuite. La differenza è che un wrapper del drive NDIS o al driver di pacchetti che viene spedito con la scheda di rete

1.2.2. Boot da dispositivo locale

- **Disco Floppy**

Ci sono due metodi per far effettuare il boot ad una workstation LTSP con un floppy. Il primo metodo consiste nel copiare l'Etherboot nel boot sector del floppy. Quindi tutto funziona come nel caso del bootrom. Il codice letto dal boot record del floppy viene eseguito, la scheda di rete viene inizializzata e il kernel viene caricato in memoria dal server via rete.

L'altro metodo consiste nello scrivere il kernel e l'initrd sul floppy e fare il boot dal floppy. Tuttavia è estremamente più veloce caricare il boot dalla rete.

- **Hard disk (disco fisso)**

Si utilizza LILO o GRUB per caricare dal disco fisso il kernel Linux e l'initrd. Oppure si può caricare l'immagine Etherboot bootrom dal disco fisso e agire come nel caso del bootrom.

- **CD-ROM**

Un disco CD-ROM con capacità di boot può essere caricato in memoria. Anche qui si può caricare l'intero kernel e l'initrd oppure l'immagine Etherboot.

- **Dispositivo di Memoria USB (USB Memory device)**

Esattamente come nel caso di un CD-ROM, di un disco floppy o di un disco fisso si può usare un dispositivo di memoria USB per far fare il boot. Anche qui si può caricare direttamente il kernel Linux e l'initrd oppure si può caricare il modulo Etherboot.

Capitolo 2. Installare LTSP sul server

La cosa migliore è pensare a LTSP come ad una distribuzione completa di Linux, che si aggiunge ad una distribuzione normalmente utilizzata per installare linux su un computer. La distribuzione host può essere quella che preferite. In realtà non c'è neppure alcuna reale motivo per cui il server debba aver il sistema operativo Linux. L'unico requisito necessario è che il sistema operativo del server sia in grado di fornire i servizi NFS (Network File System). La maggior parte dei sistemi operativi Unix è in grado di fornire questo servizio. In effetti anche alcune versioni di Microsoft Windows possono essere configurate per funzionare come server LTSP.

L'implementazione di un server LTSP avviene in tre passi.

- Installazione delle utility LTSP (dei programmi di utilità di LTSP)
- Installazione dei pacchetti client di LTSP
- Configurazione dei servizi necessari a LTSP

2.1. Installazione delle utility LTSP

Dalla versione 4.1 in poi, LTSP è dotata di un insieme di utility per l'installazione e la gestione dei pacchetti client di LTSP. (I programmi che sono eseguiti dai clienti leggeri) e per configurare i servizi sul server LTSP.

Lo strumento per l'amministrazione si chiama **ltspadmin** e lo strumento per la configurazione si chiama **ltspcfg**. Entrambi questi strumenti fanno parte del pacchetto **ltsp-utils**.

Il pacchetto **ltsp-utils** è disponibile sia nel formato **RPM** sia in quello **TGZ**. Scegliete quello che preferite o che è migliore per voi e seguite le relative istruzioni .

2.1.1. Installazione del pacchetto RPM

Fate il download dell'ultima versione del pacchetto RPM delle **ltsp-utils** ed installatelo con il comando seguente:

```
rpm -ivh ltsp-utils-0.1-0.noarch.rpm
```

Questo comando installerà le utility sul server.

2.1.2. Installazione del pacchetto TGZ

Fate il download dell'ultima versione del pacchetto TGZ delle **ltsp-utils** ed installatelo con i comandi seguenti:

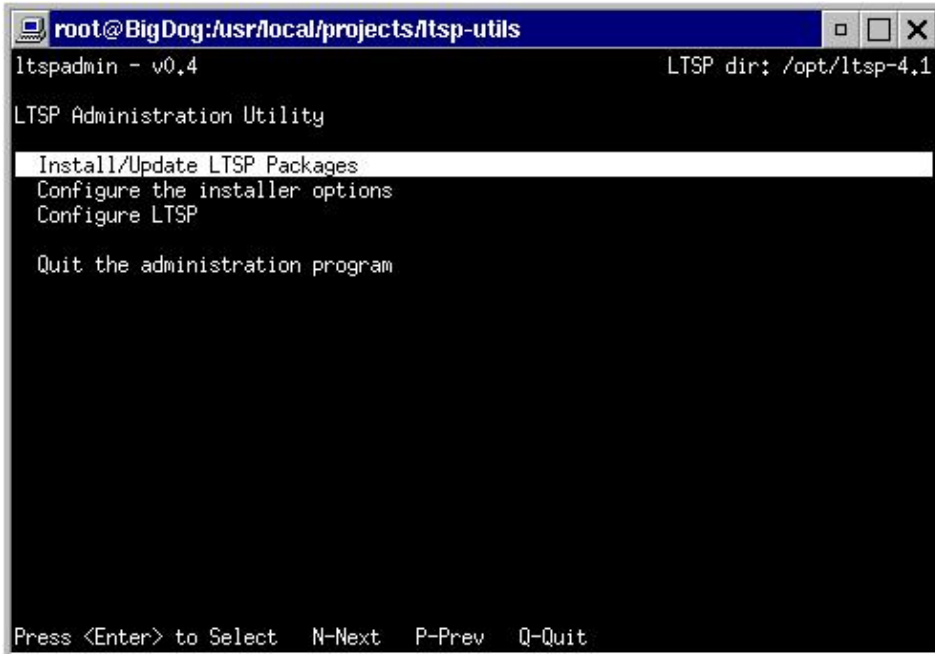
```
tar xzf ltsp-utils-0.1-0.noarch.tgz
cd ltsp_utils
./install.sh
cd ..
```

Questo comando installerà le utility sul server. Questo metodo è utile nel caso di sistemi che non supportano gli RPM.

2.2. Installazione dei pacchetti client di LTSP

Una volta che è stata completata l'installazione del pacchetto `ltsp-utils`, si ha a disposizione il comando **ltspadmin**. Questa utility permette di gestire i pacchetti del client LTSP. L'utility svolge il compito di interrogare il deposito per i download di LTSP e di ottenere l'elenco dei pacchetti attualmente disponibili.

Lanciate il comando **ltspadmin** e vedrete una schermata come quella riportata qui:



```
root@BigDog:/usr/local/projects/ltsp-utils
ltspadmin - v0.4                               LTSP dir: /opt/ltsp-4.1

LTSP Administration Utility

Install/Update LTSP Packages
Configure the installer options
Configure LTSP

Quit the administration program

Press <Enter> to Select  N-Next  P-Prev  Q-Quit
```

Figura 2–1. Installazione di LTSP – Schermata principale

Da questa schermata potete scegliere "Install/Update" (Installa/Aggiorna). Se è la prima volta che si utilizza questa utility verrà mostrata la schermata di configurazione del programma d'installazione, come mostrato nella seguente figura.

```

root@BigDog:/usr/local/projects/ltsp-utils
LTSP Installer configuration

Where to retrieve packages from?
[http://www.mcquill.com/ltsp-4.1/] http://www.ltsp.org/ltsp-4.1

In which directory would you like to place the LTSP client tree?
[/opt/ltsp-4.1]

If you want to use an HTTP proxy, enter it here
Use 'none' if you don't want a proxy
Example: http://proxy.yourdomain.com;3128

[none]

If you want to use an FTP proxy, enter it here
(Use 'none' if you don't want a proxy)

[none]

Correct? (y/n/c) █

```

Figura 2–2. Installazione di LTSP – Schermata di configurazione

nella schermata di *configurazione* si possono impostare alcuni parametri che saranno utilizzati dal programma d'installazione per fare il download dei pacchetti LTSP e per installarli. Questi parametri sono:

Where to retrieve packages from

(Da dove prelevare i pacchetti) Specifica l'URL del deposito dei pacchetti. Tipicamente sarà `http://www.ltsp.org/ltsp-4.1`. Tuttavia nel caso in cui si voglia installare i pacchetti da un supporto locale (come un disco fisso o un CD), si può utilizzare una URL che inizi per `file:`. Ad esempio se i pacchetti sono su un CD-ROM e questo CD-ROM è stato montato sotto la directory `/mnt/cdrom`, allora il valore per questa opzione sarà: `file:///mnt/cdrom`. (Fate attenzione devono esserci tre barre dopo i due punti).

In which directory would you like to place the LTSP client tree

(In quale directory volete mettere i file e le sottodirectory del client LTSP). Questa è la directory sul server dove volete che saranno messi i file e le sotto directory del client LTSP. Tipicamente sarà: `/opt/ltsp`. La directory specificata, se non esiste di già, verrà creata automaticamente.

A partire da questa directory verranno create una directory principale per ciascuna delle architetture supportate. Attualmente soltanto l'architettura x86 è ufficialmente supportata da LTSP, ma ci sono diverse persone che stanno lavorando a dei porting verso altre architetture (cioè stanno lavorando a rendere i programmi funzionanti anche su altre architetture), come ad esempio PPC e Sparc.

HTTP Proxy

(Proxy per le connessioni HTTP) se il server è protetto da un firewall e se le connessioni al web devono passare dal proxy, questo parametro permette di configurare il programma d'installazione per poter utilizzare il proxy. Il valore da inserire deve contenere la URL del proxy, compreso di protocollo e numero di porta, come ad esempio: `http://firewall.yourdomain.com:3128`.

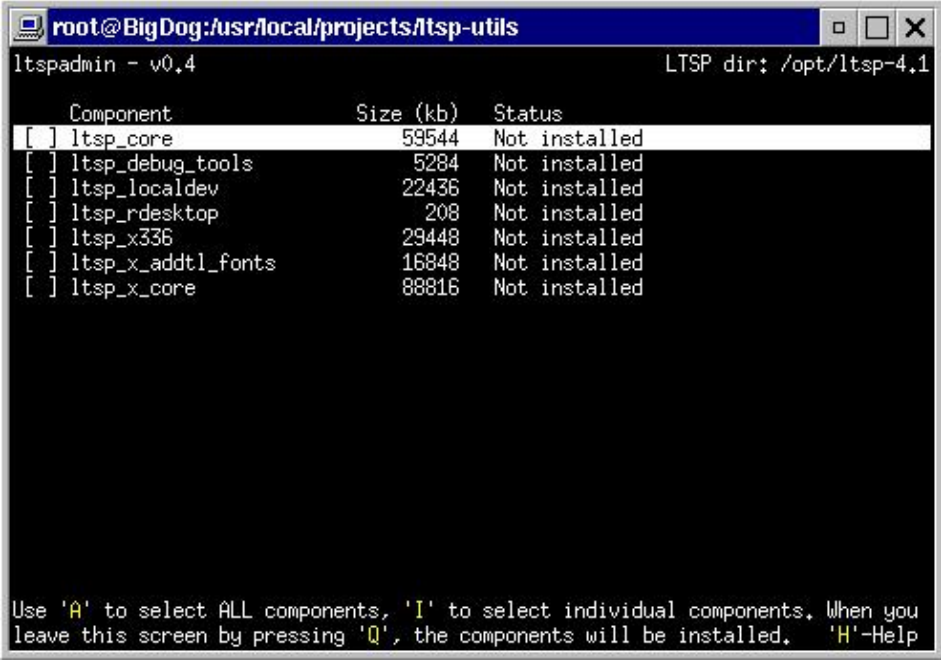
Se non c'è alcun proxy o non è necessario utilizzarlo, questo parametro deve essere impostato a "none".

FTP Proxy

(Proxy per le connessioni FTP) Se i pacchetti sono collocati su un server FTP ed è necessario passare attraverso un proxy FTP, questo parametro permette di specificare le necessarie informazioni. La sintassi è simile al caso del Proxy per la connessione HTTP.

Se non c'è alcun proxy o non è necessario utilizzarlo, questo parametro deve essere impostato a "none".

Una volta inserito le necessarie informazioni nella schermata di configurazione, il programma d'installazione farà una richiesta a chi contiene i pacchetti e otterrà la lista dei componenti al momento disponibili.



```

root@BigDog:/usr/local/projects/ltsp-utils
ltspadmin - v0.4                               LTSP dir: /opt/ltsp-4.1

  Component                Size (kb)  Status
[ ] ltsp_core               59544     Not installed
[ ] ltsp_debug_tools        5284      Not installed
[ ] ltsp_localdev           22436     Not installed
[ ] ltsp_rdesktop           208       Not installed
[ ] ltsp_x336                29448     Not installed
[ ] ltsp_x_addtl_fonts      16848     Not installed
[ ] ltsp_x_core              88816     Not installed

Use 'A' to select ALL components, 'I' to select individual components. When you
leave this screen by pressing 'Q', the components will be installed. 'H'-Help

```

Figura 2–3. Installazione di LTSP – Lista dei componenti

Potete selezionare i componenti uno ad uno. Per selezionare un componente, muovete la barra evidenziata sulla riga del componente voluto e premete il tasto 'I'. Per selezionare tutti i componenti basta premere il tasto 'A'. Nella maggior parte dei casi probabilmente è questa la cosa da fare. In questo modo potete supportare la più vasta gamma di hardware come client.

Ci sono alcuni tasti che possono essere utilizzati per spostarsi all'interno dello schermo. Per ottenere una schermata d'aiuto potete premere il tasto 'H'.

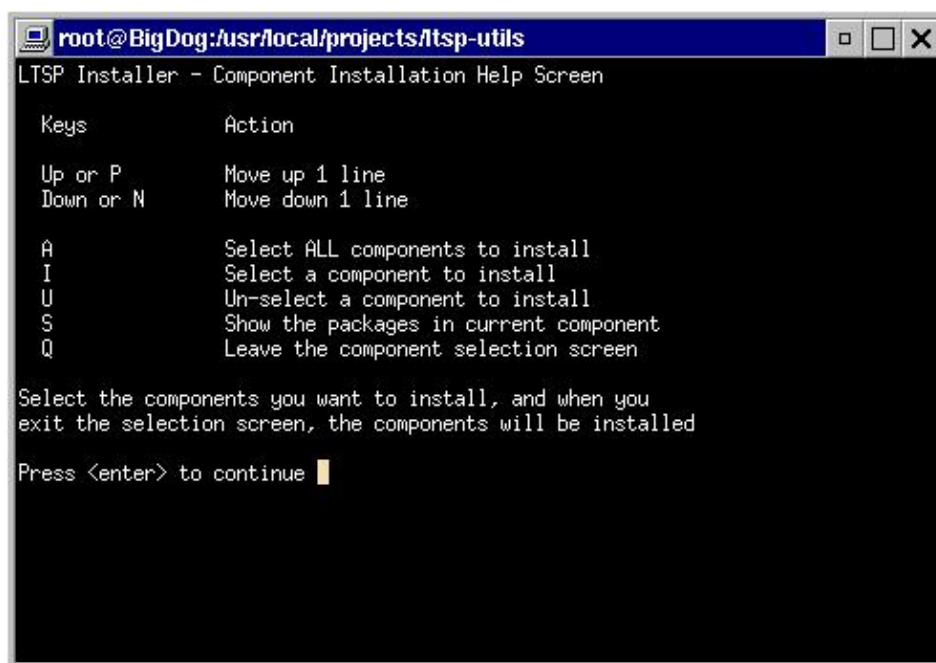


Figura 2–4. LTSP installer – Finestra d'Aiuto

Se volete vedere la lista dei pacchetti che sono presenti in un certo componente, premete il tasto 'S', e verrà mostrata la lista dei pacchetti. Verrà indicata la versione attualmente installata e l'ultima versione disponibile.



Figura 2–5. LTSP installer – Package list

Una volta terminata la selezione, uscite dalla schermata di selezione dei componenti, premendo il tasto 'S'. Il programma vi chiederà se volete realmente installare o aggiornare i pacchetti selezionati. Rispondete premendo il tasto 'Y' per fare iniziare il download e l'installazione dei pacchetti selezionati.

2.3. Configurazione dei servizi necessari a LTSP

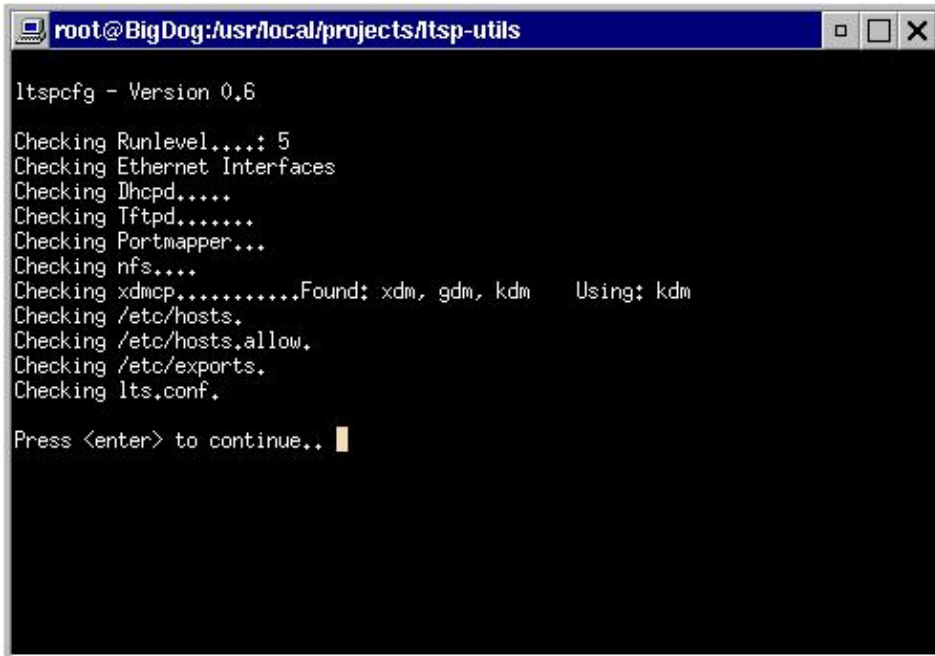
I servizi principali necessari per permettere l'avvio (il boot) delle workstation LTSP sono i quattro qui sotto elencati:

- DHCP
- TFTP
- NFS
- XDMCP

Il programma **ltspcfg** permette di configurare tutti e quattro questi servizi. Inoltre permette di configurare molti altri parametri necessari a LTSP.

Si può accedere a **ltspcfg** da **ltspadmin**, oppure può essere lanciato da riga di comando digitando **ltspcfg**.

Una volta lanciato il programma `ltspcfg`, verrà fatta una scansione (un'analisi) del server per accertarsi di cosa sia installato sul server e quali programmi siano in esecuzione. Verrà visualizzata una schermata del tipo:



```
root@BigDog:/usr/local/projects/ltsp-utils
ltspcfg - Version 0.6
Checking Runlevel.....: 5
Checking Ethernet Interfaces
Checking Dhcpd.....
Checking Tftpd.....
Checking Portmapper...
Checking nfs....
Checking xdmcp.....Found: xdm, gdm, kdm    Using: kdm
Checking /etc/hosts.
Checking /etc/hosts.allow.
Checking /etc/exports.
Checking /etc/ltsp.conf.

Press <enter> to continue.. █
```

Figura 2–6. ltspcfg – Schermata iniziale

Questa schermata mostra tutto ciò che è stato cercato dal programma.

Per configurare i parametri di configurazione premete il tasto 'C'. Verrà mostrato il menù di configurazione. Dal menù di configurazione dovete controllare sotto ogni voce che tutti i parametri siano correttamente configurati per servire le workstation LTSP.

```

xterm
ltspcfg v0.6      The Linux Terminal Server Project (http://www.LTSP.org)

 1 - Runlevel
 2 - Interface selection
 3 - DHCP configuration
 4 - TFTP configuration
 5 - Portmapper configuration
 6 - NFS configuration
 7 - XDMCP configuration
 8 - Create /etc/hosts entries
 9 - Create /etc/hosts.allow entries
10 - Create /etc/exports entries
11 - Create lts.conf file

R - Return to previous menu

Make a selection: █

```

Figura 2-7. ltspcfg – Schermata dei servizi

1 – Runlevel

La variabile **Runlevel** viene usata dal programma **init**. Nei sistemi operativi Linux ed Unix il sistema può essere in uno specifico "Runlevel" tra più possibili Runlevel. I Runlevel 2 e 3 in genere sono utilizzati per avere il server in modalità testo. Il Runlevel 5 rende il sistema in modalità grafica.

Normalmente, per un server LTSP, si utilizza il Runlevel 5. La maggior parte dei servizi sono già configurati per rispondere alle le richieste NFS e XDMCP quando il sistema è in Runlevel 5. Se un programma non è già configurato per questo, verrà configurato in tale modo da questa utilità.

2 – Interface selection

(selezione dell'interfaccia) Nel caso di sistemi con più di un'interfaccia (scheda) di rete, bisogna specificare su quale interfaccia sono collegati i client Thin (i terminali).

Il programma di configurazione, una volta nota quale sia l'interfaccia da utilizzare, creerà gli altri file di configurazione in modo opportuno. Tra i quali i file `dhcpd.conf` e `/etc/exports`.

3 – DHCP configuration

(Configurazione del DHCP) Il sistema DHCP deve essere configurato per poter fornire le necessarie informazioni alle workstation. I parametri necessari sono **fixed-address**, **filename**, **subnet-mask**, **broadcast-address** e **root-path**.

Utilizzando questo menù il programma sarà in grado di creare il file di configurazione `dhcpd.conf` e il programma **dhcpd** verrà attivato all'avvio del server.

4 – TFTP configuration

(Configurazione del TFTP) Il servizio TFTP viene utilizzato dalle workstation (client) per fare il download del kernel linux. Per far sì che il kernel possa essere fornito in questo modo il servizio **fttpd** deve essere attivato sul server.

5 – *Portmapper configuration*

(Configurazione della mappa delle porte) Il **Portmapper** è utilizzato dai servizi RPC, quali ad esempio NFS.

6 – *NFS configuration*

(Configurazione di NFS) Il servizio NFS permette di far fare il mount di directory locali da parte di computer remoti. Nella configurazione LTSP è necessario che questo servizio sia attivo sul server per permettere alle workstation di fare il mount del proprio root filesystem dal server.

Questo menù permette di configurare l'avvio di NFS all'avvio del server. I dettagli sul file di configurazione `/etc/exports` e sulla sua creazione vengono dati più avanti in questa sezione.

7 – *XDMCP configuration*

(Configurazione del XDMCP) XDMCP è l'abbreviazione di "X Display Manager Control Protocol" (Protocollo di Controllo e Gestione di Display X). Il server X invia una richiesta al programma di gestione (manager) del server per ottenere una finestra grafica di login.

I display manager di uso più comune sono **XDM**, **GDM** e **KDM**. Questo menù mostra i display manager disponibili e indica quale è stato selezionato.

Per ragioni di sicurezza, il Display Manager è configurato di default (come impostazione predefinita) per rifiutare le connessioni remote delle workstation. Questa è, in genere, la causa del comune problema di vedere soltanto *una schermata nera con una cursore a forma di X*. Il programma `ltspcfg` può essere utilizzato, in genere, per configurare il display manager in modo da consentire la connessione da parte di workstation (terminali) remoti.

8 – *Create /etc/hosts entries*

(Creare delle linee di riferimento all'interno del file `/etc/hosts`) Molti servizi, tra cui NFS e il Display manager, hanno bisogno di avere una mappa di corrispondenza tra gli indirizzi IP della workstation e il nome del computer (hostname). Si può utilizzare e configurare, a questo scopo, il Berkeley Internet Naming Daemon (BIND), ma bisogna essere certi che il sistema *reverses* (per il riconoscimento a ritroso) sia configurato in modo corretto. Probabilmente utilizzare `bind` è il miglior modo per ottenere lo scopo, tuttavia la spiegazione della configurazione di `bind` è al di là degli scopi di questa documentazione e del programma `ltspcfg`.

Un metodo più semplice per configurare la mappa di corrispondenza tra gli indirizzi IP e i nomi dei computer (hostname) è quello di utilizzare il file `/etc/hosts`.

9 – *Create /etc/hosts.allow entries*

(Creare delle linee di riferimento nel file `/etc/hosts.allow`) Alcuni servizi usano un sistema di sicurezza chiamato *tcpwrappers*. Questo sistema viene configurato tramite il file `/etc/hosts.allow`. Questo menù permette di configurare questo file.

10 – *Create the /etc/exports file*

(Creare il file `/etc/exports`) Questo è il file utilizzato da NFS, per determinare quali sono le directory che è consentito che vengano montate da una macchina remota. Questo menù permette di creare e configurare questo file.

11 – *Create the lts.conf file*

(Creare il file `lts.conf`) La configurazione di ogni singola workstation viene stabilita da delle linee del file `lts.conf`. Se si utilizzano workstation recenti con il bus PCI, non dovrebbe essere necessario

inserire alcuna linea aggiuntiva al file `lts.conf` file. Tuttavia è necessario che questo file esista. Questo menù creerà un file `lts.conf` con delle configurazione preimpostate (default).

2.4. Configurazione specifica delle workstation

A questo punto è necessario fornire al server LTSP le informazioni specifiche di ogni singola workstation. Ci sono tre file che contengono le informazioni specifiche delle singole workstation.

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

2.4.1. `/etc/dhcpd.conf`

La workstation ha bisogno di un indirizzo IP e di altre informazioni. Riceverà le seguenti informazioni dal server DHCP:

- IP address
- hostname
- Server IP address
- Default gateway
- Pathname of kernel to load
- Server and directory path to be mounted as the root filesystem

Nella configurazione mostrata in esempio abbiamo scelto che l'indirizzo IP sia gestito dal server DHCP, che lo assegna alle workstation.

Durante l'esecuzione dello script `ltsp_initialize`, viene installato un file `dhcpd.conf` di esempio. Potete copiare il file `/etc/dhcpd.conf.example` come file `/etc/dhcpd.conf` ed usarlo come base per la vostra configurazione del dhcp. Bisogna modificare questo file per adattarlo alle specifico ambiente in cui sono presenti le workstation e il server.

```
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 2–8. /etc/dhcpd.conf

A partire dalla versione 2.09pre2 di LTSP, non c'è più bisogno di specificare quale particolare kernel caricare in memoria. Il pacchetto standard del kernel supporta tutte le schede di rete supportate da Linux. Nel pacchetto del kernel distribuito da LTSP ci sono due file per il kernel. Uno che ha applicata la Linux Progress Patch (LPP) e l'altro che non le ha. I nomi di questi file sono

```

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

```

Probabilmente avete notato che questi file del kernel risiedono nella directory `/tftpbboot/lts`, ma nella parte dove inserire il nome del file ("filename") nel file `/etc/dhcpd.conf` manca la parte iniziale del nome del path, `/tftpbboot`. Questo è dovuto al fatto che a partire dalla versione 7.1 di Red Hat TFTP viene eseguito con l'opzione `'-s'`. Questo fa sì che il demone `tftpd` viene eseguito in modalità *sicura*. Pertanto viene eseguito all'avvio un **chroot** alla directory `/tftpbboot`. Pertanto tutti i file accessibili al demone `tftpd` sono relativi alla directory `/tftpbboot`.

Altre distribuzioni di Linux potrebbero non avere l'opzione `'-s'` impostata per `tftpd`. In questo caso dovete aggiungere il prefisso `/tftpbboot` al nome del path del kernel.

2.4.2. /etc/hosts

Mappa tra gli indirizzi IP address e il nome del computer (hostname)

I computer utilizzano normalmente soltanto l'indirizzo IP per comunicare. Tuttavia noi umani abbiamo bisogno di dare dei nomi ai computer perché è troppo difficile ricordare tutti i numeri. Da qui entra in gioco la necessità di usare DNS o il file `/etc/hosts`. Questa mappa di corrispondenza tra gli indirizzi IP e il nome dei computer (hostname) di solito non è necessaria. Tuttavia è necessario nel caso di un ambiente LTSP, perché se non ci fosse il sistema NFS darebbe dei permessi erronei quando una workstation tenta di montare il filesystem principale.

Oltre ai problemi con NFS, se la workstation non è compresa nel file `/etc/hosts`, potrebbero esserci problemi con i display manager *GDM* o *KDM*.

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

Nel file `lts.conf` ci sono parecchie voci che possono configurate.

Il file `lts.conf` ha una sintassi semplice. C'è una sezione di default (per i valori predefiniti) chiamata **[default]** e poi possono esserci delle sezioni per ciascuna singola workstation. La workstation può essere identificata dal nome del computer (hostname), dall'indirizzo IP oppure dal MAC-address.

Un tipico file `lts.conf` è mostrato qui sotto:

```

#

```

```
# 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. Its.conf file

La seguente è una lista di alcune voci:

XSERVER

Se la vostra scheda video è PCI e è supportata da X.org 6.7.0, allora avete bisogno soltanto del pacchetto `lts_x_core` package, che contiene tutti i moduli driver per X4.

Ci sono parecchi pacchetti per XFree86 3.3.6 disponibili per LTSP. Questo nel caso in cui la vostra scheda video non sia supportata da X.org 6.7.0.

Potete mettere una linea di configurazione nel file `lts.conf` per ogni workstation oppure potete metterne una nella sezione default, che verrà condivisa da tutte le workstation.

La nostra workstation, usata nell'esempio, ha una scheda video con chipset Intel i810 video e viene correttamente rivelata dal rivelamento automatico. Pertanto non abbiamo bisogno di una linea di configurazione XSERVER nel file `lts.conf` file. La linea di configurazione XSERVER può essere specificata, se volete, oppure può essere posta ad 'auto' per indicare che deve essere effettuato l'autodetect.

RUNLEVEL

Se vogliamo utilizzare la workstation n modalità grafica dobbiamo indicare come Runlevel il valore '5'. Questo viene fatto con l'apposita linea di configurazione del file `lts.conf`.

2.5. Visualizzazione della configurazione corrente

Con il programma **ltspcfg**, si può guardare lo stato corrente dei tutti i servizi necessari con LTSP. Dal menù principale (main menu) di `ltspcfg` premendo il tasto 'S' viene mostrato lo stato corrente.

```

xterm
ltspcfg v0.6          The Linux Terminal Server Project (http://www.LTSP.org)

```

| Interface | IP Address | Netmask | Network | Broadcast | Used |
|-----------|---------------|---------------|-------------|---------------|--------|
| eth0 | 192.168.0.254 | 255.255.255.0 | 192.168.0.0 | 192.168.0.255 | <----- |

| Service | Installed | Enabled | Running | Notes |
|------------|-----------|---------|---------|---------------------|
| dhcpd | Yes | no | no | Version 3 |
| tftpd | Yes | Yes | Yes | No '-s' flag |
| portmapper | Yes | no | Yes | |
| nfs | Yes | Yes | Yes | |
| xdmcp | Yes | no | Yes | xdm, gdm Using: gdm |

| File | Configured | Notes |
|------------------------------|------------|-------|
| /etc/hosts | Yes | |
| /etc/hosts.allow | Yes | |
| /etc/exports | Yes | |
| /opt/ltsp/i386/etc/ltsp.conf | Yes | |

```

Configured runlevel: 5      (value of initdefault in /etc/inittab)
Current runlevel: 5        (output of the 'runlevel' command)

Installation dir...: /opt/ltsp

Press <enter> to continue.. █

```

Figura 2–9. ltspcfg – Current Status

Capitolo 3. La Configurazione della workstation

Una volta configurato il server è arrivato il momento di configurare la workstation.

Il progetto LTSP copre tutto quello che succede dopo che il kernel è caricato in memoria. Ci sono diversi metodi per caricare il kernel in memoria. Tra questi ci sono: Etherboot, Netboot, PXE e disco floppy.

3.1. Avviare il computer con PXE

Se la vostra scheda di rete ha il supporto PXE o se lo ha il vostro computer, potete usarlo per caricare il kernel Linux. PXE è una tecnologia bootrom, simile a Etherboot or Netboot.

Potreste essere in grado di utilizzare il bootrom PXE sulla vostra scheda di rete. Bisogna cambiare il settaggio nel BIOS per far sì che il "Boot from LAN" sia la prima scelta tra i metodi di boot, invece di "floppy" o "hard disk".

Il sistema PXE ha la limitazione di essere in grado di caricare soltanto file che siano non più grandi di 32 kB. Il kernel di Linux è parecchio più grande. Pertanto non è possibile caricare direttamente il kernel Linux con PXE. Bisogna caricare qualcosa come un 'Network Bootstrap Program' o NBP.

Uno degli NBP disponibili per caricare il kernel Linux chiamato **pxelinux.0**. Fa parte del pacchetto **syslinux** per il kernel sviluppato da H. Peter Anvin.

Il pacchetto del kernel fornito con LTSP comprende lo NBP pxelinux.0 e i file di configurazione necessari per caricare il kernel Linux sul disco ram iniziale.

Il funzionamento avviene in questo modo:

- Il bootrom PXE inizializza la scheda di rete e manda una richiesta DHCP.
- Il server DHCP risponde fornendo un indirizzo IP e il nome del NBP da caricare..
- Il bootrom PXE prende lo NBP, lo mette in memoria e lo fa eseguire.
- Lo NBP usa tftp per far il download del file di configurazione dal server.
- Il file di configurazione contiene il nome del kernel, il nome del file del disco ram iniziale e le opzioni da passare al kernel.

Questo è un esempio di un file di configurazione per pxelinux:

```
prompt=0
label linux
    kernel bzImage-2.4.24-ltsp-4
    append init=/linuxrc rw root=/dev/ram0 initrd=initrd-2.4.24-ltsp-4.gz
```

- Lo NBP, a questo punto, usa tftp per fare il download del kernel Linux e del disco ram iniziale (initrd).
 - Quindi il controllo viene passato al kernel Linux, avviene la fase di boot, viene montato il disco ram iniziale (initrd) e prosegue l'avvio della workstation.
-

3.2. Avviare il computer con Etherboot

Etherboot è un programma software che permette di creare immagini ROM che possono essere downloadate sulla scheda Ethernet per essere eseguita da un computer x86. Molte schede di rete hanno un'alloggio in cui può essere collocato un chip ROM. Il codice Etherboot può essere collocato in questo chip ROM.

—Ken Yap

Etherboot è disponibile come Open Source, con la licenza GNU General Public License, Version 2 (GPL2).

Per usare Etherboot, se avete una scheda Ethernet con il bootrom Etherboot, avete bisogno di cambiare la configurazione del BIOS per dirgli di fare l'avvio dalla LAN ("Boot from LAN") prima di tentare di fare il boot dai dischi floppy o dall'harddisk.

Se non avete un bootrom Etherboot potete o fare un bootrom o potete fare un disco floppy con l'immagine di Etherboot nel suo boot sector.

Etherboot supporta un numero molto grande di schede di rete. Più di 200 modelli, e di nuovi ne vengono continuamente aggiunti Sia che decidiate di fare un disco floppy sia che mettiate il codice nel chip Eprom, avrete bisogno di conoscere il modello di scheda di rete che avete.

3.2.1. Scelta del driver Etherboot per la scheda di rete ISA

Per le vecchie schede di rete di tipo ISA non è così tanto importante determinare il tipo di scheda. Per prima cosa la maggior parte di queste schede sono schede ne2000 oppure 3Com 3c509. Vi basta prendere il driver Etherboot giusto, quello che utilizza il giusto sistema di collegamento a seconda della vostra scheda tra il 10 base-2 (Coax, per il cavo coassiale) e il 10 base-T (Twisted pair, per il cavo con i fili intrecciati).

3.2.2. Scelta del driver Etherboot per la scheda di rete PCI

Con le schede di rete PCI è essenziale che sia scelto il driver Etherboot che combaci con il numeri PCI Vendor e Device ID della scheda di rete.

Alcune volte può andarvi bene e conoscete esattamente quale sia il modello della vostra scheda perché c'è scritto sulla scheda e corrisponde esattamente alla descrizione di uno dei moduli di Etherboot. Tuttavia in molti casi dovrete ottenere i numeri di identificazione PCI.

se la vostra workstation) ha un disco floppy potete fare il boot con un disco tomsrtbt (Tom's Root Boot). Altrimenti se la vostra workstation ha un lettore di CD-ROM potete fare il boot con un disco Knoppix. Se non siete in grado di caricare il kernel Linux sulla vostra workstation allora la vostra unica speranza è quella di spostare la scheda di rete su un computer che riesca a fare il boot.

Una volta che avete avviato Linux, potete usare il comando **lspci** con l'opzione '-n'

```
[root@jamlap root]# lspci -n
0000:00:00.0 Class 0600: 8086:7190 (rev 03)
0000:00:01.0 Class 0604: 8086:7191 (rev 03)
0000:00:03.0 Class 0607: 104c:ac1c (rev 01)
0000:00:03.1 Class 0607: 104c:ac1c (rev 01)
0000:00:07.0 Class 0680: 8086:7110 (rev 02)
0000:00:07.1 Class 0101: 8086:7111 (rev 01)
```

```
0000:00:07.2 Class 0c03: 8086:7112 (rev 01)
0000:00:07.3 Class 0680: 8086:7113 (rev 03)
0000:00:08.0 Class 0401: 125d:1978 (rev 10)
0000:01:00.0 Class 0300: 1002:4c4d (rev 64)
0000:06:00.0 Class 0200: 8086:1229 (rev 09)
```

Nell'esempio qui sopra, si vede una linea per ogni scheda PCI presente nel sistema. Dovete guardare soltanto alle linee dei dispositivi con la classe **Class 0200**. Lanciando il comando in modo da guardare solo alle schede Ethernet la lista diventa più facile da leggere.

```
[root@jamlap root]# lspci -n | grep "Class 0200"
0000:06:00.0 Class 0200: 8086:1229 (rev 09)
```

In questo esempio i numeri PCI ID sono: **8086:1229**. Il primo campo è il PCI Vendor ID in questo esempio **8086** corrispondente alla Intel Corporation. Il secondo campo è **1229**, che è il PCI Device ID. Questo ci dice il modello della scheda di rete. In questo caso è una scheda EtherExpress 100.

3.2.3. Creare un disco floppy di avvio

Potete prelevare il pacchetto Etherboot package e configurarlo per il tipo di bootrom di cui avete bisogno. Quindi compilate il sorgente per produrre un immagine del bootrom che può essere scritta in EPROM oppure su un disco floppy.

Una soluzione più semplice è andare sul sito web www.Rom-O-Matic.net di Marty Connor.

Marty ha fatto un eccellente lavoro mettendo in piedi un'interfaccia per il web per la configurazione e la compilazione necessaria per creare un immagine bootrom con Etherboot. Sul suo sito, selezionate il tipo di scheda di rete che avete, e che tipo di immagine volete. Inoltre avete la possibilità di modificare alcune opzioni di configurazione del Etherboot. A questo punto premete il pulsante 'Get ROM' (ottieni ROM) e una immagine personalizzata del bootrom verrà generata in breve tempo.

Per avere il formato per il chip ROM scegliete 'Floppy Bootable ROM Image'. Questa scelta aggiunge un header di 512 byte che fa sì che il boot loader venga caricato in memoria da dove poi viene eseguito.

Dopo alcuni secondi che avete premuto il bottone, il vostro browser farà comparire una finestra pop-up "Save As" (o "Salva con nome") in cui sceglierete il nome con cui salvare il file sul vostro computer.

Una volta che avete salvata l'immagine sul vostro disco fisso dovete scriverla sul vostro floppy. Inserite il disco floppy nel vostro lettore di floppy e date il seguente comando per scrivere sul floppy:

```
dd if=Etherboot_Image of=/dev/fd0
```

dove al posto di *Etherboot_Image* dovete mettere il nome del vostro file immagine.

3.2.4. Creare un bootrom

Per scrivere l'immagine sulla eprom serve un sistema per la programmazione di EPROM. Questo strumento hardware ha un costo che varia, a seconda dei modelli, da poche centinaia di dollari a parecchie migliaia di dollari (da poche centinaia di euro a parecchie migliaia di euro)

Le modalità da seguire per creare il bootrom dipende completamente da sistema di programmazione di EPROM. Questo va al di là dello scopo di questa documentazione.

Capitolo 4. La workstation in fuzione

Se il server e la workstation sono configurati in modo corretto basterà inserire il disco floppy di boot e accendere il computer workstation.

Il codice Etherboot sarà letto dal floppy e messo in memoria, la scheda di rete viene identificata e inizializzata, viene inviata la richiesta dhcp attraverso la rete e la risposta sarà inviata dal server e il kernel sarà scaricato nella workstation. Quando il kernel avrà inizializzato l'hardware della workstation, partirà X windows e apparirà sullo schermo della workstation la schermata di login, come nell'esempio qui sotto.



Figura 4–1. Schermata di Login

A questo punto puoi loggarti. Un punto importante da tenere bene a mente è che entri e ti loggi sul server. Tutti i comandi che eseguti vengono in realtà eseguiti sul server e mostrano l'output sulla workstation. Questa è la forza, di X Windows.

Puoi eseguire qualsiasi programma che è supportato dal server.

Capitolo 5. La Stampa

Un computer così collegato oltre ad essere una workstation con una funzionalità completa in modalità GUI (grafica) o in modalità a caratteri, può servire anche da server di stampa. Fino a 3 stampanti possono essere attaccate alla porte parallele e seriali.

Tutto questo in maniera completamente trasparente per l'utente della workstation. Non si noterà neppure la piccola quantità di traffico che va dalla workstation alla stampante.

5.1. La parte della configurazione della workstation

LTSP usa il programma **lp_server** sulla workstation (terminale) per reindirizzare i lavori di stampa dal server alla stampante attaccata ad una delle porte della workstation (terminale).

Alcune linee del file di configurazione `lts.conf` permettono di abilitare le stampanti sulla workstation.

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

Le righe qui sopra fanno sì che il programma `lp_server` sia eseguito come demone, ascolti le connessioni TCP/IP sulla porta 9100 in attesa di uno flusso di stampa proveniente dal server. I dati da stampare saranno reindirizzati alla stampante attaccata alla porta parallela `/dev/lp0`.

Sono disponibili molte opzioni. Maggiori informazioni per la configurazione delle stampanti sono disponibili nella sezione di spiegazione del file di configurazione `lts.conf`.

5.2. La parte della configurazione del server

La configurazione della parte del server riguarda la definizione delle code di stampa (print queue), utilizzando lo strumento di configurazione delle stampanti sul server.

Nella distribuzione Red Hat 7.2 sono presenti strumenti di configurazioni delle stampa sia grafici che in modalità caratteri. Lo strumento grafico (GUI) si chiama **printconf-gui**, e quello in modalità testo si chiama **printconf-tui**. Versioni precedenti di Red Hat hanno un programma che si chiama **printtool**. Il programma `printtool` c'è anche in Red Hat 7.2, ma lancerà `printconf-gui`. Le altre distribuzioni di Linux dispongono di un proprio strumento per la configurazione delle stampanti `printer configuration tool`.

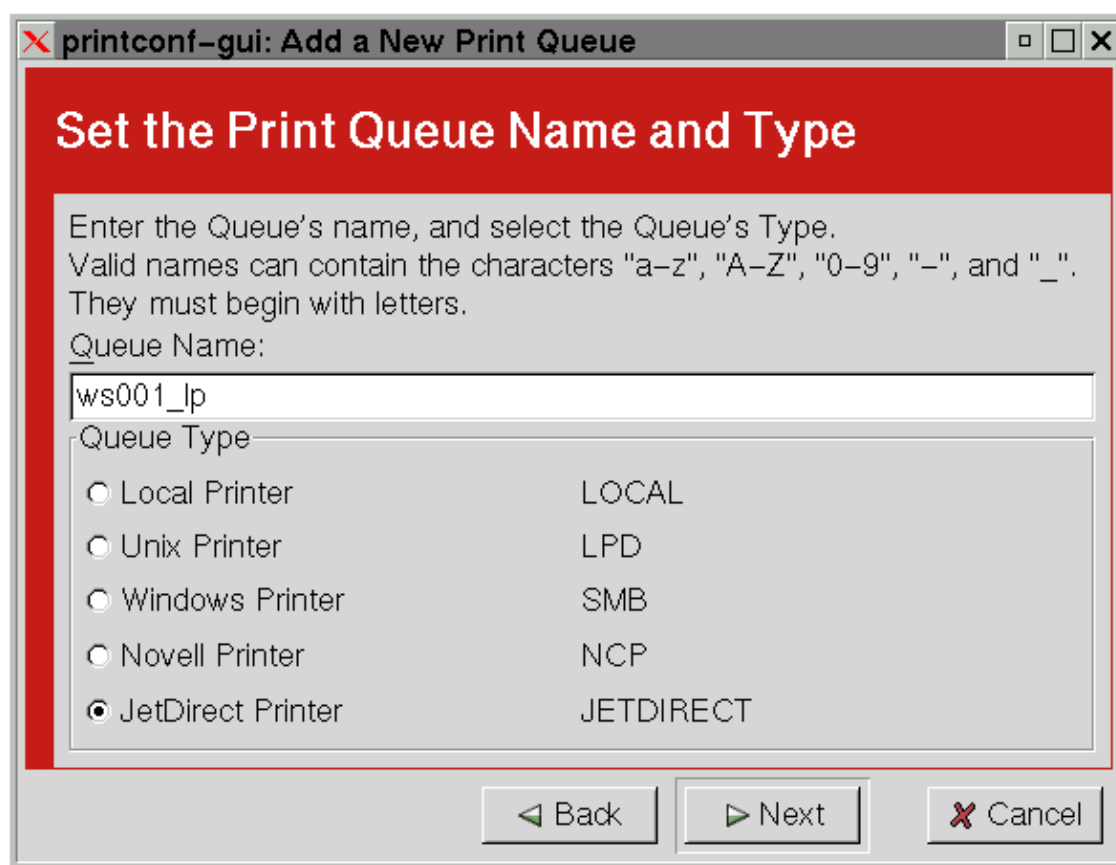


Figura 5–1. Aggiungere una nuova stampante con printconf-gui

Dopo aver lanciato lo strumento di configurazione delle stampanti bisogna aggiungere una nuova stampante,. Il programma `lp_server` permette alla workstation (terminale) di emulare un server di stampa HP JetDirect print server. Basta che create una stampante **JetDirect**.

Dovete dare un nome alla coda di stampa. Il nome può essere qualsiasi cosa, ma risulta più utile dare un nome che abbia un significato. Il nome della coda di stampa può contenere soltanto i seguenti caratteri:

- "a-z" lettere minuscole
- "A-Z" lettere maiuscole
- "0-9" cifre numeriche
- "-" trattino
- "_" :nbsp; segno di sottolineatura

Il nome della coda di stampa scelto nell'esempio di sopra è `ws001_lp`. Un nome di questo genere rende facile ricordarsi che è attaccata alla workstation (terminale) `ws001`.

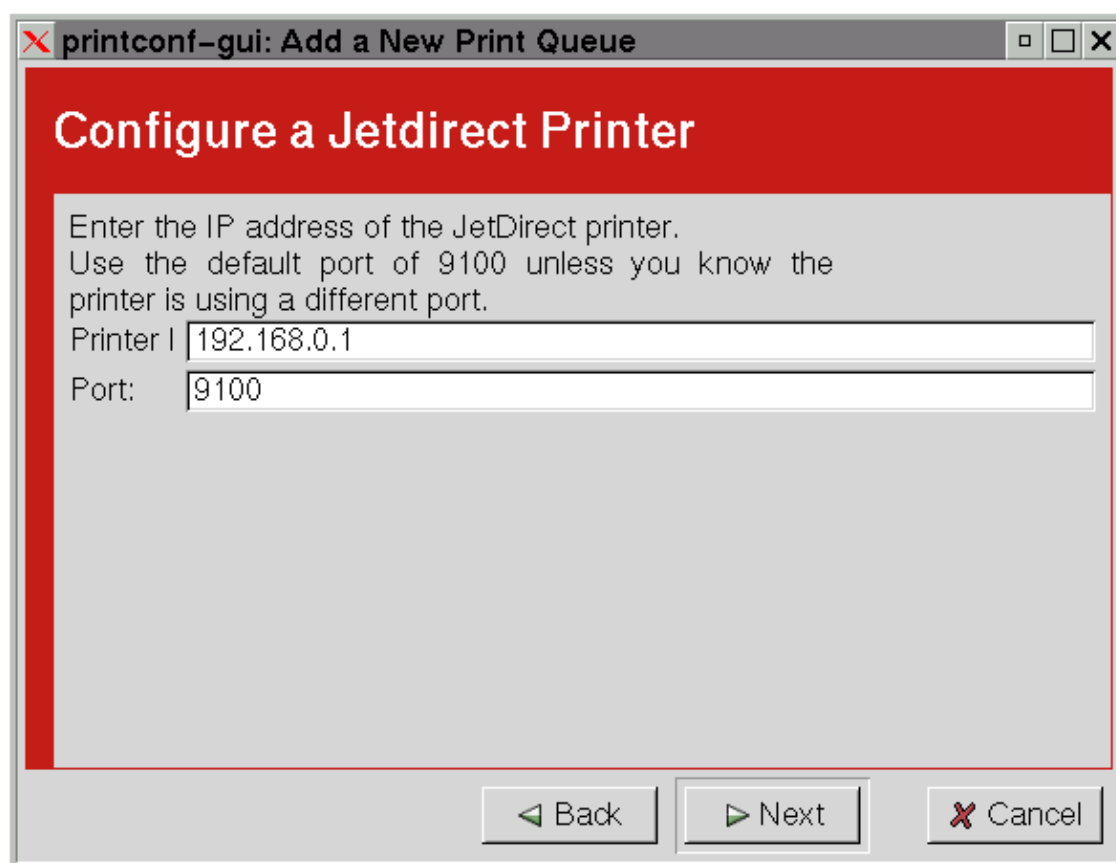


Figura 5–2. Dettagli Informazioni di Printconf–gui

Vi sono due campi da configurare necessariamente per poter comunicare con la stampante:

1. L'indirizzo IP o il nome del computer (hostname) della workstation cui è attaccata.
2. La porta TCP/IP su cui il demone **lp_server** è in ascolto.

La prima stampante connessa alla workstation (terminale) sarà sulla porta TCP/IP **9100**. La seconda stampante sarà sulla porta **9101**, e la terza sulla porta **9102**.

Capitolo 6. Gli script per le sessioni dello schermo

Una caratteristica che è stata affinata nella versione 4.0 di LTSP è quella denominata **Screen Scripts** (Script per le sessioni dello schermo). Questi script permettono di far partire sessioni di diverso tipo.

Si può specificare più di uno screen script per una workstation. In questo modo si ottengono delle sessioni multiple. Possono essere sessioni di diverso tipo o sessioni dello stesso tipo. Per esempio è possibile specificare nel seguente modo:

```
SCREEN_01 = startx
SCREEN_02 = shell
```

Questa configurazione farà partire un server X (una sessione grafica) sul primo schermo e una sessione a caratteri (con un prompt di una shell) sul secondo schermo. Per passare al primo schermo si premono contemporaneamente i tasti Ctrl–Alt–F1 e per passare al secondo schermo si premono contemporaneamente i tasti Ctrl–Alt–F2.

Si possono specificare fino a 12 screen script per workstation. Tuttavia nella maggior parte dei casi basta averne uno solo.

I possibili tipi di screen scripts sono:

startx

Questo script lancerà il server X con l'opzione `–query`. Con questa opzione verrà fatta una richiesta XDCPM al display manager del server per ottenere una schermata di login sullo schermo.

shell

Questo script lancerà una finestra shell sul terminale. Questo serve principalmente per risolvere eventuali problemi con la workstation. Fornisce una sessione sulla workstation e non sul server. Pertanto non è utile per eseguire delle applicazioni.

telnet

Questo script lancia una sessione telnet che si conatterà al server dando una sessione in modalità carattere con il server.

Di default telnet si conatterà al server LTSP. Se volete indicare un server differente, potete passare il suo nome sulla stessa linea del nome del screen script. Per esempio:

```
SCREEN_01 = telnet server2.mydomain.com
```

Si può anche specificare una qualsiasi altra opzione che telnet sia in grado di comprendere. Però se si specifica una o più opzioni deve anche essere indicato il nome del server a cui connettersi.

rdesktop

Questo script lancerà il programma rdesktop, che si collegherà ad un server Microsoft Window server. Sulla linea è possibile specificare, dopo il nome dello screen script, qualsiasi opzione rdesktop. Per esempio, se si vuole specificare il server a cui connettersi, si può fare nel seguente modo:

```
SCREEN_01 = rdesktop -f w2k.mydomain.com
```

Con la configurazione di questo esempio verrà lanciato rdesktop in modalità tutto schermo. L'utente vedrà una finestra di login di Windows e avrà bisogno di loggarsi una volta sola. Questa cosa è molto utile quando avete

bisogno soltanto di un finestra di login Windows senza avere la login Linux o il window manager. L'utente non saprà neppure di essere su una macchina Linux.

I file che contengono gli screen script risiedono nella directory `/opt/ltsp/i386/etc/screen.d`. Potete anche creare il vostro script personalizzato e collocarlo in questa directory. La cosa migliore, in questo caso, è partire da uno script già esistente ed usarlo come esempio.

Capitolo 7. Risoluzione dei problemi (Troubleshooting)

Se, dopo aver seguito quanto descritto nei capitoli precedenti, la vostra workstation non si avvia allora dovete seguire il processo per la risoluzione degli errori (trubleshooting) dell'installazione. Cioè dovete seguire i passi qui sotto descritti.

La prima cosa da fare è determinare fino a che punto del procedimento d'avvio è riuscita ad arrivare la workstation.

7.1. Risoluzione dei problemi nel caso dell'immagine Etherboot su dischetto floppy

Quando parte il boot dal floppy dovrebbe apparire qualcosa del tipo:

```
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 riportato qui sopra mostra cosa dovrete aspettarvi di vedere quando avviene il boot da floppy. Se non vengono visualizzati questi messaggi, che indicano che è partito l'Etherboot, allora probabilmente il dischetto floppy è sbagliato: o il dischetto floppy è danneggiato o l'immagine non è stata scritta sul floppy in modo corretto.

Se viene visualizzato un messaggio del tipo riportato qui sotto, allora probabilmente vuol dire che l'immagine che è stata generata non è quella adatta per la vostra 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 arriva al punto in cui rivela la scheda di rete e mostra il corretto MAC address, allora il dischetto floppy è probabilmente corretto e funzionante.

7.2. Risoluzione dei problemi con DHCP

Dopo che la scheda è stata inizializzata, viene inviato una richiesta DHCP in broadcasting sulla rete locale, per trovare il server DHCP.

Se la workstation riceve una risposta valida dal server DHCP, allora significa che la scheda di rete è stata configurata correttamente. Potete rendervi conto che tutto, fino a qui, è andato bene dal fatto che sullo schermo vengano visualizzate le informazioni sull'indirizzo IP . Qui sotto è riportato ad esempio cosa potrebbe apparire:

```
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 vedete una linea che inizia per 'Me:' seguito da un indirizzo IP, allora potete considerare che il sistema DHCP funziona correttamente e potete passare a controllare la parte del TFTP.

Se invece appare il seguente messaggio sulla workstation, seguito da tanti messaggi <sleep>, allora c'è qualcosa che non va. Tuttavia è possibile che appaiono uno o due messaggi <sleep> e dopo la risposta del DHCP in questo caso la situazione è normale.

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

Scoprire dove sia l'errore può essere, a volte, difficile. Ecco alcune cose da controllare.

7.2.1. Controllare le connessioni

La workstation è fisicamente connessa alla stessa rete cui è connesso il server?

Quando la workstation viene accesa, controllate che si accendano le luci del link su tutte le connessioni.

Se la connessione è diretta dalla workstation al server (senza passare da hub o switch) accertatevi che state utilizzando un cavo cross-over. Se state passando da un hub o da uno switch, allora accertatevi che state utilizzando un cavo dritto (chiamato anche patch cable) sia dalla workstation all'hub che dall'hub al server.

7.2.2. Il sistema DHCP è in funzione?

Bisogna determinare se il sistema **dhcpcd** sia in funzione sul server. Questo può essere determinato in diversi modi.

Il programma **dhcpcd** normalmente diventa demone in background e ascolta la porta udp numero 67. Provate a eseguire il comando **netstat** e guardate se c'è qualcosa in ascolto su quella porta. Dando il comando:

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

si dovrebbe vedere un output del tipo:

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

La quarta colonna contiene l'indirizzo IP e la porta separata dal simbolo dei due punti (:). Un indirizzo composto da tutti zero ('0.0.0.0') indica che l'ascolto è effettuato su tutte le interfacce. Cioè se avete sia una interfaccia **eth0** ed una **eth1** vuol dire che **dhcpcd** da ascoltando entrambe le interfacce..

Tuttavia il fatto che netstat mostri che c'è qualcosa che sta ascoltando la porta udp numero 67 non vuol dire che sia sicuramente **dhcpcd** ad ascoltare. Potrebbe essere **bootpd**, ma è poco probabile perché **bootp** non è più presente nella maggior parte delle distribuzioni Linux.

Per accertarvi che **dhcpcd** sia in esecuzione, provate a eseguire il comando **ps**.

```
ps aux | grep dhcpcd
```

Dovreste vedere qualcosa del tipo:

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

La prima riga mostra che il programma **dhcpcd** è in esecuzione. La seconda riga è semplicemente il comando **grep** che stiamo eseguendo adesso.

Se non vedete alcuna riga che ostra che dhcp è in esecuzione dovete controllare che il server sia configurato per il runlevel 5 e che il sistema è configurato in modo che **dhcpcd** venga avviato quando il sistema è in runlevel 5. Sui sistemi basati su Red Hat potete usare **ntsysv** per controllare questa cosa.

Potete tentare di avviare il **dhcpcd** con questo comando:

```
service dhcpcd start
```

Prestate attenzione all'output, e controllate se sono evidenziati degli errori.

7.2.3. Controllo incrociato della configurzzione dhcpcd

Il file `/etc/dhcpcd.conf` ha una linea per la workstation in esame?

Dovete fare un controllo incrociato del indirizzo fisso (fix-address) nel file di configurazione per accertarsi che corrisponda alla workstation.

7.2.4. La richiesta a ipchains o a iptables è stata fatta?

7.2.4.1. Controllare ipchains

Usate il seguente comando e guardate cosa vi dice:

```
ipchains -L -v
```

Se vedete qualcosa del tipo:

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

allora non è ipchains che è responsabile dei problemi.

7.2.4.2. Controllare iptables

Usate il seguente comando e guardate cosa vi dice:

```
iptables -L -v
```

Se vedete qualcosa del tipo:

```
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
```

allora non è iptables che è responsabile dei problemi.

7.2.5. La workstation sta spedendo la richiesta?

Provate a guardare il file `/var/log/messages` mentre la workstation si sta avviando. Potete utilizzar il seguente comando:

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

Questo comando continuerà a mostravi le nuove righe registrate nel file.

```
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 vedete dei messaggi come quelli qui sopra, la scritta 'no free leases', indica che **dhcpd** è in funzione, ma non sa nulla della workstation che sta richiedendo un indirizzo IP.

7.3. Risoluzione dei problemi con TFTP

Etherboot usa il TFTP per prelevare il kernel Linux dal server. Il TFTP è un protocollo relativamente semplice, ma ogni tanto è possibile che ci si imbatta in qualche problema nel farlo funzionare.

Se vedete un messaggio simile a questo:

```
Loading 192.168.0.254:/lts/vmlinuz-2.4.24-ltsp-4.....
```

Con dei puntini che compaiono sullo schermo abbastanza velocemente significa che il TFTP sta funzionando correttamente e che sta avvenendo il download del kernel.

Se invece non compaiono i puntini, allora c'è qualche problema. Tra le possibilità da considerare ci sono:

7.3.1. tftpd non è in esecuzione

Se **tftpd** non è configurato in modo che venga eseguito allora certamente non sarà in grado di rispondere alle richieste della workstation. Controllate se è in esecuzione con **netstat** usando un comando del tipo:

```
[root@bigdog]# netstat -anp | grep ":69 "
udp        0      0 0.0.0.0:69          0.0.0.0:*           453/inetd
```

Se non vedete nessun output dopo aver dato il comando allora è molto probabile che **tftpd** non è in esecuzione.

Vi sono due metodi per avviare **tftpd**, Sono **inetd** e il più moderno **xinetd**

inetd usa un file di configurazione chiamato `/etc/inetd.conf`. Accertatevi che in questo file la riga che fa partire **tftpd** NON sia commentata. la linea dovrebbe essere qualcosa di simile a:

```
tftp dgram udp wait nobody /usr/sbin/tcpd /usr/sbin/in.tftpd -s /tftpboot
```

xinetd usa una directory con contiene dei file individuali di configurazione. Ciascun file per ogni servizio. Se il vostro server utilizza **xinetd**, allora il file di configurazione per **ftpd** si chiama `/etc/xinetd.d/tftp`. Qui sotto c'è un esempio di come può essere questo file:

```

service tftp
{
    disable          = no
    socket_type      = dgram
    protocol         = udp
    wait             = yes
    user             = root
    server           = /usr/sbin/in.tftpd
    server_args      = -s /tftpboot
}

```

Accertatevi che la linea **disable** NON sia posta a **yes**.

7.3.2. Il Kernel non è dove tftpd pensa che sia

Il kernel deve essere in un posto a cui il demone tftpd possa accedere. Se l'opzione '-s' è stata specificata per il demone **tftpd**, allora qualunque cosa la workstation richieda deve essere richiesta relativamente alla directory /tftpboot. Pertanto, se il parametro **filename** nel file /etc/dhcpd.conf ha il valore /lts/vmlinuz-2.4.24-ltsp-4, il kernel dovrà in realtà essere /tftpboot/lts/vmlinuz-2.4.24-ltsp-4

7.4. Risoluzione dei problemi con il filesystem principale NFS

Ci sono parecchie motivi per cui potrebbero essere la causa per cui non riesca a montare il filesystem principale. Tra queste:

7.4.1. Non viene trovato il file init (No init found)

Se ricevete il seguente messaggio d'errore:

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

allora è molto probabile che state montando come filesystem principale una directory sbagliata o che la directory /opt/ltsp/i386 sia vuota.

7.4.2. Il server ritorna error -13

Se ricevete il seguente messaggio d'errore:

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

questo significa che la directory /opt/ltsp/i386 non è elencata nel file /etc/exports file.

Controllate il file /var/log/messages per vedere se riuscite a scoprire qualcosa. Una linea del tipo:

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

confermerà il sospetto che la lista /etc/exports non è corretta.

7.4.3. Problemi con il demone NFS (portmap, nfsd & mountd)

Risolvere i problemi di NFS può rivelarsi una cosa complicata. Comprendere come dovrebbe essere il setup e quali sono gli strumenti utili a diagnosticare il problema renderà sicuramente le cose più facili.

Ci sono tre demoni che devono essere in esecuzione sul server perché NFS funzioni correttamente. **portmap**, **nfsd** e **mountd**.

7.4.3.1. portmap

Se vedete dei messaggi del tipo:

```
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
```

allora molto probabilmente questo è dovuto al fatto che **portmap** non è in esecuzione. Potete accertarvi della cosa utilizzando il comando **ps**, ad esempio: command:

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

se portmap è in esecuzione dovrebbe esserci una linea del tipo:

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

Un'altro test è quello di utilizzare il comando **netstat**. Le porte TCP ed UDP utilizzate da portmap sono le 111. Provate con il comando:

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

Dovreste vedere, come risultato, qualcosa di questo tipo:

```
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 vedete qualcosa di simile allora probabilmente portmap non è in esecuzione.

Potete avviare portmap con il comando:

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

Quindi accertatevi che portmap sia correttamente configurato per essere avviato ogni volta che il server viene acceso. Utilizzate il comando **ntsysv** per accertarvi di ciò.

7.4.3.2. I demoni NFS e MOUNT (nfsd & mountd)

NFS ha 2 demoni che devono essere in funzione: **nfsd** e **mountd**. Entrambi vengono fatti partire dallo script `/etc/rc.d/init.d/nfs`.

Potete utilizzare il comando **ps** per accertarvi che siano in esecuzione.

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

se scoprite che uno o entrambi non sono in funzione dovete farli partire.

Normalmente dovrete essere in grado di far eseguire lo script di avvio con il parametro **restart** che dovrebbe farli ripartire tutte e due. Tuttavia lo script `/etc/rc.d/init.d/nfs`, per qualche oscura ragione, non riavvia **nfsd**, bensì riavvia solo **mountd** (forse questo è un bug). Pertanto dovete dare questi due comandi in sequenza:

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

Non dovete preoccuparvi dei messaggi d'errore che ricevete nella fase di stop. Il comando **start** deve avere OK come risultato.

Se entrambi i due demoni sono in funzione ma NFS continua a non funzionare potete controllare che si siano registrati a portmap utilizzando il comando **rpcinfo**

```
rpcinfo -p localhost
```

Dovreste ottenere qualcosa:

```
program vers proto  port
 100000    2   tcp    111  portmapper
 100000    2   udp    111  portmapper
 100003    2   udp    2049 nfs
 100003    3   udp    2049 nfs
 100021    1   udp    32771 nlockmgr
 100021    3   udp    32771 nlockmgr
 100021    4   udp    32771 nlockmgr
 100005    1   udp    648  mountd
 100005    1   tcp    651  mountd
 100005    2   udp    648  mountd
 100005    2   tcp    651  mountd
 100005    3   udp    648  mountd
 100005    3   tcp    651  mountd
 100024    1   udp    750  status
 100024    1   tcp    753  status
```

Questo indica che **nfs** (più propriamente **nfsd**) e **mountd** sono entrambi in funzione e si sono registrati al portmap.

7.5. Risoluzione dei problemi con X

Beh ... Probabilmente è la parte più difficile della configurazione della workstation LTSP è proprio riuscire ad avere il server X configurato correttamente. Se state usando una scheda video abbastanza recente ed è supportata da Xservers di Xorg e avete un monitor abbastanza recente, che possa utilizzare un vasto numero di possibili frequenze, allora il lavoro dovrebbe essere abbastanza semplice. Nel caso in cui non funzioni in genere è molto probabile che sia dovuto al fatto che si sta usando lo X server non adatto al tipo di scheda video.

Scoprire se il server X non funziona è relativamente semplice: in questo caso o non partirà neppure o quanto mostrato sullo schermo non è corretto.

Quando la workstation è pronta per far partire il server X chiama lo script `startx`, che avvia il sever X sulla workstation locale, con il parametro **-query** che punta al server, su cui deve essere in funzione un display

manager come **XDM**, **GDM** o **KDM**.

Poiché il server X è avviato dallo script `startx` che a sua volta è avviato dal programma **init**, quando fallisce il programma **init** tenterà nuovamente di lanciarlo. **init** continuerà questa procedura (loop) di provare a lanciare il server X 10 volte e poi deciderà di abbandonare perché pensa che stia avvenendo un respawning troppo velocemente. Alla fine di tutto questo un messaggio di errore del server X dovrebbe essere rimasto sullo schermo.

Aspettare che il server X fallisca per 10 volte è abbastanza frustrante. Un modo per evitare questa attesa è di avviare la workstation in runlevel 3 in modo che il server X NON venga avviato automaticamente. Quando avviate la workstation in questo modo otterrete un prompt della shell **bash**. Dal prompt della shell bash potrete avviare il server X manualmente con il comando:

```
sh /tmp/start_ws
```

Il server X tenterà di avviarsi e non appena il tentativo fallisce per la prima volta ritornerà al prompt della shell bash. In questo modo potete vedere il motivo del fallimento.

7.6. Risoluzione dei problemi con il Display manager

Il display manager è il demone che è in esecuzione sul server in attesa che un server X lo contatti. Quando viene effettuato un contatto apre una finestra di login sullo schermo, dando la possibilità all'utente di loggarsi sul server.

I tre più comuni display manager sono:

- XDM – è in giro da moltissimo tempo ed è incluso nella distribuzione standard del sistema X window. standard X windows system.
- GDM – The 'Gnome Display Manager' (Il Display Manager di GNOME). Fa parte della distribuzione Gnome.
- KDM – The 'KDE Display Manager' (Il Display Manager di KDE). Fa parte di KDE, la distribuzione K Desktop.

La maggior parte delle recenti distribuzioni GNU/Linux hanno tutti questi tre display manager.

7.6.1. Una schermata grigia con soltanto una grande cursore a forma di X

Se siete in questa situazione significa che il server X è in esecuzione ma che non è stato in grado di contattare il display manager. Alcuni possibili motivi sono:

1. Il display manager non è in esecuzione

Nelle versioni recenti di Redhat (dalla 7.0 in poi), il display manager viene fatto partire da **init**. Nel file `/etc/inittab` c'è una riga del tipo:

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

Lo script **prefdm** si preoccupa di determinare quale display manager avviare.

Quale sia il display manager di default dipende da quali pacchetti siano stati installati. Se è stato installato Gnome allora il display manager di default è GDM. Se GDM non è stato installato allora

prefdm controllerà se è stato installato KDE, nel qual caso il display manager di default sarà KDM. Se non è stato installato neanche KDE allora il display manager di default sarà XDM.

Potete usare il comando **netstat** per vedere se c'è un windows manager in funzione. Sul server date il seguente comando:

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

Dovreste vedere nei risultati che c'è un processo che è in ascolto sulla porta xdmcp (177).

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

Questo mostra chiaramente che **gdm** è in esecuzione, con PID (numero identificativo del processo) pari a 1493, e in ascolto sulla porta xdmcp.

Se vedete una linea simile a quella mostrata sopra, che indica chiaramente che c'è un display manager in ascolto, allora rimane da accertarsi se la workstation stia inviando correttamente la richiesta XDMCP al giusto server.

All'interno del file `lts.conf` può esserci un parametro che specifica l'indirizzo IP del server su cui c'è il display manager. Le linee con questi parametri sono opzionali, ma se presenti devono essere di questo tipo:

```
XDM_SERVER = 192.168.0.254
```

Ovviamente l'indirizzo IP deve essere quello corretto per la vostra rete e non quello dell'esempio qui sopra.

Se il parametro 'XDM_SERVER' non è presente allora sarà usato il valore presente nel parametro 'SERVER'. se presente. Se non presente allora sarà usato **192.168.0.254**.

Dovete accertarvi che l'indirizzo IP, indipendentemente da come sia specificato, sia realmente l'indirizzo IP corretto del server su cui è in funzione il display manager.

2. Può essere che il display manager sia configurato per ignorare le richieste degli host remoti (degli altri computer).

Se avete scoperto che il display manager è in funzione, allora è possibile che sia stato configurato per ignorare le richieste XDMCP provenienti da computer remoti. Occorre controllare il file di configurazione dello specifico display manager che è in funzione per verificare che sia configurato in maniera opportuna.

◆ XDM

La configurazione di default di XDM nella distribuzione Red Hat è quella di disabilitare la possibilità che una workstation remota ottenga il login. Lo script **lts_initialize** dovrebbe fare in modo di abilitare queste connessioni. Potrebbe però essere che questo non funzioni. Se non funziona dovete controllare il file `/etc/X11/xdm/xdm-config`. Controllate che ci sia una linea del tipo:

```
DisplayManager.requestPort: 0
```

Questa linea DEVE essere commentata per far sì che XDM ascolti le richieste remote sulla porta 177.

Un altro file di configurazione importante a riguardo dei login remoti con XDM è il file `/etc/X11/xdm/Xaccess`. In questo file DEVE esserci una linea che inizi con un asterisco '*'. Questa linea è normalmente presente nel file, ma nelle ditribuzioni Red Hat è commentata. Lo script **ltsp_initialize** dovrebbe aver sistemato questa situazione. Tuttavia se dovesse continuare a sembrare che XDM non funziona dovete controllare questo file. Una linea valida deve essere di questo tipo:

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

◆ KDM

Le nuove versioni di KDM hanno un file chiamato **kdmrc**. Le diverse distribuzioni di Linux collocano questo file in diversi punti. Nel caso di Red Hat 7.2 è in `/etc/kde/kdm/kdmrc`. Per trovare dove sia in altre distribuzioni utilizzate il comando **locate**.

C'è una sessione, che si chiama **[Xdmcp]**, che controlla se una workstation remota possa ottenere la login. Assicuratevi che il parametro **Enable** sia impostato a **true**.

Le versioni più vecchie di KDM usano il file di configurazione di XDM che è collocato in `/etc/X11/xdm`.

◆ GDM

GDM usa un gruppo di file di configurazione. Sono collocati nella directory `/etc/X11/gdm`.

Il file principale da guardare è il file `gdm.conf`. Guardate la sezione **[xdmcp]**. Dovrebbe esserci un parametro, all'interno di questa sezione, chiamato 'Enable'. Questo parametro deve essere impostato a '1' o a 'true' a seconda delle versioni di GDM. Qui sotto è riportato un esempio:

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

Notate la linea 'Enable=true'. Le versioni più vecchi utilizzano '0' e '1' per indicare Disable e Enable delle connessioni remote XDMCP. Le versioni più moderne utilizzano 'false' e 'true'.

3. Se il display manager è sicuramente in funzione ed è in ascolto alle richieste delle workstation remote potrebbe essere che il problema sia semplicemente che il display manager non è in grado di fare la conversione dall'indirizzo IP al nome del computer (hostname). C'è bisogno che la workstation sia inclusa nella lista del file `/etc/hosts`, oppure che siano correttamente configurate le tabelle del DNS.

Capitolo 8. I Kernel

Ci sono alcune decisioni da prendere a riguardo del kernel che si vuole far girare sulla workstation. Potete decidere di utilizzare un versione standard del kernel, disponibile per il download, oppure di costruirne uno voi stessi. E dovete decidere se volete visualizzare le schermate grafiche, complete con la barra di progressione, cosa possibile grazie al **Linux Progress Patch (LPP)**.

8.1. Kernel standard forniti con LTSP

Il pacchetto del kernel fornito con LTSP contiene in realtà due kernel. Una con la Linux Progress Patch già applicata e configurata, e l'altra che non ha questa patch applicata.

Entrambi i kernel hanno la patch NFS Swap applicata.

8.2. Costuitevi il vostro kernel

Ci sono due modi di configurare un kernel per LTSP. Il metodo usuale è quello di utilizzare un 'Disco Ram Iniziale' o **initrd** in breve. L'immagine **initrd** immagine è un piccolo filesystem che viene aggiunto al kernel. Il filesystem **initrd** è caricato in memoria e quando il kernel ha finito il boot lo monta come filesystem principale. Ci sono alcuni vantaggi ad utilizzare l'immagine **initrd**. Come prima cosa permette di compilare i driver per le schede di rete come moduli e di caricare il modulo corretto durante il boot. Questo permette di avere un solo kernel che è in grado di supportare praticamente tutte le schede di rete. Un altro vantaggio è che possiamo eseguire il client DHCP nello spazio dell'utente anziché nello spazio del kernel. Eseguire il client DHCP nello spazio dell'utente fornisce un migliore controllo sulle opzioni richieste e ricevute dal server. Inoltre questo fa diventare il kernel un po' più piccolo. L'altro modo di configurare il kernel è senza **initrd**. La costruzione di un kernel senza **initrd** richiede che il driver della specifica scheda di rete sia staticamente unito al kernel e richiede che siano impostati i parametri IP-Autoconfig e "Root filesystem on NFS" quando si compila il kernel. Il vantaggio di non utilizzare **initrd** è che leggermente più piccolo e farà fare il boot più velocemente. Una volta che la workstation è stata avviata ed è in funzione teoricamente non c'è alcuna differenza a riguardo di come funzioni la workstation.

Il kernel standard per LTSP include un disco ram iniziale (**initrd**) che si prende carico di identificare la scheda di rete e di fare la richiesta DHCP nello spazio dell'utente. Uno degli obiettivi ricercati nel fare questa immagine è stata quella di farla più piccola possibile. Per questo motivo abbiamo scelto di utilizzare la libreria **uClinux** al posto della **libc** e **busybox** per le utility necessarie durante il boot.

Se volete costruirvi il vostro kernel dovete procurarvi il pacchetto **ltspl_initrd_kit**. Contiene la gerarchia del filesystem principale e uno script per costruire l'immagine.

8.2.1. Ottenere i sorgenti del kernel

Quando si costruisce un kernel personalizzato, è generalmente una buona idea partire con dei sorgenti non modificati, presi direttamente dal sito **ftp.kernel.org**. Questo perché molte distribuzioni, tra cui ad esempio Red Hat, applicano molte patch ai sorgenti e l'insieme dei sorgenti che forniscono non coincide con quello ufficiale del kernel.

Scaricate il pacchetto dei sorgenti del kernel di vostra scelta e salvatelo nella directory `/usr/src`. I kernel sono collocati nella directory `/pub/linux/kernel` del server `ftp.kernel.org` ftp server. Avrete bisogno di una versione recente (la 2.4.x), perché avete bisogno di includere il supporto **devfs**.

Inoltre se volete applicare le patch per lo swapping su NFS o la Linux Progress Patch (LPP), avrete bisogno di prendere le patch che corrispondono alla versione del kernel che vi siete procurati. Al momento della scrittura di questo documento la versione 2.4.9 del kernel è la più nuova che supporta queste caratteristiche.

In questo esempio useremo il kernel 2.4.9 kernel. Il percorso completo è

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

Scomprimate i sorgenti del kernel nella directory `/usr/src`. Dovete prestare attenzione perché quando scomprimate il file tar verrà decompresso nella directory `linux`. Potreste già avere una directory chiamata `linux` con un diversa versione del codice sorgente e di certo non volete che uno copra l'altro. Pertanto, prima di estrarre il file tar, controllate se già esiste una directory con il nome `linux` e se esiste rinominatela con un altro nome.

Il pacchetto dei sorgenti che abbiamo scaricato è stato compresso con l'utility di compressione **bzip2**. Pertanto dobbiamo scomprimerla prima di passarla al programma **tar**. Potete utilizzare il seguente programma per estrarre il pacchetto:

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

Quando avete completato l'estrazione del pacchetto avrete una directory chiamata `linux` contenente l'intero albero del codice sorgente. A questo punto in genere rinomino la directory ad un nome più significativo.

```
mv linux linux-2.4.9
```

Una volta che è stata rinominata la directory, cambiamo la directory di lavoro entrando dentro la nuova directory

```
cd linux-2.4.9
```

In genere modifico il file `Makefile` prima di configurare il nuovo kernel. Nella parte iniziale del file `Makefile` c'è una variabile chiamata **EXTRAVERSION**. La imposto a `'ltsp-1'`. In questo modo il numero di versione completo del kernel diventa `'2.4.9-ltsp-1'`, cosa che permette di identificare questo kernel nel seguito. Dopo aver fatto questo la parte iniziale del file `Makefile` sarà qualcosa del tipo:

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

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

8.2.2. Le Patch del Kernel

Dopo aver estratto il kernel, potreste volere applicare alcune patch. Ad esempio la patch per NFS Swap o quella per il la barra d'avanzamento (Linux Progress Patch). Queste patch DEVONO essere applicate PRIMA di configurare il kernel (se volete applicarle).

8.2.2.1. NFS Swap patch

La patch NFS Swap permette al kernel della workstation di utilizzare un file di swap collocato sul server NFS. Mentre in genere si raccomanda di avere abbastanza memoria sulla workstation da non richiedere lo swapping, in alcuni casi, specie nel caso di vecchi computer, potrebbe essere difficile aggiungere altra memoria. La possibilità di fare lo seap tramite NFS può rendere utilizzabile un computer altrimenti inutilizzabile.

Se la directory corrente è /usr/src/linux-2.4.9, e la patch è in /usr/src, potete dare il seguente comando per verificare la patch:

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

Questo comando farà solo una prova, per controllare che possa essere applicata senza problemi. se il test funziona senza errori, potete applicare la patch ripetendo il comando precedente senza l'opzione **---dry-run**.

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

8.2.2.2. Linux Progress Patch (LPP)

La patch Linux Progress Patch (LPP) permette di configurare un'immagine grafica che venga mostrata durante la fase d'avvio. I normali messaggi che apparirebbero sullo schermo sono dirottati ad un altro schermo tty e alcune speciali istruzioni, aggiunte agli script d'avvio, fanno sì che l'avanzamento della barra mostrata sullo schermo rifletta l'avanzamento del processo d'avvio.

come nel caso della patch NFS Swap patch potete verificare questa patch con il comando:

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

Se il test funziona senza errori, potete applicare la patch con:

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

8.2.3. Configurare i parametri del kernel

Potete ora eseguire il programma che preferite per la configurazione. Tra le varie possibilità ci sono:

- make xconfig

Utility di configurazione del kernel nella versione X Windows.

- make menuconfig

Utility di configurazione del kernel nella versione basata su curses

- make config

Utility di configurazione del kernel nella versione con una semplice interfaccia a caratteri riga per riga.

8.2.3.1. Configurazione del Kernel per l'uso con initrd

La configurazione del kernel per l'uso di initrd richiede che siano impostate le seguenti opzioni:

- File systems → /dev filesystem support

Il supporto per il filesystem /dev deve essere abilitato. Questa impostazione si trova nella sezione 'File systems'. NON specificate 'Automatically mount at boot'. L'operazione di mount di questo filesystem verrà fatto dallo script /linuxrc.

- Block devices → RAM disk support

Le workstations LTSP hanno bisogno del supporto del kernel per i dischi RAM. Questa impostazione è nella sezione 'Block devices'.

- Block devices → Initial RAM disk (initrd) support

Anche questo deve essere attivato.

- Processor type and features → Processor family

Dovete accertarvi che il kernel che stato costruendo possa veramente essere eseguito sulla CPU della workstation. Controllate la sezione 'Processor type and features'. Dovete anche disattivare il supporto SMP, a meno che non abbiate veramente delle CPU multiple.

- File systems → Network file systems → NFS Client support

La workstation monta il filesystem principale tramite NFS, pertanto è necessario che il supporto per il client NFS sia attivato.

Questo per quanto riguarda le impostazioni necessarie. Potete anche disattivare parecchie funzioni del kernel, per ridurre le dimensioni del kernel.

8.2.3.2. Configurazione del Kernel per l'uso senza initrd

La configurazione del the kernel per l'uso senza initrd è diversa da quella con initrd per alcuni aspetti:

- Block devices → RAM disk support

Le workstation LTSP hanno bisogno del supporto del kernel per i dischi RAM. Questa impostazione è nella sezione 'Block devices'.

- Block devices → Initial RAM disk (initrd) support

Questa opzione deve essere disabilitata.

- Networking options → IP:kernel level autoconfiguration

Questo deve essere abilitato. Questa impostazione dà informazioni a kernel di configurare automaticamente l'interfaccia di rete ethernet eth0, basandosi su valori passati al kernel sulla linea di comando.

Non è necessario specificare le opzioni DHCP, BOOTP o RARP perché il bootrom Etherboot ha già effettuato la richiesta DHCP o BOOTP e avrà reso disponibile il valore dell'indirizzo IP al kernel passandolo come parametro alla linea di comando del kernel. Questo evita al kernel dei dover effettuare lui una richiesta.

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

Quando non viene utilizzato initrd, bisogna scrgliere il driver per la specifica scheda che corrisponda alla scheda di rete utilizzata . Deve essere staticamente unito al kernel (e non compilato come modulo), perché l'interfaccia di rete deve essere utilizzata prima che venga montato il filesystem principale. Questa è una differenza importante rispetto al kernel con initrd.

- File systems → /dev filesystem support

Dalla versione 2.09pre2 di LTSP, è necessario il supporto **devfs**. Questo supporto è necessario sia che si utilizzi `initrd` sia che non si utilizzi `initrd`.

- File systems → Automatically mount at boot

Quando NON si utilizza `initrd` il filesystem /dev DEVE essere montato dal kernel, durante il boot. Dite 'Y' (si) a questa opzione.

- File systems → Network file systems → NFS Client support

La workstation monta il filesystem principale tramite NFS, pertanto è necessario che il supporto per il client NFS sia attivato.

8.2.4. La creazione del kernel

Per rendere le cose più facili, una copia del file `.config` è incluso nel pacchetto `ltsp_initrd_kit`. Potete copiare questo file nella directory `/usr/src/linux-2.4.9`.

Quando avete terminato di selezionare o deselezionare i parametri del kernel, dovete passare alla sua creazione (build). Per creare il kernel dovete dare il seguenti comandi:

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

Potete anche unirli tutti assieme con:

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

Le due `&&` commerciali (&&) indicano di eseguire il primo comando e quando il primo comando è stato completato con successo eseguire il secondo comando. Quando il secondo comando è stato completato con successo viene eseguito il terzo e così via.

Il novo kernel, una volta creato, si troverà nel file `/usr/src/linux-2.4.9/arch/i386/boot/bzImage`.

8.2.5. Contrassegnare il kernel per Etherboot

Per fare in modo che Etherboot possa gestire il kernel Linux deve essere preparato per questo. Questa operazione si chiama contrassegnatura ('Tagging') del kernel. Questa operazione aggiungerà un pezzo di codice al kernel che verrà eseguito prima che il controllo sia passato al kernel. Lo strumento per eseguire questa contrassegnatura si chiama **'mknbi-linux'**.

Il pacchetto `ltsp_initrd_kit` include uno script shell chiamato **buildk** che include tutte i comandi necessari per preparare l'immagine del kernel per il boot tramite rete.

Capitolo 9. Le voci del file di configurazione lts.conf

Quando si prepara LTSP, una delle cose che bisogna tenere a mente è che bisogna considerare diverse configurazioni hardware per le workstation. Ovviamente le combinazioni di processore, scheda video e scheda video che abbiamo presente oggi non sarà quella che avremo bisogno tra tre mesi quando aggiungeremo altre workstation alla rete.

Per questo motivo è stato creato un modo per specificare la configurazione di ogni workstation. Il file di configurazione si chiama `lts.conf` ed è collocato nella directory `/opt/ltsp/i386/etc`.

Il formato del file `lts.conf` permette di definire alcune impostazioni di 'default' e alcune impostazioni per le specifiche workstation. Se tutte le vostre workstation sono identiche, potete specificare tutte le impostazioni di configurazione nella sezione '[Default]'.

9.1. Esempio del file lts.conf

Qui sotto è riportato un file `lts.conf` d'esempio:

```
[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
SCREEN_01        = startx

[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]
SCREEN_01        = shell
```

9.2. Elenco dei parametri disponibili per il file lts.conf

9.2.1. Parametri generali

Commenti

I commenti incominciano dal segno '#' e continuano fino alla fine della riga.

LTSP_BASEDIR

Questo parametro indica dove è collocato il filesystem principale di LTSP. Il valore di default (predefinito) per questo parametro è `/opt/ltsp`

SERVER

Questo parametro indica il server che è usato al posto di XDM_SERVER, TELNET_HOST, XFS_SERVER e SYSLOG_HOST, quando uno o più di questi parametri non sono specificati esplicitamente. Se avete una macchina che funziona da server per tutti questi servizi allora basta che specificate l'indirizzo del server in questo parametro e omettete gli altri parametri. Se questo parametro non è specificato viene utilizzato il valore predefinito **192.168.0.254**.

SYSLOG_HOST

Se volete mandare i messaggi di log ad una macchina diversa dal server principale (quello definito dal parametro SERVER) potete specificare il server per i log con questo parametro. Se questo parametro non è specificato verrà, come descritto sopra, utilizzato il valore specificato come 'SERVER'.

NFS_SERVER

Questo parametro specifica l'indirizzo IP del server NFS quando il filesystem /home viene montato. Se questo parametro non è specificato il valore utilizzato è quello specificato come parametro **SERVER**.

USE_NFS_SWAP

Impostate il valore di questo parametro ad **Y** se volete attivare lo swap via NFS. Il default per questo parametro è **N**

SWAPFILE_SIZE

Questo parametro controlla la dimensione del file di swap. Il valore di default è **64m**.

SWAP_SERVER

Il file di swap utilizzato può essere messo su un qualsiasi server della rete che è in grado di gestirlo. Questo parametro specifica l'indirizzo IP del server su cui mettere il file di swap. Il valore di default per questo parametro è il valore di NFS_SERVER.

NFS_SWAPDIR

Questo parametro indica la directory del server che è esportata tramite NFS. Il valore di default è /var/opt/ltsp/swapfiles. Accertatevi che questa directory sia specificata nel file /etc/exports.

TELNET_HOST

Se la workstation è configurata per avere un'interfaccia a caratteri allora questo parametro è usato per specificare a quale computer (host) collegarsi in telnet. Se questo valore non è impostato viene utilizzato il valore **SERVER** specificato sopra.

DNS_SERVER

Indica il server DNS, utilizzato per costruire il file resolv.conf.

SEARCH_DOMAIN

Utilizzato per costruire il file resolv.conf.

Parametri da SCREEN_01 a SCREEN_12

In questi parametri possono essere specificati fino a 12 screen script per ciascuna workstation. Questo dà accesso fino a 12 sessioni su ogni workstation, ognuna accessibile premendo i tasti da Ctrl-Alt-F1 a Ctrl-Alt-F12.

```
SCREEN_01 = startx
SCREEN_02 = shell
```

Attualmente i valori possibili per questo parametro sono:

- ◆ startx
- ◆ telnet
- ◆ rdesktop
- ◆ shell

Guardate nella directory `/opt/ltsp/i386/etc/screen.d` per eventuali altri script aggiuntivi, oppure scrivetene uno voi stesso e mettetelo in questa directory.

Da **MODULE_01** a **MODULE_10**

Fino a 10 moduli del kernel possono essere caricati utilizzando questi parametri di configurazione. Potete specificare l'intera riga di comando che utilizzereste con il comando `insmod` (con `insmod` escluso). Per esempio:

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

Se il valore del parametro è un pathname assoluto (cioè se è specificata anche la directory in cui trovare il file) allora viene utilizzato il comando **insmod**. Altrimenti viene utilizzato il comando **modprobe**.

RAMDISK_SIZE

Quando la workstation viene avviata, viene creato un disco RAM che viene montato come directory `/tmp`. Con questo parametro è possibile specificare la dimensione di questo filesystem. Il numero specificato deve essere in kilobyte (1024 byte). Ad esempio per creare un disco ram di 1 MB, impostate questo parametro come **RAMDISK_SIZE = 1024**

Se cambiate questo parametro, la dimensione del file ram, dovrete anche cambiare la dimensione del disco ram all'interno del kernel. Questo valore può essere compilato nel kernel, oppure se usate Etherboot o Netboot, potete indicare al kernel la dimensione del disco ram quando fate la contrassegnatura del kernel con il programma `mknbi-linux`.

Il valore di default è 1024 (1 MB)

da **RCFILE_01** a **RCFILE_10**

Degli script RC possono essere eseguiti dallo script `rc.local`. Per ottenere questo basta collocare gli script nella directory `/etc/rc.d` directory e specificare i nomi degli script in questi parametri.

SOUND

Se il pacchetto LTSP del suono è installato, dovete mettere questo parametro a **Y** e verrà eseguito lo script **rc.sound** per configurare la scheda sonora e il demone. Il valore di default per questo parametro è **N**.

9.2.2. Parametri di X-Windows

XDM_SERVER

Se volete che XDM punti ad una macchina diversa dal server principale dovete specificare la macchina da utilizzare come server X in questo parametro. Se questo parametro non è specificato verrà utilizzato il valore di 'SERVER'.

XSERVER

Questo parametro specifica quale tipo di X Server sarà utilizzato dalla workstation. Per le schede video PCI e AGP questo parametro dovrebbe non risultare necessario. Potete anche impostare questo parametro al valore **auto** per indicare che deve essere effettuata un autorivelamento.

Per le schede video ISA o se volete forzare uno specifico valore impostate questo parametro al nome del driver o del Xserver da utilizzare.

se il valore di questo parametro inizia con **XF86_**, sarà usato XFree86 3.3.6. altrimenti sarà usato X.org 6.7.0. Il valore predefinito per questo parametro è **auto**.

Da X_MODE_0 a X_MODE_2

Fino a 3 Modelines o risoluzioni possono essere configurati per ciascuna workstation. Questi parametri possono avere due differenti tipi di valore. Può essere una risoluzione o una completa modeline (linea di modo)

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

Se non è specificato nessun valore come X_MODE_x, sarà utilizzato il modelines standard e le risoluzioni 1024x768, 800x600 e 640x480.

Se uno o più valori di X_MODE_x vengono specificati, questi sostituiscono completamente le modalità preimpostate.

X_MOUSE_PROTOCOL

In questo parametro può essere specificato qualsiasi parametro valido per il valore di Pointer Protocol di X.org. Tipicamente i valori utilizzati sono "Microsoft" o "PS/2". il valore di default per questo parametro è "**PS/2**".

X_MOUSE_DEVICE

Questo parametro specifica il device node a cui è connesso il mouse. Se il mouse è di tipo seriale sarà la porta seriale, come **/dev/ttyS0** o **/dev/ttyS1**. Se il mouse è di tipo PS/2 questo valore sarà **/dev/psaux**. Il valore di default per questo parametro è **/dev/psaux**.

X_MOUSE_RESOLUTION

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

X_BUTTONS

Questo parametro indica al sistema quanti tasti abbia il mouse. In genere questo parametro ha il valore **2** o **3**. Il valore di default per questo parametro è **3**.

X_MOUSE_EMULATE3BTN

Questo parametro dice al server X di emulare un mouse a 3 tasti. Quando vengono premuti simultaneamente i due tasti (quello sinistro e quello destro) è come se fosse premuto il tasto centrale. Il valore di default per questo parametro è **N**.

X_MOUSE_BAUD

Per un mouse seriale, questo parametro specifica la velocità della connessione seriale. Il valore di default per questo parametro è **1200**.

X_COLOR_DEPTH

Questo parametro è il numero di bit utilizzato per la profondità dei colori (color depth). I possibili valori sono **8, 15, 16, 24** e **32**. Un valore di 8 bits darà 256 colori, 16 darà 65536 colori, 24 darà 16 milioni di colori e 32 bit darà 4.2 miliardi di colori! Non tutti i server X supportano tutti questi valori. Il valore di default per questo parametro è **16**.

USE_XFS

Potete scegliere se eseguire lo X Font Server (XFS) oppure se leggere i font tramite NFS. Il sistema che usa il font server dovrebbe fornire un metodo più semplice per tenere tutti i file dei font in un unico posto. Tuttavia sembra che si verificano alcuni problemi con il sistema del font server quando il numero delle workstation supera 40. Ci sono due possibili valori per questo parametro: **Y** e **N**. Il valore di default è **N**. Se volete utilizzare il font server potete utilizzare il parametro **XFS_SERVER** per specificare quale server funzionerà da X font server.

XFS_SERVER

Se utilizza lo X Font Server per fornire i font allora potete utilizzare questo parametro per specificare l'indirizzo IP del computer che farà da font server. Se questo parametro non è specificato sarà utilizzato il server di default, quello specificato dal parametro **SERVER**, come descritto sopra.

X_HORZSYNC

Questo parametro specifica il parametro di configurazione **HorizSync** di X.org. Il suo valore di default è "**31-62**".

X_VERTREFRESH

Questo parametro specifica il parametro di configurazione **VertRefresh** di X.org. Il suo valore di default è to "**55-90**".

XF86CONFIG_FILE

Se volete crearvi un vostro personale file di configurazione XF86Config potete farlo e collocarlo nella directory **/opt/lts/i386/etc** con un nome di file di vostra scelta. Mettete il nome del file come valore di questo parametro. Per esempio:

```
XF86CONFIG_FILE = XF86Config.ws004
```

9.2.3. Parametri di configurazione per il Touch screen

USE_TOUCH

Se usate un touch screen nella workstation potete abilitarlo impostando questo parametro al valore **Y**. Se abilitato il touch screen sarà configurato da altri specifici paramtri. Il valore predefinito di questo parametro è **N**.

X_TOUCH_DEVICE

Uno schero touch screen funziona come un mouse e generalmente funziona collegato ad una porta seriale. Potete specificare la porta seriale con questo parametro. Per esempio potete impostarlo a **/dev/ttyS0**. Non c'è un valore di default per questo parametro.

X_TOUCH_MINX

Questo parametro permette la calibrazione di uno schermo touch screen EloTouch. Il valore di default per questo parametro è **433**.

X_TOUCH_MAXX

Questo parametro permette la calibrazione di uno schermo touch screen EloTouch. Il valore di default per questo parametro è **3588**.

X_TOUCH_MINY

Questo parametro permette la calibrazione di uno schermo touch screen EloTouch. Il valore di default per questo parametro è **569**.

X_TOUCH_MAXY

Questo parametro permette la calibrazione di uno schermo touch screen EloTouch. Il valore di default per questo parametro è **3526**.

X_TOUCH_UNDELAY

Questo parametro permette la calibrazione di uno schermo touch screen EloTouch. Il valore di default per questo parametro è **10**.

X_TOUCH_RPTDELAY

Questo parametro permette la calibrazione di uno schermo touch screen EloTouch. Il valore di default per questo parametro è **10**.

9.2.4. Parametri per le applicazioni locali.***LOCAL_APPS***

Se volete abilitare l'esecuzione di applicazioni locali sulla workstation impostate questo parametro ad **Y**. Per abilitare le applicazioni locali occorrerà configurare molti altri punti. Consultate la sezione 'Local Apps' (Applicazioni Locali) del manuale LTSP per maggiori informazioni. Il valore di default per questo parametro è **N**.

NIS_DOMAIN

Se avete impostato LOCAL_APPS a Y, allora dovete avere un server NIS nella rete. Il parametro NIS_DOMAIN serve a specificare il nome del NIS domain Deve coincidere con il dome che è stato definito nel server NIS. Questo nome NON è la stessa cosa del nome di domino internet. Il valore di default per questo parametro è **ltsp**.

NIS_SERVER

Impostate questo parametro al valore del server NIS se non volete che sia effettuata una chiamata broadcast per cercare il server NIS.

9.2.5. Parametri della tastiera (Keyboard)

Tutti i file per il supporto delle tastiere sono copiati nella directory /opt/ltsp/i386 e nelle sue sottodirectory, pertanto configurare una tastiera internazionale (non statunitense) è semplicemente una questione di configurazione dei X.org. Ci sono molti parametri per questa cosa.

I valori che devono assumere questi parametri sono i quelli descritti nella documentazione di X.org. Tutti i parametri validi per X.org sono parametri validi per questi parametri.

Vorremmo aggiungere alla documentazione la descrizione dei valori necessari per configurare ciascun tipo di tastiera internazionale. Se avete lavorato per configurare una tastiera internazionale e ci siete riusciti, le indicazioni e i commenti su come riuscirci saranno molto graditi dal ltsp core group.

XkbTypes

Il valore di default per questa parola è **'default'**.

XkbCompat

Il valore di default per questa parola è **'default'**.

XkbSymbols

Il valore di default per questo è **'us(pc101)'**.

XkbModel

Il valore di default per questo è **'pc101'**.

XkbLayout

Il valore di default per questa parola è **'us'**.

9.2.6. Parametri di configurazione delle stampanti

Fino a 3 stampanti possono essere connesse ad una workstation senza dischi. Si può configurare una combinazione di stampanti collegate a porte seriali e parallele, tramite i seguenti parametri ne file **lts.conf**:

PRINTER_0_DEVICE

Nome del device della prima stampante. Sono utilizzabili nomi come **/dev/lp0**, **/dev/ttyS0** o **/dev/ttyS1**.

PRINTER_0_TYPE

Specifica il tipo della stampante. Valide possibilità sono: **'P'** per le parallele o and **'S'** per le seriali.

PRINTER_0_PORT

Specifica il numero di porta della connessione TCP/IP da usare. Il valore di default è **'9100'**

PRINTER_0_SPEED

Se la stampante è seriale, questo parametro indica la velocità di connessione. Il valore di default è **'9600'**.

PRINTER_0_FLOWCTRL

Per una stampante seriale specifica il flow control Le possibile scelte sono: **'S'** per controllo Software (XON/XOFF) del flusso di dati, **'H'** per il controllo Hardware (CTS/RTS). Se non viene specificato il valore di default utilizzato è **'S'**.

PRINTER_0_PARITY

Se la stampante è seriale specifica la parità. I valori possibili sono: **'E'**–Even, (parità pari), **'O'**–Odd (parità dispari) o **'N'**–None (nessuna parità). Se non specificato viene utilizzato di default il valore **'N'**.

PRINTER_0_DATABITS

Per le stampanti seriali specifica il numero dei bit dei dati. Le possibili scelte sono: **'5'**, **'6'**, **'7'** e **'8'**. Se non specificato sarà utilizzato il valore **'8'**.

PRINTER_1_DEVICE

Nome del device della seconda stampante

PRINTER_1_TYPE

Tipo del device della seconda stampante

PRINTER_1_PORT

Porta del device TCP/IP per la seconda stampante

PRINTER_1_SPEED

baud rate per la seconda stampante (se seriale)

PRINTER_1_FLOWCTRL

Tipo di controllo di flusso (flow control) per la seconda stampante (se seriale)

PRINTER_1_PARITY

Parità della connessione per la seconda stampante (se seriale)

PRINTER_1_DATABITS

Numero di bit dei dati per la seconda stampante. (se seriale)

PRINTER_2_DEVICE

Nome del device della terza stampante

PRINTER_2_TYPE

Tipo del device della terza stampante

PRINTER_2_PORT

Porta del device TCP/IP per la terza stampante

PRINTER_2_SPEED

baud rate per la terza stampante (se seriale)

PRINTER_2_FLOWCTRL

Tipo di controllo di flusso (flow control) per la terza stampante (se seriale)

PRINTER_2_PARITY

Parità della connessione per la terza stampante (se seriale)

PRINTER_2_DATABITS

Numero di bit dei dati per la terza stampante. (se seriale)

Capitolo 10. Applicazioni locali

In un ambiente LTSP potete scegliere se eseguire le applicazioni localmente sulla workstation o in remoto sul server.

Il metodo di gran lunga più semplice da configurare in un ambiente LTSP è quello che fa eseguire le applicazioni sul server. In questo modo le applicazioni sono eseguite sul server usando la CPU e la memoria del server, mentre l'output è mostrato sullo schermo della workstation e vengono usate la tastiera e il mouse della workstation.

Questa è una capacità fondamentale di X Windows. La workstation funziona proprio come un normale terminale X Windows.

Per far sì che un utente possa eseguire le applicazioni sulla workstation, la workstation deve conoscere alcune informazioni sull'utente. Le informazioni necessarie sono le seguenti:

- User id (identificativo dell'utente)
- Gruppo primario a cui appartiene l'utente.
- Home directory dell'utente.

LTSP si basa sul Network Information Service – NIS, (precedentemente chiamato *Yellow Pages*) per rendere disponibile alla workstation le informazioni sull'utente e sul gruppo.

10.1. Vantaggi di far eseguire le applicazioni sulla workstation

Ci sono alcuni vantaggi a far eseguire le applicazioni sulla workstation.

- Si riduce il carico sul server. Nel caso di una vasta rete e nel caso si utilizzano applicazioni che fanno un uso intenso della memoria, come Netscape, far eseguire le applicazioni sulla workstation può fornire migliori prestazioni, fintanto che la workstation è sufficientemente potente da riuscire a farlo.
- Le applicazioni che entrano in runaway (continuano l'esecuzione senza terminare ed usando molta CPU e molta memoria) non disturbano gli altri utenti.
- Il supporto per il suono è molto più facile da configurare quando l'applicazione che produce il suono è in esecuzione sulla workstation.

10.2. Svantaggi di far eseguire le applicazioni sulla workstation

La configurazione per far sì che le applicazioni siano eseguite localmente presenta però alcuni svantaggi.

- In questa modalità la workstation deve fare un lavoro maggiore. C'è bisogno di più memoria RAM e di una CPU più potente. Una workstation con 64 MB di RAM; è un buon punto di partenza.
- NIS – Per far eseguire le applicazioni localmente è necessario, per prima cosa che la workstation identifichi l'utente. Come metodo di autenticazione degli utenti nella rete è stato scelto NIS.
- C'è bisogno di esportare altre directory dalla workstation da montare con NFS.
- L'avvio delle applicazioni sarà più lento. Infatti l'applicazione deve essere letta tramite NFS, questo causa una maggiore attività di rete. Inoltre, siccome ogni copia del programma è in esecuzione sulla sua CPU, non si avrà il vantaggio dato dalla capacità di Linux/Unix di condividere il segmento di codice tra istanze multiple del programma, cosa che renderebbe più veloce l'avvio dei programmi alla seconda invocazione e alle successive.

10.3. Configurazione del server per l'esecuzione locale delle applicazioni

10.3.1. Opzioni del file `lts.conf`

Alcune opzioni devono essere configurate nel file `lts.conf`:

LOCAL_APPS

Questa opzione deve essere impostata a **Y**. Questa opzione così impostata causa l'avvenire delle seguenti cose durante il processo di avvio della workstation:

1. La directory `/home` sul server viene montata tramite NFS.
2. Il file `/var/yp/nicknames` viene creato sulla workstation.
3. Il **portmapper** viene avviato sulla workstation.
4. **xinetd** viene avviato sulla workstation.
5. Il file `/etc/yp.conf` viene creata sulla workstation.
6. Il comando **domainname** viene eseguito con il valore di **NIS_DOMAIN** preso dal dal file `lts.conf`.
7. Viene mandato in esecuzione **ypbind** sulla workstation.

NIS_DOMAIN

Con il sistema NIS, tutti i nodi della rete che vogliono essere associati con uno specifico server NIS devono appartenere allo stesso dominio NIS. (Questo non ha nulla a che fare con i domini del DNS). Dovete utilizzare **NIS_DOMAIN** per specificare il nome del dominio NIS a cui appartiene la workstation.

NIS_SERVER

Il client NIS cercherà di collegarsi al server NIS specificato se in questo parametro viene specificato l'indirizzo IP del server NIS. Altrimenti se non è specificato invierà un messaggio di tipo broadcast sulla rete in cerca del server.

10.3.2. Network Information Service – NIS

Il sistema NIS è un servizio di tipo Client/Server. Sul server c'è un demone che accetta le richieste dal client (workstation). Il demone sul server si chiama **ypserv**.

Sulla workstation c'è un programma chiamato **ypbind**. Quando la workstation ha bisogno di cercare alcune informazioni sull'utente, come verificare una password o trovare quale sia la home directory dell'utente, userà **ypbind** per stabilire una connessione con il server.

Se state già utilizzando un server NIS nel vostro ambiente di rete non c'è bisogno di configurare il server LTSP per eseguire **ypserv**. Basta che configurate il parametro **NIS_DOMAINNAME** e **NIS_SERVER** nel file `lts.conf` per essere compatibili con l'ambiente NIS.

Se NON avete di già un server NIS in funzione sulla vostra rete dovete configurare il vostro server perché esegua **ypserv**.

Per delle informazioni complete su come installare e configurare un server NIS, consultate il documento HOWTO disponibile sul sito LDP (Linux Documentation Project) chiamato *The Linux NIS(YP)/NYS/NIS+ HOWTO*. (Dovrebbe essere disponibile anche una traduzione in italiano) Consultate la lista delle fonti d'informazione alla fine di questo documento.

10.4. Configurazione delle Applicazioni

Installare un'applicazione perché sia eseguita sulla workstation richiede che tutti i componenti dell'applicazione siano messi in uno spazio accessibile dalla workstation.

Nelle vecchie versioni di LTSP (la 2.08 e le precedenti) un gran numero di directory erano esportate dal server e montate sulla workstation. Tra queste c'erano delle directory quali: `/bin`, `/usr/bin`, `/lib` e `/usr`.

Il problema di questo schema è che funziona solo se la workstation e il server hanno la stessa architettura. In effetti perfino differenze minori possono esserci dei problemi, come il caso in cui sul server ci sia un Pentium II (i686) e sulla workstation un classico Pentium, perché il server probabilmente avrà le librerie per l'architettura i686 e non le librerie per i386, i486 o i586.

Pertanto il modo più pulito di trattare la cosa è di aver un albero completo con i file binari (eseguibili) e le librerie che sono richieste dalla workstation, indipendentemente dai binari e dalle librerie del server.

L'installazione e la configurazione di un applicazione per essere eseguita localmente richiede che tutte le cose richieste nell'albero. Uno dei pacchetti disponibili LTSP per il download è il pacchetto Netscape locale. Questo pacchetto installa molti file nella directory `/opt/ltsp/i386/usr/local/netscape`. Le cose come le classi java, i file di help, i file binari eseguibili e gli script sono messe in questa directory.

Netscape non richiede alcuna libreria addizionale di sistema, pertanto non c'è bisogno di aggiungere nulla alla directory `/opt/ltsp/i386/lib`. D'altra parte molte applicazioni richiedono delle librerie addizionali.

A questo punto c'è da chiedersi come scoprire quali siano le librerie necessarie. Per questo torna utile utilizzare il comando **ldd**.

Prendiamo in considerazione il caso in cui volete installare una applicazione per essere eseguita localmente. Come esempio consideriamo **gaim**. **gaim** è un AOL Instant Messenger client, che permette di comunicare con le persone nei forum di AOL.

La prima cosa da fare è quella di trovare il file eseguibile **gaim**. Sui sistemi Red Hat 7.2 si trova nella directory `/usr/bin`.

Una volta che avete trovato il file binario eseguibile **gaim** potete eseguire su di lui il comando **ldd**:

```
[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_implicit.so.1 => /usr/lib/libgdk_implicit.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)

```

La lista qui sopra mostra tutte le librerie con cui il programma **gaim** è stato dinamicamente collegato (linked).

Molti programmi che utilizzano shared libs (librerie condivise) si basano sul dynamic loader (caricatore dinamico) **ld-linux** per rintracciare le librerie e caricarle. Tuttavia alcuni programmi caricano le librerie direttamente a mano con la chiamata di funzione **dlopen()**. Nel caso di queste applicazioni il comando **ldd** non mostrerà queste librerie. In questo caso il programma **strace** può essere utilizzato per prendere traccia di cosa avviene durante l'esecuzione del programma e sarete in grado di vedere le chiamate **dlopen()** con il nome delle librerie elencate negli argomenti.

Si può così fare una lista delle librerie richieste e copiarle nell'apposito posto nella gerarchia dell'albero `/opt/ltsp/i386`.

10.5. Lanciare le applicazioni locali

In X windows i programmi tipicamente sono eseguiti relativamente al luogo dove è in esecuzione il window manager. Pertanto se il window manager è in esecuzione sul server, e il suo output è mostrato sulla workstation, allora ogni programma che venga lanciato sarà eseguito sul server, inviando l'output alla workstation.

Qui serve un trucco per far sì che il server dica alla workstation di lanciare ed eseguire il programma. Questo è in genere fatto con il comando **rsh**.

Qui sotto è riportato un esempio di come eseguire il programma **gaim** sulla workstation:

```

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

```

L'esempio precedente può essere inserito in una finestra **xterm** oppure possono essere messi in un file script e lanciato da un'icona sul desktop.

Si può avviare Netscape localmente in modo simile. In questo caso, però, è necessario impostare un'ulteriore variabile d'ambiente, prima di far eseguire il programma:

```

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

```

Capitolo 11. Esempi di configurazioni

Quasi tutte le caratteristiche della workstation possono essere configurate nel file `lts.conf`, che, di solito, si trova nella directory `/opt/ltsp/i386/etc`.

11.1. Mouse seriale

Queste linee, parte del file `lts.conf`, mostrano come fare la configurazione per un mouse seriale standard a 2 tasti:

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

11.2. Mouse PS/2 con rotella

Queste linee, parte del file `lts.conf`, mostrano come fare la configurazione per un mouse Intellimouse:

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

11.3. Stampante USB su una ThinkNic

Una workstation ThinkNIC ha una porta USB port, che può essere usata per collegarci una stampante locale. Queste linee, parte del file `lts.conf`, mostrano come fare la configurazione per una stampante locale collegata tramite la porta USB:

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

11.4. Come forzare la workstation a caricare XFree86 3.3.6 Xserver

Di default, viene usato X.org 6.7.0 sulla workstation. Se volete forzare la workstation ad utilizzare il più vecchio XFree86 3.3.6 Xserver, dovete innanzitutto caricare l'opportuno pacchetto 3.3.6 Xserver. Quindi dovete aggiungere questa voce al file `lts.conf`. Questo esempio mostra come specificare l'utilizzo del server X **SVGA**:

```
XSERVER = XF86_SVGA
```

Capitolo 12. Altre fonti di informazioni

12.1. Informazioni Online

1. La pagina principale del sito web di LTSP e

www.LTSP.org

(è in inglese, ma sul sito ci sono anche delle pagine italiano)

2. Diskless-Nodes HOW-TO document for Linux

en.tldp.org/HOWTO/Diskless-HOWTO.html

(in inglese)

3. Etherboot Home Page

etherboot.sourceforge.net

(in inglese)

4. The Rom-O-Matic site

www.Rom-O-Matic.net

(in inglese)

5. X.org Mouse Support

www.xfree86.org/current/mouse.html

(in inglese)

6. XFree86-Video-Timings-HOWTO

www.tldp.org/HOWTO/XFree86-Video-Timings-HOWTO/index.html

(in inglese) La traduzione in italiano è disponibile presso

it.tldp.org/HOWTO/XFree86-Video-Timings-HOWTO.html

7. The Linux NIS(YP)/NYS/NIS+ HOWTO

www.tldp.org/HOWTO/NIS-HOWTO/index.html

(in inglese) La traduzione in italiano è disponibile presso it.tldp.org/HOWTO/NIS-HOWTO/index.html

12.2. Pubblicazioni stampate (libri, riviste ecc) (in inglese)

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)