

LTSP – Linux Terminal Server Project – v4.1

James McQuillan

<jam@LTSP.org>

Alexandre Cavalcante Alencar

Lead Brazilian Translator

<alexandre@debian-ce.org>

Grace Prado

Brazilian Translator

<gracempb@yahoo.com.br>

Erlon Sousa Pinheiro

Brazilian Translator

<erlon@debian-ce.org>

Thiago Bortoletto Vaz

Brazilian Translator

<tiago@debian-ba.org>

Copyright © 2005 James A. McQuillan

Histórico de Revisões

Revisão 4.1.3

2004-06-20

Revisado por: jam

Revisão 4.1.3-0a-ptbr

2005-11-12

Revisado por: aac

O GNU/Linux é uma grande plataforma para implementar thin clients. O propósito primário deste documento é mostrar como implantar thin clients usando o LTSP. Mas, este documento aborda muitos problemas com estações sem disco em geral.

<!-- TOCTITLE="Tabela de Conteúdo" -->

| | |
|--|-----------|
| <u>Introdução</u> | 1 |
| <u>1. Disclaimer</u> | 1 |
| <u>2. Direitos Autorais e Licença</u> | 2 |
| <u>Capítulo 1. Teoria da operação</u> | 3 |
| <u>1.1. As etapas que a estação de trabalho passará</u> | 3 |
| <u>1.2. Carregando o kernel na memória</u> | 6 |
| <u>Capítulo 2. Instalando o LTSP no servidor</u> | 8 |
| <u>2.1. Instalação dos utilitários LTSP</u> | 8 |
| <u>2.2. Instalando os pacotes dos clientes LTSP</u> | 9 |
| <u>2.3. Configurando os serviços necessários ao LTSP</u> | 13 |
| <u>2.4. Configurações específicas da estação</u> | 15 |
| <u>2.5. Exibindo a configuração atual</u> | 18 |
| <u>Capítulo 3. Configurando a estação de trabalho</u> | 20 |
| <u>3.1. Carga com PXE</u> | 20 |
| <u>3.2. Carga com Etherboot</u> | 21 |
| <u>Capítulo 4. Ligando a estação</u> | 24 |
| <u>Capítulo 5. Impressão</u> | 25 |
| <u>5.1. Configurações no lado cliente</u> | 25 |
| <u>5.2. Configurações no lado do servidor</u> | 25 |
| <u>Capítulo 6. Scripts de tela</u> | 28 |
| <u>Capítulo 7. Solução de Problemas</u> | 30 |
| <u>7.1. Solucionando problemas em disquetes com imagens Etherboot</u> | 30 |
| <u>7.2. Solucionando problemas no DHCP</u> | 30 |
| <u>7.3. Solucionando problemas no TFTP</u> | 33 |
| <u>7.4. Solucionando problemas do sistema de arquivos raiz NFS</u> | 34 |
| <u>7.5. Solucionando problemas no Xserver</u> | 36 |
| <u>7.6. Solucionando problemas no gerenciador de telas</u> | 37 |
| <u>Capítulo 8. Kernels</u> | 40 |
| <u>8.1. Kernels padrões fornecidos pelo LTSP</u> | 40 |
| <u>8.2. Construa seu próprio kernel</u> | 40 |
| <u>Capítulo 9. Entradas em lts.conf</u> | 45 |
| <u>9.1. Arquivo exemplo lts.conf</u> | 45 |
| <u>9.2. Parâmetros disponíveis no arquivo lts.conf</u> | 45 |
| <u>Capítulo 10. Aplicações Locais</u> | 53 |
| <u>10.1. Benefícios da execução de aplicações localmente</u> | 53 |
| <u>10.2. Questões com a configuração do suporte de aplicações locais</u> | 53 |
| <u>10.3. Configuração do servidor para aplicações locais</u> | 54 |
| <u>10.4. Configuração de Aplicações</u> | 55 |

<!-- TOCTITLE="Tabela de Conteúdo" -->

Capítulo 10. Aplicações Locais

10.5. Executando aplicações locais.....56

Capítulo 11. Exemplos de Configuração.....57

11.1. Mouse Serial.....57

11.2. Mouse PS/2 com roda.....57

11.3. Impressora USB em um ThinkNic.....57

11.4. Forçando a estação a carregar o Xserver XFree86 3.3.6.....57

Capítulo 12. Outras fontes de informação.....58

12.1. Referências Online.....58

12.2. Publicações Impressas.....58

Introdução

O LTSP fornece um meio simples de utilizar estações de baixo custo como terminais gráficos ou caracteres em um servidor GNU/Linux.

Em uma configuração tradicional de escritório há PC's relativamente potentes baseados em processadores Intel em cada mesa de trabalho. Cada um com muitos gigabytes de espaço em disco. Os usuários armazenam seus dados no disco local, e cópias de segurança são raramente (quando são) realizadas.

Será que realmente faz sentido ter um computador completo em cada mesa?

Dizemos que não.

Por sorte, há outro meio. Utilizando o LTSP, você pode ter PC's baratos, remover seu disco rígido, disquete, CD-ROM, e adicionar uma placa de rede com suporte a boot por rede. Muitas placas de rede possuem encaixes, apenas esperando por um chip de boot ser instalado.

Durante a fase de boot, a estação sem disco obtém suas informações IP e o kernel do servidor, e então monta o sistema de arquivos raiz via NFS.

A estação de trabalho pode ser configurada em um dos três modos:

- **Interface Gráfica X Window System**

Usando o X Windows, a estação de trabalho pode ser usada para acessar qualquer aplicação no servidor, ou em outros servidores na rede.

- **Interface Caractere Sessões Telnet**

A estações de trabalho pode obter múltiplas sessões telnet no servidor. Cada sessão telnet estará em uma janela virtual separada. Pressionando Alt-F1 até Alt-F9 irá alternar entre as sessões.

- **Aviso Shell**

A estação de trabalho pode ser configurada para cair diretamente em um shell bash no console. Isto é muito útil quando estiver depurando problemas com o X Windows ou NFS.

Uma coisa realmente interessante é que você pode ter várias estações de trabalho conectadas a um único servidor GNU/Linux. Quantas estações de trabalho? Bem, isto depende da capacidade do servidor e das aplicações que serão usadas.

Não é comum ter 50 estações de trabalho, todas utilizando o Mozilla e OpenOffice.org em uma Dual P4-2.4 com 4GB de ram. Nós sabemos que funciona. De fato, a carga média estará raramente acima de 1.0.

1. Disclaimer

Nem o autor ou distribuidores, ou qualquer outro contribuidor deste documento é responsável de qualquer forma por danos físicos, financeiros, morais ou qualquer outro tipo ocorridos por seguir sugestões deste texto.

2. Direitos Autorais e Licença

This document is copyright 2004 by James McQuillan, and is released under the terms of the GNU Free Documentation License, which is hereby incorporated by reference.

Capítulo 1. Teoria da operação

Iniciar uma estação sem disco envolve alguns passos. Entender o que está acontecendo ao longo do processo irá tornar muito mais fácil a solução de problemas se eles chegarem a acontecer.

Há quatro serviços básicos que são necessários para iniciar uma estação LTSP. São eles:

- DHCP
- TFTP
- NFS
- XDMCP

O LTSP é muito flexível. Cada um dos serviços relacionados acima podem ser oferecidos pelo mesmo servidor, ou de diferentes servidores. Por exemplo, iremos descrever uma configuração simples, na qual consiste de um único servidor, fornecendo todos os serviços acima.

1.1. As etapas que a estação de trabalho passará

1. Carregar o kernel Linux na memória da estação de trabalho. Isto pode ser feito de diferentes maneiras, incluindo:
 - a. Bootrom (Etherboot,PXE,MBA,Netboot)
 - b. Disquete
 - c. Disco rígido
 - d. CD-ROM
 - e. Dispositivos de armazenamento USBCada um dos métodos acima será explicado neste capítulo.
2. Uma vez que o kernel tenha sido carregado na memória, ele iniciará sua execução.
3. O kernel irá iniciar todo o sistema básico e todos os periféricos que reconhecer.
4. É aqui que a diversão começa realmente. Durante o processo de carga do kernel, a imagem de um ramdisk também será carregada em memória. Um argumento na linha de comando do kernel **root=/dev/ram0** o informa para montar a imagem como o diretório raiz.
5. Normalmente, quando o kernel finaliza sua carga, ele carregará o programa **init**. Mas, neste caso, instruímos o kernel para carregar um pequeno shell script em vez do padrão. Fazemos isto passando **init=/linuxrc** na linha de comando do kernel.
6. O script **/linuxrc** inicia por analisar o barramento PCI, procurando por uma placa de rede. Para cada dispositivo PCI que ele encontrar, será feita uma busca no arquivo `/etc/niclist`, para ver se há uma coincidência. Uma vez que coincidência é encontrada, o nome do módulo da placa de rede é retornado, e o modulo do kernel é carregado. Para as placas ISA, o módulo precisa ser especificado na linha de comando do kernel, juntamente com qualquer endereço IRQ, ou parâmetros de endereço que possam ser requeridos.
7. Um pequeno cliente DHCP chamado **dhclient** irá então, ser executado para realizar outra consulta ao servidor DHCP. É preciso fazer esta consulta no espaço de usuário, porque precisamos de mais

informações do que foram obtidas pela consulta ao DHCP pelo bootrom.

8. Quando o **dhclient** receber uma resposta do servidor, ele irá executar o arquivo **/etc/dhclient-script**, o qual utilizará as informações recebidas, e configurará a interface eth0.
9. Até este ponto, o sistema de arquivos raiz era um disk. Agora, o script **/linuxrc** irá montar um novo sistema de arquivos raiz via NFS. O diretório exportado do servidor é tipicamente **/opt/ltsp/i386**. Ele não pode simplesmente montar o novo sistema de arquivos como **/**. Ele primeiro precisa montar este em **/mnt**. Então, ele chama **pivot_root**. A **pivot_root** irá trocar o sistema de arquivos raiz atual por um novo sistema de arquivos. Quando ele completar, o sistema de arquivos NFS estará montado como **/** e o sistema de arquivos raiz anterior estará montado em **/oldroot**.
10. Uma vez que a montagem e troca do novo sistema de arquivos raiz estiver completa, nós terminamos com o shell script **/linuxrc** e precisamos chamar o programa **/sbin/init** real.
11. O **Init** irá ler o arquivo **/etc/inittab** e começará a configurar o ambiente da estação.
12. Um dos primeiros itens é o arquivo **inittab**. É o comando **rc.sysinit** que será executado enquanto a estação estiver no estado **'sysinit'**.
13. O script **rc.sysinit** irá criar um disco em memória de 1MB para conter todas as coisas que precisam ser gravadas ou modificadas de alguma forma.
14. O ramdisk será montado como o diretório **/tmp**. Qualquer arquivo que precisa ser gravado, existirá atualmente no diretório **/tmp** haverá links simbólicos apontando para estes arquivos.
15. O sistema de arquivo **/proc** está montado.
16. O arquivo **lts.conf** será processado, e todos os parâmetros neste arquivo que pertencem a esta estação, serão configurados nas variáveis de ambiente para uso do script **rc.sysinit**.
17. Se a estação estiver configurada para usar SWAP via NFS, o diretório **/var/opt/ltsp/swapfiles** será montado como **/tmp/swapfiles**. Então, se ainda não existir um arquivo de swap para esta estação, ele será criado automaticamente. O tamanho do arquivo de swap é configurado no arquivo **lts.conf**.

O arquivo de swap será então ativado, através do comando **swapon**.
18. A interface de rede **loopback** é configurada. Esta é a interface de rede que usa o endereço IP **127.0.0.1**.
19. Se as aplicações locais estiverem ativadas, então o diretório **/home** será montado, de forma que as aplicações possam acessar o diretório pessoal dos usuários.
20. Diversos diretórios são criados. O sistema de arquivos **/tmp** para manter alguns dos arquivos transitórios que são necessários durante a execução do sistema. Diretórios como:

a. **/tmp/compiled**

b. **/tmp/var**

c. **/tmp/var/run**

- d. `/tmp/var/log`
- e. `/tmp/var/lock`
- f. `/tmp/var/lock/subsys`

serão todos criados.

21. O arquivo `/tmp/syslog.conf` será criado. Este arquivo conterá informações com a localização do daemon **syslogd** o qual irá armazenar as informações enviadas pela rede. O computador com o serviço `syslog` é especificado no arquivo `lts.conf`. Há um link simbólico chamado `/etc/syslog.conf` que aponta para o arquivo `/tmp/syslog.conf`.
22. O daemon **syslogd** é iniciado, usando o arquivo de configurações criado no passo anterior.
23. Uma vez que o script `rc.sysinit` estiver concluído, o controle retorna para o programa `/sbin/init`, o qual irá mudar o runlevel de **sysinit** para **5**.

Isto fará com que todas as entradas no arquivo `/etc/inittab` sejam executadas.
24. Por padrão, haverá uma entrada no `inittab` para executar o script `/etc/screen_session` no `tty1`, `tty2` e `tty3`. Isto significa que você pode executar 3 sessões ao mesmo tempo, e o tipo da sessão é controlado pelas variáveis **SCREEN_01**, **SCREEN_02** e **SCREEN_03** no arquivo `lts.conf`.

Mais entradas podem ser configuradas no `inittab` para mais sessões, se desejado.
25. Se **SCREEN_01** é configurada para o valor **startx**, então, o script `/etc/screen.d/startx` será executado, o qual irá lançar o X Windows System, dando-lhe uma interface gráfica de usuário.

Há um parâmetro chamado **XSERVER** no arquivo `lts.conf`, Se este parâmetro estiver faltando, ou configurado para "**auto**", então uma detecção automática da placa de vídeo será tentada. Se a placa for PCI ou AGP, então serão obtidos o PCI Vendor e o Device id, e feita uma busca no arquivo `/etc/vidlist`.

Se a placa for suportada pelo Xorg 6.7, a rotina `pci_scan` irá retornar o nome do módulo do driver. Se esta for suportada apenas pelo XFree86 3.3.6, o `pci_scan` irá retornar o nome do servidor X a ser usado. O script `startx` pode informar a diferença por que os nomes dos módulos do X Server 3.3.6 começam com 'XF86_', enquanto na nova versão do X Server Xorg, os nomes dos módulos são tipicamente em minúsculas, como *ati* ou *trident*.
26. Se o Xorg for usado, então o script `/etc/build_x4_cfg` será chamado para criar um arquivo `XF86Config` file. Se o XFree86 3.3.6 for usado, então o script `/etc/build_x3_cfg` será chamado para criar o arquivo `XF86Config`. Estes arquivos serão postos no diretório `/tmp`. Os quais, se você estiver lembrado, é um ramdisk, visto apenas pela estação.

O arquivo `XF86Config` será criado, baseado nas entradas do arquivo `/etc/lts.conf`.
27. Uma vez que o arquivo `XF86Config` foi criado, então o script **startx** irá carregar o X Server com aquele novo arquivo de configuração.

28. O X Server irá enviar uma consulta *XDMCP* para o servidor LTSP, o qual irá oferecer um diálogo de login.
29. Neste ponto, o usuário pode logar. Ele irá obter uma sessão no servidor.

Em um primeiro momento, isto confunde muitas pessoas. Elas estão à estação, mas elas estão executando uma sessão no servidor. Todos os comandos que elas executam, serão executados no servidor, mas a saída será mostrada na estação.

1.2. Carregando o kernel na memória

Pôr o kernel Linux dentro das memórias das estações pode ser feito de várias maneiras.

- Boot ROM
- Mídia local

1.2.1. Boot ROM

- Etherboot

O Etherboot é um projeto de bootrom open-source muito popular. Ele contém drivers para muitas placas de rede comuns, e funciona muito bem com o LTSP.

O kernel Linux kernels precisa ser 'rotulado' com o **mknbi-linux**, o qual irá preparar o kernel para boot via rede, através da prefixação do kernel com alguns códigos adicionais e anexando uma *initrd* ao final do kernel.

O kernels que é fornecido com o LTSP já está 'rotulado' e pronto para uso com o Etherboot.

O Etherboot também pode ser gravado em disquete, funcionando perfeitamente para testes.

- PXE

Parte da especificação 'Wired for Management' da década de 1990 incluiu a especificação de uma tecnologia de bootrom conhecida como *Pre-boot Execution Environment* comumente abreviada como **PXE**.

Uma bootrom PXE bootrom pode carregar até no máximo um arquivo de 32 Kilobytes. Um kernel Linux é um pouco maior que isto. Por isso, configuramos um carregador de boot de segundo estágio chamado **pxelinux**. O **pxelinux** é pequeno o bastante para ser carregado, e ele sabe como carregar arquivos grandes, como o kernel Linux.

- MBA

O Managed Boot Agent (MBA) é uma bootrom de uma companhia chamada **emBoot**. A emBoot foi utilizada como a divisão Lanworks da 3Com. O MBA é na realidade, quatro bootroms em uma. Ele irá tratar PXE, TCP/IP, RPL e Netware.

A implementação PXE do MBA funciona muito bem. Você pode usá-la com o **pxelinux** para carregar o kernel Linux.

O método *TCP/IP* pode ser usado, mas antes, o kernel precisa ser preparado com um utilitário chamado **imggen**.

- Netboot

O Netboot, assim como Etherboot, é um projeto de software livre que fornece imagens ROM livres para boot. A diferença é que ele é um 'invólucro' para o driver NDIS ou drivers de pacotes distribuídos com as placas de rede.

1.2.2. Mídia Local

- Disquetes

Há dois modos de iniciar uma estação LTSP com disquete. Um modo é carregar o Etherboot no setor de boot do disquete. Então, ele funcionará como uma bootrom. O código de boot será executado, a placa de rede será iniciada, e o kernel será carregado do servidor de rede.

Você poderá também gravar o kernel e a initrd no disquete e iniciar daquela maneira. Entretanto, atualmente, é mais rápido carregar o kernel via rede.

- Disco rígido

O disco rígido pode ser usado com o LILO ou GRUB, para carregar o kernel Linux e a initrd ou você pode carregar uma imagem bootrom Etherboot do disco rígido, e esta atuará como uma bootrom.

- CD-ROM

Um CD-ROM 'bootavel' pode ser carregado com o kernel Linux ou uma imagem Etherboot.

- Dispositivos de armazenamento USB

Assim como o CD-ROM, disquete e disco rígido, você pode usar um dispositivo de armazenamento USB tanto para iniciar um módulo Etherboot, como um kernel Linux e uma imagem initrd.

Capítulo 2. Instalando o LTSP no servidor

É melhor pensar no LTSP como uma distribuição completa de Linux. É uma distribuição que funciona no topo de uma distribuição host. A distribuição host pode ser qualquer distribuição de Linux que você quiser. De fato, não há nenhuma exigência real que o host esteja rodando Linux. A única exigência é que o sistema host seja capaz de servidor NFS (Network File System). A maioria dos sistemas Unix pode fazê-lo. De fato, mesmo algumas versões de Microsoft Windows podem ser configuradas para trabalhar como um servidor LTSP.

Há três fases para construir um servidor LTSP.

- Instalação dos utilitários LTSP
- Instalando os pacotes dos clientes LTSP
- Configurando os serviços necessários ao LTSP

2.1. Instalação dos utilitários LTSP

A partir da versão 4.1, o LTSP possui um pacote de utilitários para a instalação e gerenciamento de pacotes dos clientes LTSP (O software que é executado nos clientes), e para configurar os serviços no servidor LTSP.

O utilitário de administração é chamado de **ltspadmin** e a ferramenta de configuração de **ltspcfg**. Ambas as ferramentas são partes do pacote **ltsp-utils**.

O pacote **ltsp-utils** está disponível nos formatos **RPM** e **TGZ**. Escolha o formato que preferir instalar, e siga as instruções apropriadas.

2.1.1. Instalando o pacote RPM

Faça o download da última versão do pacote RPM **ltsp-utils**, e instale-o utilizando o seguinte comando:

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

O comando acima irá instalar os utilitários LTSP no servidor.

2.1.2. Instalando o pacote TGZ

Faça o download da última versão do pacote RPM **ltsp-utils**, e instale-o utilizando o seguinte comando:

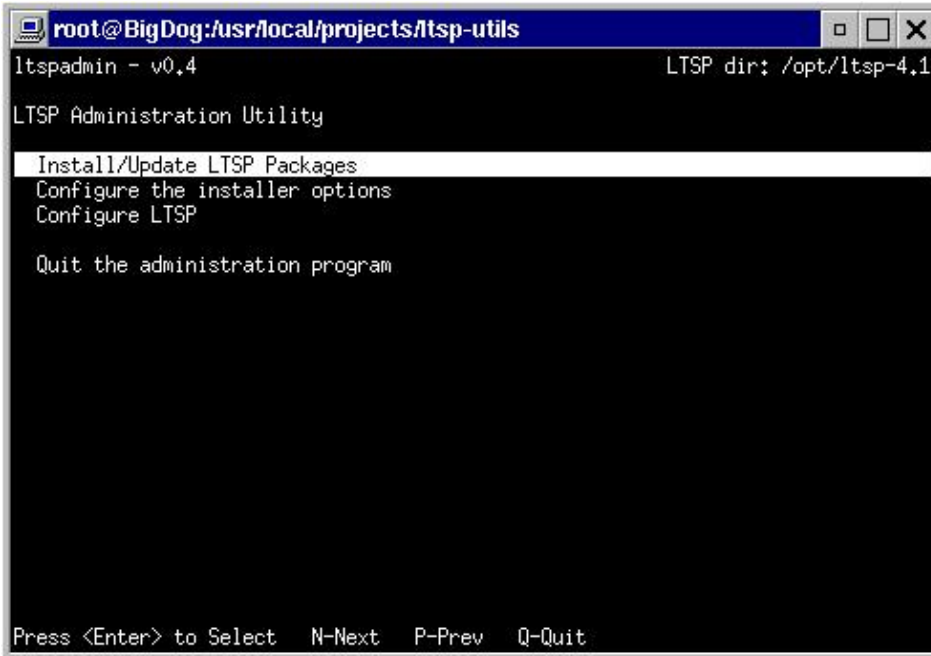
```
tar xzf ltsp-utils-0.1-0.noarch.tgz
cd ltsp_utils
./install.sh
cd ..
```

O comando acima irá instalar os utilitários LTSP no servidor. Isso é útil para servidores cujo gerenciamento de pacotes não é baseado em RPM.

2.2. Instalando os pacotes dos clientes LTSP

Uma vez que a instalação do pacote `ltsp-utils` estiver completa, você pode executar o comando **ltspadmin**. Este utilitário é utilizado para gerenciar os pacotes dos clientes LTSP. Ele irá consultar o repositório de download do LTSP, e obter a lista dos pacotes atualmente disponíveis.

Execute o comando **ltspadmin** e você irá ver uma tela como o seguinte:

A terminal window titled "root@BigDog:/usr/local/projects/ltsp-utils" showing the "ltspadmin - v0.4" utility. The window title bar also indicates "LTSP dir: /opt/ltsp-4.1". The main menu is titled "LTSP Administration Utility" and lists four options: "Install/Update LTSP Packages" (highlighted), "Configure the installer options", "Configure LTSP", and "Quit the administration program". At the bottom, it says "Press <Enter> to Select N-Next P-Prev Q-Quit".

```
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. Instalador LTSP – Tela Principal

Nesta tela, você pode escolher "Install/Update", e se esta for a sua primeira vez executando o utilitário, ele irá exibir a tela de configuração do instalador.



Figura 2–2. Instalador LTSP – Tela de Configuração

Na tela de *configuração*, você pode definir diversos parâmetros que o instalador irá utilizar, para fazer o download e instalar os pacotes LTSP. Estes parâmetros são:

Where to retrieve packages from

Esta é a URL, que aponta para o repositório de pacotes. Tipicamente, deverá ser `http://www.ltsp.org/ltsp-4.1`, mas se você desejar instalar os pacotes de um sistema de arquivos local, você pode usar `file:.` Por exemplo, se os pacotes estiverem em um CD-ROM, e você tiver montado o CD-ROM em `/mnt/cdrom`, então o valor para o repositório de pacotes será: `file:///mnt/cdrom`. (Note as três barras).

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

Este é o diretório no servidor, onde você deseja manter a árvore dos clientes LTSP. Tipicamente, será: `/opt/ltsp`. O diretório será criado, caso o mesmo ainda não exista.

Dentro deste diretório, o diretório raiz para cada uma das arquiteturas será criado. Atualmente, apenas as estações x86 são oficialmente suportadas pelo LTSP, mas há várias pessoas trabalhando em 'ports' para outras arquiteturas, como PPC e SPARC.

HTTP Proxy

Se o servidor estiver por trás de um firewall, e o acesso a web deva ser feito através de um proxy, você pode configurar o instalador para usar o proxy aqui. Os valores devem conter a URL para o proxy, incluindo o protocolo e a porta. Um exemplo para esta configuração é: `http://firewall.seudominio.com.br:3128`.

Se você não precisa de um proxy, deverá configurar esta opção para "none".

FTP Proxy

Para pacotes localizados em um servidor FTP, se você precisar acessar através de um proxy FTP, você pode informar aqui. A sintaxe é similar para a opção proxy HTTP acima.

Se você não precisa de um proxy, deverá configurar esta opção para "none".

Uma vez que você tenha concluído a tela da configuração, o instalador consultará o repositório de pacotes e obterá a lista dos componentes atualmente disponíveis.

```

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_addt1_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. LTSP installer – Component list

Selecione cada um dos componentes que deseja instalar. Para selecionar um componente, mova a linha de destaque para o componente desejado e pressione **T** para selecionar o componente individual. Você pode pressionar também **A** para selecionar TODOS os componentes. Na maioria das vezes, será este o seu objetivo. Desta forma, você poderá dar suporte à maioria dos hardwares dos terminais.

Há várias teclas que podem ser utilizadas para navegar nesta tela. Você pode obter ajuda para estas, pressionando a tecla **H**.

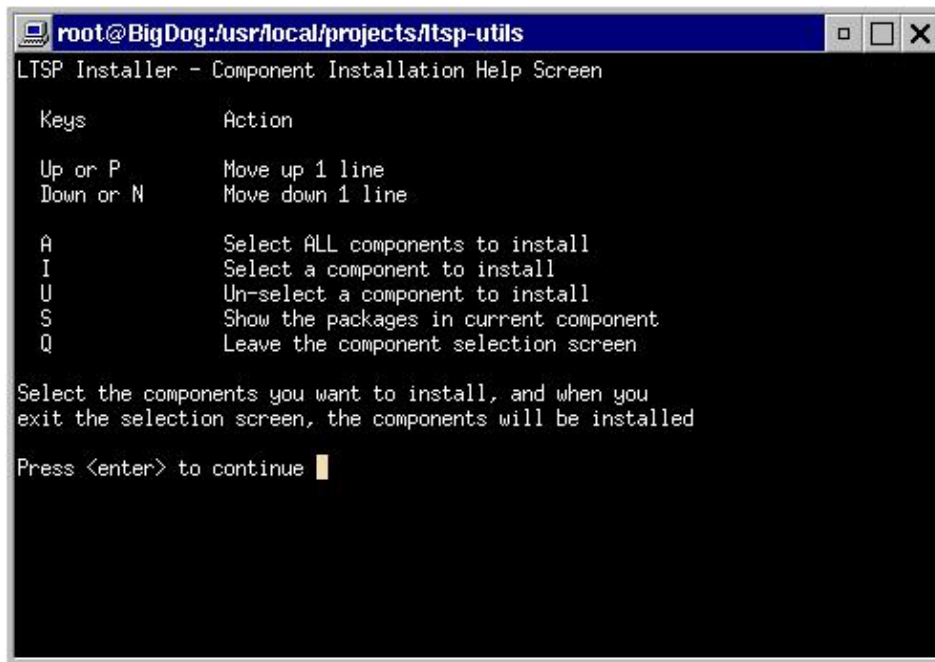


Figura 2–4. LTSP installer – Help window

Se você deseja ver a lista de pacotes que estão em um determinado componente, você pode pressionar 'S', e a lista de pacotes será exibida. Será mostrada a versão atualmente instalada, bem como a última versão disponível.

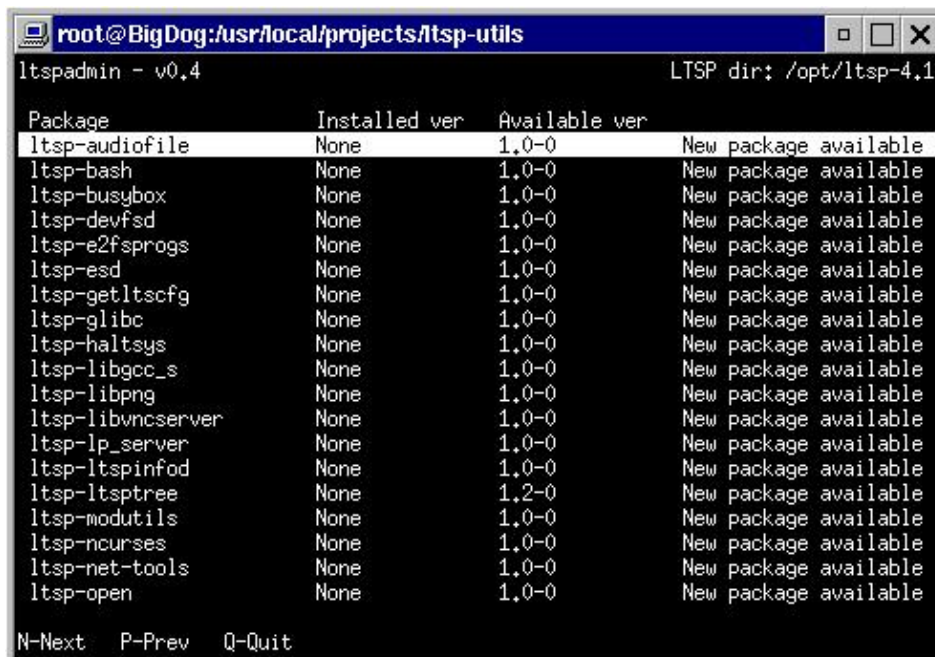


Figura 2–5. LTSP installer – Package list

Uma vez que os componentes desejados estejam selecionados, você pode sair da tela de seleção de componentes. O instalador irá perguntar a você, para se certificar que você realmente deseja instalar/atualizar

os pacotes selecionados. Se você responder 'Y', então ele irá fazer o download e a instalação dos pacotes selecionados.

2.3. Configurando os serviços necessários ao LTSP

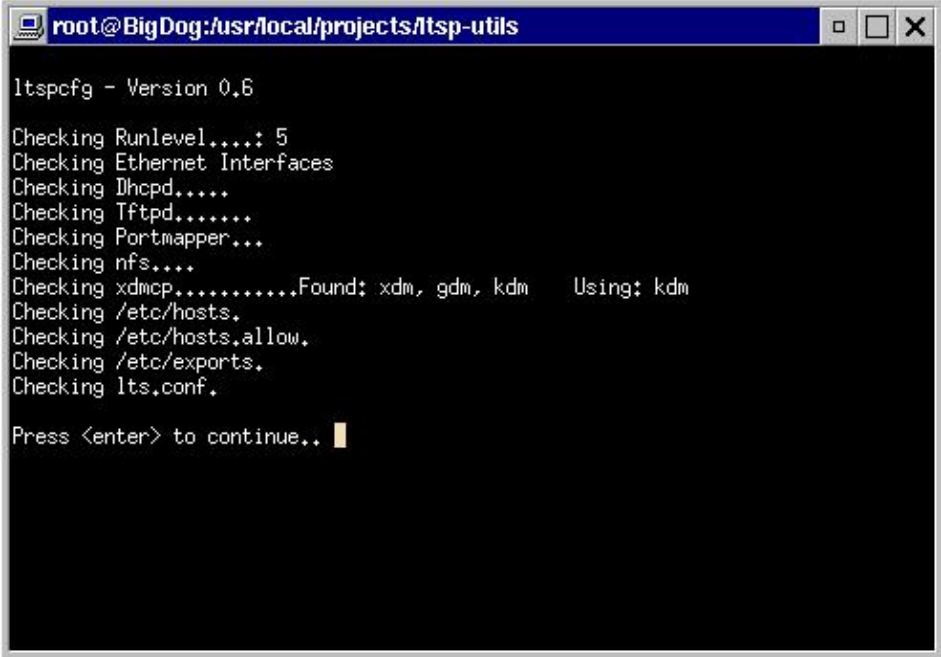
Há quatro serviços básicos necessários para suportar o boot de uma estação LTSP. São eles:

- DHCP
- TFTP
- NFS
- XDMCP

O **ltspcfg** pode ser utilizado para configurar todos estes serviços, além de muitas outras coisas relacionadas ao LTSP.

Você pode acessar o **ltspcfg** do **ltspadmin**, ou pode executá-lo digitando **ltspcfg** em um shell.

Quando você executar o utilitário **ltspcfg**, ele irá verificar o servidor, para determinar o que está atualmente instalado e executando. Você verá uma tela como esta:



```
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 lts.conf.
Press <enter> to continue..
```

Figura 2–6. **ltspcfg** – Initial screen

Esta tela mostra todos os itens que o utilitário procura.

Para configurar todos os itens que precisam de configuração, escolha 'C', e o menu de configuração será exibido. Do menu de configuração, você precisará percorrer por cada item, para certificar-se que estes estão configurados corretamente para servir às estações LTSP.

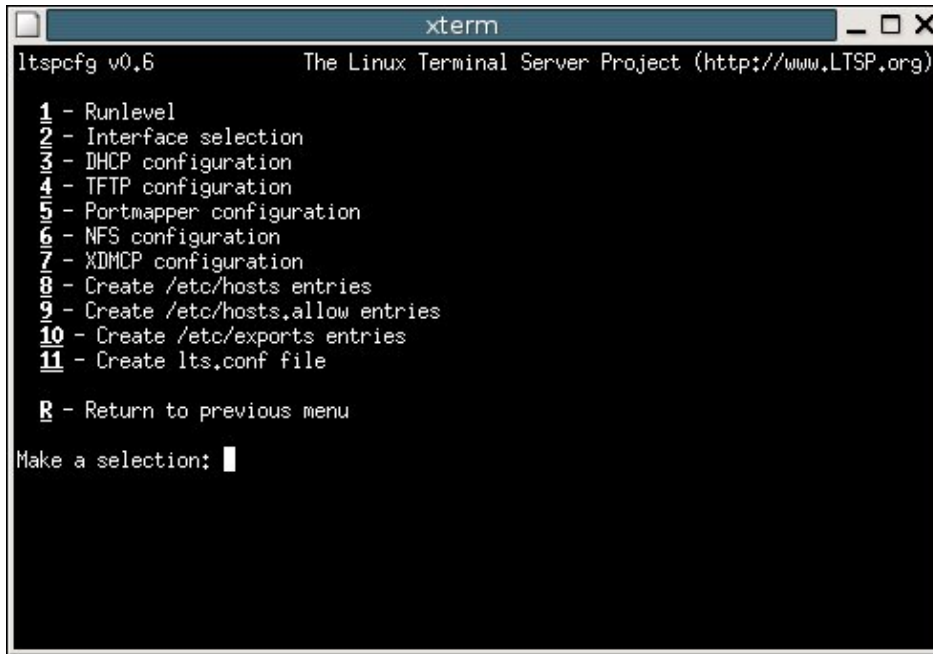


Figura 2–7. ltspcfg – Initial screen

1 – Runlevel

A variável **Runlevel** é usada pelo programa **init**. Com sistemas GNU/Linux e Unix, em um determinado momento, é dito estar em um "Runlevel" específico. Os Runlevels 2 ou 3 são tipicamente usados quando o servidor está em modo texto. O Runlevel 5 tipicamente indica que o sistema está em modo gráfico com suporte a rede.

Para um servidor LTSP, tradicionalmente o Runlevel 5 é usado. A maioria dos sistemas já está configurada para servir NFS e XDMCP quando no Runlevel 5. Para aqueles sistemas que ainda estão configurados para este fim, este utilitário tomará conta deles.

2 – Interface selection

Para sistemas que possuem múltiplas interfaces de rede, você precisará especificar qual interface os Thin clients estão conectados.

Selecionando a interface, a ferramenta de configuração está apta para criar os outros arquivos de configuração apropriados, como os arquivos `dhcpd.conf` e `/etc/exports`.

3 – DHCP configuration

O DHCP precisa ser configurado para fornecer as informações necessárias para as estações. Entre estas informações estão **fixed-address**, **filename**, **subnet-mask**, **broadcast-address** e **root-path**.

Selecionando este artigo de menu, você estará apto para criar o arquivo de configuração `dhcpd.conf`, e então ativar o **dhcpd** para executar no início do sistema.

4 – TFTP configuration

O TFTP é usado pelo 'thin client' para fazer o download do kernel Linux. O serviço **tftpd** precisa ser ativado no servidor, para servir os kernels.

5 – Portmapper configuration

O **Portmapper** é usado pelos serviços RPC. Cada um dos serviços RPC, tal como o NFS.

6 – NFS configuration

O NFS é o serviço que permite que a árvore de diretórios local seja montada nas máquinas remotas. Este é necessário ao LTSP, porque as estações montam seus sistemas de arquivos raiz do servidor.

Este item de menu irá cuidar para que o serviço NFS seja configurado para iniciar com o sistema. O arquivo de configuração é `/etc/exports` e sua criação será descrita mais adiante nesta seção.

7 – XDMCP configuration

O XDMCP é o "X Display Manager Control Protocol". O servidor X envia uma requisição XDMCP para o gerenciador de 'Display' no servidor para obter uma tela de login.

Os gerenciadores de 'display' comumente em uso são o **XDM**, **GDM** e **KDM**. Este item de menu irá mostrar qual gerenciador de 'display' foi encontrado, e qual está configurado para executar.

Por medida de segurança, o gerenciador de 'Display' é configurado por padrão para não permitir as conexões de estações remotas. Esta é normalmente a razão para a *Gray screen with large X cursor*. O `ltspcfg` pode normalmente configurar o gerenciador de 'display' para permitir que estações remotas se conectem ao servidor.

8 – Create `/etc/hosts` entries

Muitos serviços, como o NFS e o gerenciador de 'Display' precisam mapear o endereço IP da estação para um nome de host. Você pode configurar o Berkeley Internet Naming Daemon (BIND) para fazer isto, mas você deve certificar-se de configurar o *reverses* corretamente. Finalmente, usar o BIND é provavelmente a melhor maneira fazê-lo, mas a configuração do BIND é além da finalidade deste documento e do utilitário `ltspcfg`.

Uma maneira mais simples para configurar o mapeamento de endereços IP e nomes de host é o arquivo `/etc/hosts`.

9 – Create `/etc/hosts.allow` entries

Alguns serviços usam uma camada de segurança conhecida como *tcpwrappers*. Este é configurado através do arquivo `/etc/hosts.allow`. Este item do menu irá configurar isto para você.

10 – Create the `/etc/exports` file

Este é o arquivo usado pelo NFS, para determinar quais diretórios permitem ser montados por máquinas remotas. Este item do menu criará este arquivo.

11 – Create the `lts.conf` file

A configuração de cada estação é direcionada por entradas no arquivo `lts.conf`. Para estações razoavelmente modernas com um barramento PCI, não deve ser necessária nenhuma entrada adicional no `lts.conf`. Mas, o arquivo ainda precisa existir. Este item do menu criará o arquivo `lts.conf` padrão para você.

2.4. Configurações específicas da estação

Agora, é o momento de informar o servidor LTSP sobre uma estação específica. Há três arquivos que contêm informações sobre as estações de trabalho.

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

2.4.1. /etc/dhcpd.conf

A estação precisa de um endereço IP e outras informações. Ela irá obter as seguintes do servidor DHCP:

- Endereço IP
- Nome de host
- Endereço IP do servidor
- Gateway padrão
- Caminho do kernel a ser carregado
- Servidor e caminho para o diretório que será montado como sistema de arquivos raiz

Para o nosso exemplo de ambiente LTSP, nós escolhemos o DHCP para gerenciar a atribuição de endereços IP das estações.

Durante a execução do script **ltsp_initialize script**, um arquivo `dhcpd.conf` é instalado. Ele estará localizado em `/etc/dhcpd.conf.example`, você pode copiá-lo para `/etc/dhcpd.conf` para utilizá-lo como base para a sua configuração do serviço DHCP. Você precisará modificar as partes deste arquivo que referenciam o ambiente de servidor(s) e estações.

```
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

Desde a versão 2.09pre2 do LTSP, você não precisa especificar um kernel particular a ser carregado. O pacote padrão do kernel suporta todos as placas da rede que o Linux suporta. Há dois arquivos de kernel incluídos no pacote do kernel LTSP. Um kernel tem o 'Linux Progress Patch' (LPP) aplicado, e o outro não. Os nomes para os kernels são:

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

Você pode ter observado que o kernel reside no diretório `/tftpboot/lts`, mas na entrada "filename" no arquivo `/etc/dhcpd.conf` está faltando o componente `tftpboot` do caminho. Isto é porque nas versões 7,1 e superiores do Redhat, o TFTP está sendo executado com a opção `'-s'`. Esta opção leva ao daemon do `tftpd` a funcionar na modalidade *secure*. Isto é, faz um **chroot** para o diretório `/tftpboot` quando inicia. Conseqüentemente, todos os arquivos que estão disponíveis ao daemon do `tftpd` são relativos ao diretório `tftpboot`.

Outras distribuições Linux podem não possuir a opção `'-s'` configurada para o `tftpd`, desta forma você precisará adicionar o prefixo `/tftpboot` ao caminho do kernel.

2.4.2. /etc/hosts

Mapeamento de endereços IP para nome de máquina

Os computadores comunicam-se perfeitamente com os endereços do IP. Então, nós seres humanos precisamos pôr nomes nos computadores, porque não podemos recordar os números. É onde o DNS ou o arquivo `/etc/hosts` entra no jogo. Este mapeamento de endereços IP para nomes de host geralmente não é necessário, exceto em um ambiente LTSP. Isto porque sem ele, o NFS lhe dará erros das permissões quando a estação de trabalho tenta montar o sistema de arquivos raiz.

Além dos problemas com o NFS, se a estação de trabalho não estiver listada no arquivo `/etc/hosts`, você poderá ter problemas também com gerenciadores de telas *GDM* ou *KDM*.

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

Há um número de entradas de configuração que podem ser especificadas no arquivo `lts.conf`.

O arquivo `lts.conf` possui uma sintaxe simples, que consiste em seções múltiplas. Há uma seção padrão chamada **[default]** e pode haver seções para estações individuais. As estações podem ser identificadas pelo nome de host, pelo endereço IP ou pelo endereço MAC.

Um arquivo `lts.conf` típico se parece com este:

```
#
# 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

```

Exemplo 2–1. Arquivo `lts.conf`

Abaixo, uma lista das principais entradas:

XSERVER

Se sua placa de vídeo é PCI, e se esta é suportada pelo X.org 6.7.0, então você apenas precisa do pacote `lts_x_core`. Este contém todos os módulos de driver para o X4.

Há vários pacotes do XFree86 3.3.6 disponíveis para o LTSP. Isto é para o caso de sua placa de vídeo não ser suportada pelo X.org 6.7.0.

Você pode criar entradas no arquivo `lts.conf` para cada estação individual, ou você pode criar uma entrada padrão que pode ser compartilhada por todas as estações.

Nossa estação de trabalho possui uma placa de vídeo com chipset Intel i810, e esta pode ser detectada automaticamente, então nós não precisaremos de uma entrada `XSERVER` no arquivo `lts.conf`. A entrada `XSERVER` pode ser especificada, se você desejar, ou esta pode ser configurada para 'auto' para mostrar que a placa de vídeo será detectada automaticamente.

RUNLEVEL

Queremos utilizar a estação de trabalho no modo gráfico, desta forma, precisamos que o runlevel esteja configurado para '5'. Isto é feito por outra entrada no arquivo `lts.conf`.

2.5. Exibindo a configuração atual

Com o comando `ltspcfg`, você pode obter o status atual da configuração de todos os serviços necessários ao LTSP. Através do menu principal do `ltspcfg`, pressione 'S', e você irá ver o status atual.

```

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/lts.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

Capítulo 3. Configurando a estação de trabalho

Uma vez que o servidor esteja configurado, é hora de concentrar-se na configuração da estação de trabalho.

Todo o que acontece depois que o kernel está na memória é sobre o projeto de LTSP. Há diversas maneiras de pôr o kernel na memória, incluindo Etherboot, Netboot, PXE e o disquete.

3.1. Carga com PXE

Se sua placa de rede possui PXE embutido, então você pode usá-la para carregar o kernel do Linux. PXE é uma tecnologia de bootrom, similar a Etherboot ou a Netboot.

Talvez você precise ativar o bootrom PXE em sua placa de rede. Você também pode precisar mudar a ordem dos dispositivos de boot em sua BIOS, para fazer a "Boot from LAN" a primeira escolha, ao invés de "Floppy" ou "HDD".

O PXE tem uma limitação de somente poder carregar arquivos de 32kb ou menores. O kernel do Linux é um pouco maior que esta limitação, desta forma não é possível carregar o kernel Linux diretamente com o PXE. Você precisa carregar algo conhecido como 'Network Bootstrap Program' ou NBP.

Há um NBP disponível para carregar o kernel Linux chamado de **pxelinux.0**. Este é parte do pacote **syslinux** de H. Peter Anvin, um dos desenvolvedores do kernel.

O pacote LTSP do kernel inclui o NBP pxelinux.0 e o arquivo de configuração necessário para carregar o kernel Linux e a imagem do ramdisk.

A maneira de funcionamento é esta:

- A bootrom PXE inicia a placa de rede e envia uma requisição DHCP.
- O servidor DHCP responde à requisição com um endereço IP, e o nome do NBP a ser carregado.
- A bootrom PXE faz o download do NBP, o põe em memória e inicia sua execução.
- O NBP utiliza-se do tftp para fazer o download do arquivo de configuração do servidor.
- O arquivo de configuração contém o nome do kernel, o nome do arquivo ramdisk inicial, e opções a serem passadas ao kernel, quando este for carregado.

Aqui está um arquivo de configuração exemplo do 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
```

- O NBP então usa o tftp para fazer o download do kernel Linux, e do ramdisk inicial (initrd).
 - O controle é então passado ao kernel Linux, este é carregado em memória, monta o initrd, e continua com o início do thin clinet.
-

3.2. Carga com Etherboot

O Etherboot é um pacote de software para criar imagens ROM que possam realizar o download de código sobre uma rede Ethernet a ser executado em um computador x86. Muitas placas de rede têm um soquete onde um chip de ROM pode ser instalado. O Etherboot é o código que pode ser posto em tal ROM.

—Ken Yap

O Etherboot é também Open Source, protegido pela GNU General Public License, Version 2 (GPL2).

Para usar o Etherboot, se você já possui uma placa de rede com uma bootrom Etherboot, talvez precise modificar a configuração de sua BIOS para escolher a opção "Boot from LAN" para a carga do sistema operacional ao invés de "Floppy" ou "HDD".

Se você não possui uma bootrom Etherboot, você pode ou criar uma bootrom, ou você pode criar um disquete com uma imagem Etherboot em seu setor de boot.

O Etherboot suporta um vasto número de placas de rede. Mais de 200 modelos, com mais sendo adicionados todo o tempo. Se você escolher criar um disquete ou gravar o código em uma Eprom, precisará determinar qual o modelo de placa de rede possui.

3.2.1. Escolhendo um driver Etherboot para placas de rede ISA

Para placas de rede antigas baseadas no padrão ISA, não é tão importante que você determine o tipo exato. Primeiro, a maioria delas são placas ne2000 ou 3Com 3c509. Você precisa apenas obter o driver Etherboot correto, o que seleciona o tipo correto de mídia na placa 10 base-2 (Coax) e 10 base-T (Twisted pair).

3.2.2. Escolhendo um driver Etherboot para placas de rede PCI

Para as placas de rede PCI, é importante escolher o driver Etherboot que corresponda ao código PCI do fabricante e da placa de rede.

Às vezes, você terá sorte. Saberá exatamente que modelo de placa de rede tem, por que o modelo é impresso na própria placa de rede, e bate exatamente com a descrição dos modelos no Etherboot. Mas, em muitos casos, será necessário encontrar os números do PCI ID.

Se sua estação de trabalho tiver uma drive de disquete, você pode carregar um disquete tomsrtbt (Tom's Root Boot). Ou, se sua estação de trabalho tiver uma unidade de CD-ROM, você pode carregar um CD do Knoppix. Se você não puder carregar o linux em sua estação de trabalho, então sua única esperança pode ser mover a placa de rede para uma estação de trabalho que possa carregar o Linux.

Uma vez que tenha o Linux carregado, você pode usar o comando **lspci** com a opção '-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)
```

No exemplo acima, você pode ver uma entrada para cada placa PCI no sistema. Você precisará procurar apenas os dispositivos **Class 0200**. Desta forma, executado novamente o comando, procurando apenas por interfaces Ethernet, a lista torna-se muito mais gerenciável.

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

Os números do PCI ID são: **8086:1229**. O primeiro campo, **8086** é o PCI ID do fabricante. Neste exemplo, o fabricante é a Intel Corporation. O segundo campo, **1229** é o PCI ID do dispositivo. Este nos informa qual o modelo da placa de rede. Neste caso, é uma placa de rede EtherExpress 100.

3.2.3. Criando um disquete de boot

Você pode fazer o download do pacote Etherboot e configurá-lo para o tipo de bootrom que precisar. Então, você pode compilar o fonte para produzir uma imagem bootrom que possa ser gravada em uma EPPROM ou em um disquete.

Uma abordagem simples é ir ao site do Marty Connor's www.Rom-O-Matic.net.

O Marty fez um bom trabalho ao por um front-end web para configuração e compilação da geração de imagens bootrom Etherboot. Neste site, você seleciona o tipo de placa de rede que tem e que tipo de imagem deseja. Então, você tem a oportunidade de modificar muitas das opções de configuração do Etherboot. Então, você pode pressionar o botão 'Get ROM' e uma imagem bootrom personalizada será gerada enquanto você espera.

Para o formato de saída da ROM, escolher 'Floppy Bootable ROM Image'. Isto irá ocasionar a inclusão de um cabeçalho de 512 bytes que é um boot loader para carregar a imagem etherboot na memória ram onde esta possa ser executada.

Pressione o botão 'Get ROM'. A imagem bootrom será gerada enquanto você espera.. Isto leva apenas alguns segundos, e quando completar, seu navegador irá abrir a janela de "Salvar como" onde você pode definir onde a imagem bootrom será salva em seu computador.

Uma vez que tenha salvo a imagem em seu disco, você precisará gravá-la em um disquete. Insira um disquete na unidade e execute o seguinte comando para gravar o disquete:

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

3.2.4. Criando uma bootrom

É necessário um programador de EPROM para gravar a imagem do Etherboot em uma EPROM. Este tipo de equipamento varia de preço entre poucas centenas de dólares até milhares de dólares dependendo das características.

O processo de criação de uma bootrom é inteiramente dependente do programador de EPROM. Isto está fora do escopo deste documento.

Capítulo 4. Ligando a estação

Assumindo que o servidor e a estação estão configurados corretamente, é necessário apenas inserir um disquete de boot na estação de trabalho e ligá-la.

O código de Etherboot será lido do disquete para a memória, a placa de rede será encontrada e inicializada, uma requisição dhcp será emitida na rede e uma resposta será emitida pelo servidor e será feito o download doo kernel para a estação de trabalho. Uma vez que o kernel inicializou o hardware da estação de trabalho, o X Windows irá iniciar e uma tela de login deve aparecer na estação de trabalho, similar ao exemplo abaixo.



Figura 4–1. Login screen

Neste ponto, você pode logar. Uma coisa importante a ter em mente é que você está logando no servidor. Todos os comandos que você executar, estarão sendo executados no servidor, e sua saída mostrada na estação de trabalho. Isto é o poder do X Windows.

Você pode executar qualquer programa que seja suportado pelo servidor.

Capítulo 5. Impressão

Além da estação de trabalho ser uma GUI funcional ou um terminal, esta pode atuar como um servidor de impressão, permitindo a conexão de até 3 impressoras conectadas a portas paralelas ou seriais.

Tudo isso é transparente para os usuários das estações de trabalho. Eles nem mesmo notarão o pequeno tráfego que está saindo das estações para as impressoras.

5.1. Configurações no lado cliente

O LTSP usa o programa **lp_server** na estação de trabalho, para redirecionar os trabalhos de impressão do servidor para a impressora conectada a uma das portas da estação de trabalho.

Para habilitar a impressora na estação de trabalho, há um conjunto de entradas de configuração no arquivo **lts.conf**.

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

As entradas acima ocasionarão na execução do programa **lp_server** como um daemon, ouvindo na porta TCP/IP 9100 por um fluxo de impressão do servidor. Os dados de impressão serão então redirecionados para a impressora conectada à porta paralela **/dev/lp0**.

Há muito mais opções disponíveis. Verifique posteriormente as seções do **lts.conf** neste documento para maiores informações sobre as entradas para a configuração de impressoras.

5.2. Configurações no lado do servidor

Configurar a impressora no servidor consiste em definir uma fila de impressão, usando a ferramenta de configuração de impressoras do servidor.

No Redhat 7.2, há ambas as ferramentas de configuração de impressoras baseadas em GUI e Texto. A ferramenta GUI é chamada de **printconf-gui**, e a baseada em Texto é chamada de **printconf-tui**. Em versões mais antigas o Redhat possui um programa chamado **printtool**. O **printtool** também existe no Redhat 7.2, mas este chamará o **printconf-gui**. Outras distribuições Linux possuem suas próprias ferramentas de configuração de impressoras.

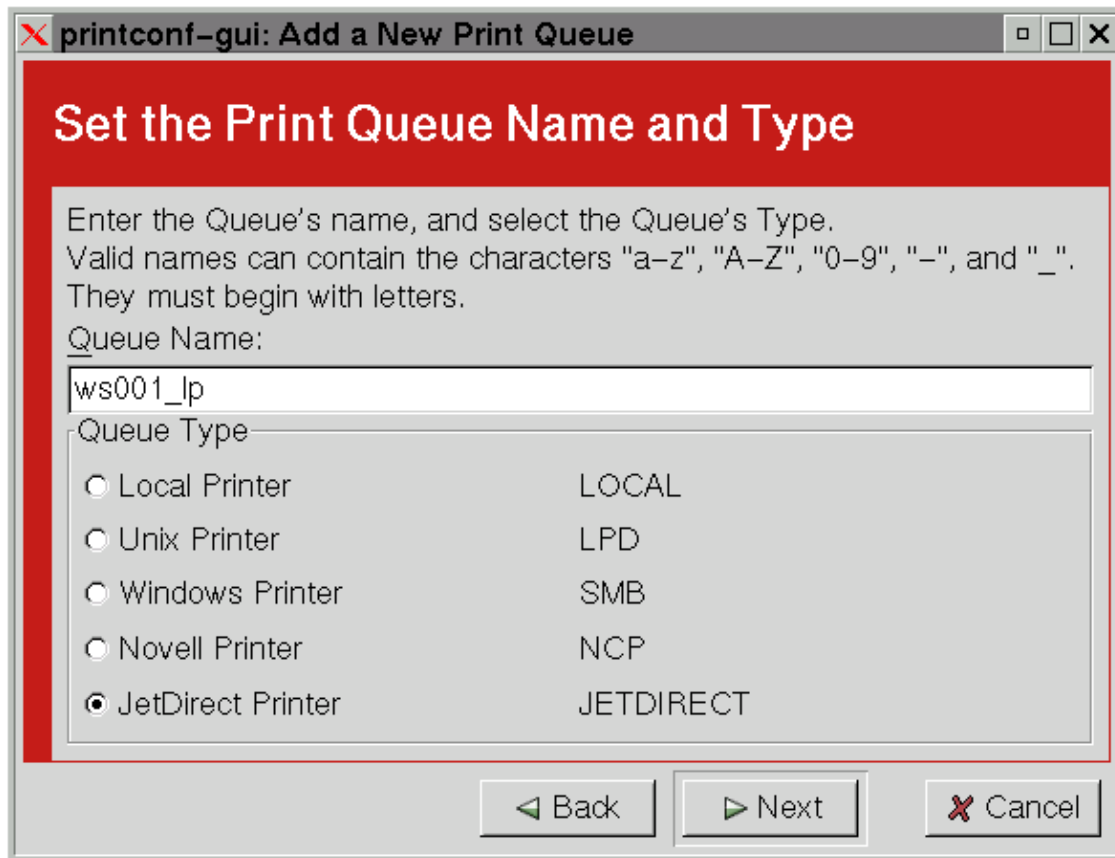


Figura 5–1. Printconf–gui Adicionando uma nova impressora

Uma vez que você execute a ferramenta de configuração de impressoras, será necessário adicionar uma nova impressora. O programa `lp_server` permite à estação de trabalho emular um servidor de impressão HP JetDirect. Você apenas precisa criar uma impressora **JetDirect**.

Você precisa atribuir um nome à fila de impressão. O nome pode ser qualquer coisa, mas crie um nome inteligível, e o nome pode conter apenas os seguintes caracteres:

- "a-z" letras minúsculas
- "A-Z" letras maiúsculas
- "0-9" dígitos numéricos
- "-" hífen
- "_" sublinhado

O nome escolhido no exemplo acima é **ws001_lp**. Este nome torna fácil ver que a impressora está associada com **ws001**.

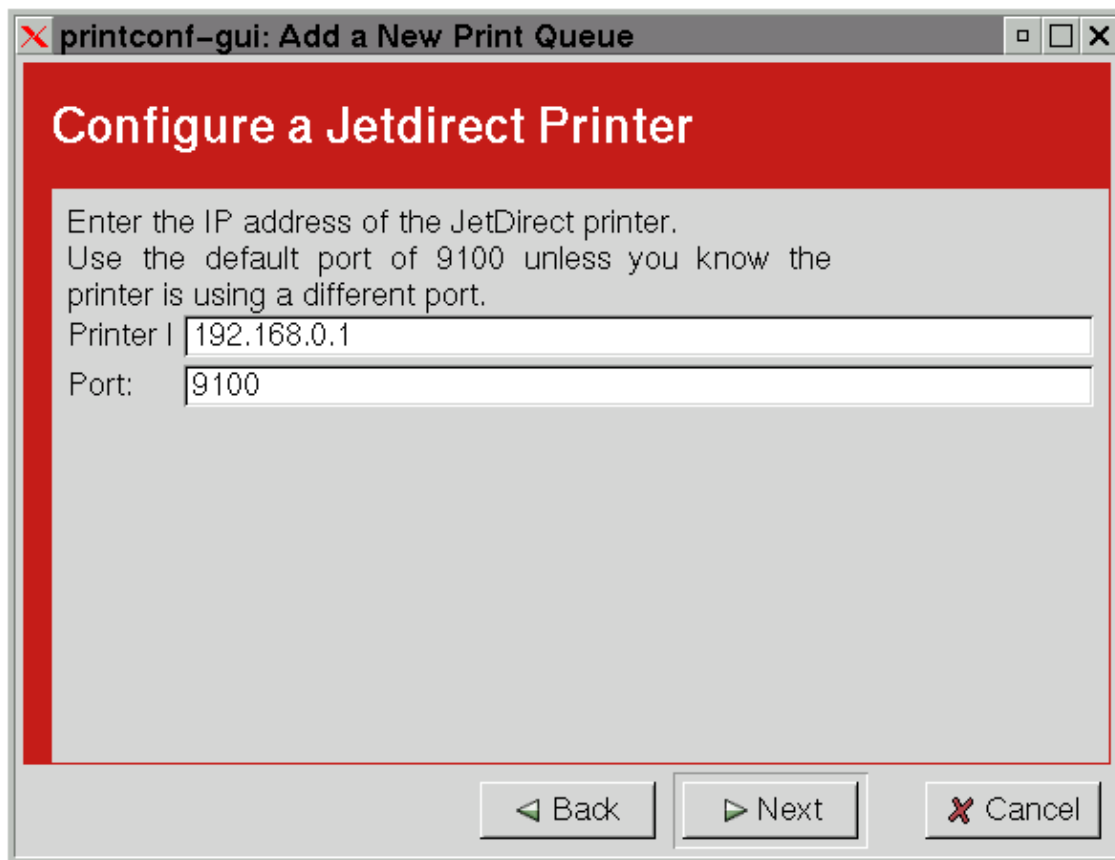


Figura 5–2. Printconf–gui Informações detalhadas

Há dois campos obrigatórios para a comunicação com a impressora:

1. Endereço IP ou nome de host da estação de trabalho a qual a impressora está associada.
2. A porta TCP na qual o daemon **lp_server** está ouvindo.

A primeira impressora que você conectar a uma estação de trabalho estará na porta TCP/IP **9100**. A segunda impressora estará na porta **9101**, e a terceira impressora estará na porta **9102**.

Capítulo 6. Scripts de tela

Uma das funcionalidades que foram adicionadas à versão 4.0 do LTSP é algo chamado **Screen Scripts**. Há script para iniciar vários tipos de sessões.

Você pode especificar múltiplos scripts de tela para uma estação de trabalho. Fazendo isto, você terá múltiplas sessões. Elas podem ser de tipos diferentes, ou elas podem ser do mesmo tipo. Por exemplo, você pode especificar o seguinte:

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

Isto deverá iniciar um servidor X na primeira tela, e um shell na segunda. Você pode alternar entre as telas pressionando Ctrl–Alt–F1 para ir à primeira tela, e Ctrl–Alt–F2 para ir à segunda.

Você pode especificar até 12 scripts de tela para uma estação de trabalho, mas a maioria das pessoas possuem apenas uma.

Tipos de scripts de tela disponíveis:

startx

Este script iniciará o servidor X com a opção `–query`, para enviar uma requisição XDMCP ao gerenciador de janelas para obter a tela de login exibida na estação.

shell

Este script iniciará um shell no terminal. Este tem por objetivo o uso na solução de problemas com a estação de trabalho. Porque ele lhe fornece uma sessão no thin client, ao invés de ser no servidor, não sendo muito útil para executar aplicações.

telnet

Este script executará uma sessão telnet que conectará ao servidor. Este lhe fornecerá uma sessão baseada em caracteres no servidor.

Por padrão, o telnet conectará ao servidor LTSP. Se você deseja especificar um servidor diferente, você pode passar este na mesma linha do script de tela. Por exemplo:

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

Você pode ainda especificar qualquer opção que o Telnet reconheça, Mas, se você especificar qualquer opção, terá que informar a qual servidor será feita a conexão.

rdesktop

Este script iniciará o programa rdesktop, o qual conectará a um servidor Microsoft Windows. Você pode especificar na mesma linha qualquer opção do rdesktop que desejar, diretamente após o nome do script de tela. Por exemplo, se você deseja especificar o servidor ao qual a conexão deverá ser realizada, você pode fazê-lo desta forma:

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


O exemplo acima iniciará o rdesktop no modo tela cheia. O usuário verá uma tela de logon Windows, e ele terá que logar apenas uma vez. Isto é muito útil quando você quer apenas um login Windows, sem haver um login Linux ou gerenciador de janelas. O usuário nem mesmo saberá que está utilizando Linux.

Os scripts de tela residem no diretório `/opt/ltsp/i386/etc/screen.d`. Você pode criar seus próprios scripts de tela e pô-los neste diretório. É melhor usar um dos scripts existentes como exemplo.

Capítulo 7. Solução de Problemas

Se, após seguir os capítulos anteriores, sua estação de trabalho não carregar, então você deve começar o processo de pesquisar os problemas de instalação.

A primeira coisa a fazer é descobrir em que ponto do processo de carga a estação ficou.

7.1. Solucionando problemas em disquetes com imagens Etherboot

Quando você iniciar por um disquete, você deverá ver algo similar a isto:

```
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>
```

O exemplo acima mostra o que você pode esperar ver na tela quando iniciando de um disquete. Se você não vir aquelas mensagens, indicando que o Etherboot iniciou, então você pode ter um disquete com defeito, ou você não gravou a imagem corretamente.

Se, você vir uma mensagem como a seguinte, então isto indica que provavelmente a imagem Etherboot que você gerou não é a imagem correta para sua placa de rede.

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

Se for até o ponto onde ele detecta a placa de rede e mostrar o endereço MAC correto, então o disquete provavelmente está normal.

7.2. Solucionando problemas no DHCP

Uma vez que a placa de rede esteja iniciada, ela irá enviar uma requisição DHCP para a rede local, procurando por um servidor DHCP.

Se a estação de trabalho receber uma resposta válida do servidor DHCP, então ela configurará a placa de rede. Você saberá se funcionou corretamente caso as informações de endereço IP forem indicadas na tela. Aqui está um exemplo de como deve aparecer:

```
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 você vir uma linha que começa com 'Me:', seguida por um endereço IP, então você sabe que o DHCP está funcionando corretamente. Você pode ir em frente e verificar se o TFTP está funcionando.

Se ao invés, você vir a seguinte mensagem na estação de trabalho, seguida de várias mensagens <sleep>, então algo está errado. Embora, seja comum ver uma ou duas mensagens <sleep>, depois da resposta do servidor dhcp.

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

Descobrir o que está errado pode ser algumas vezes difícil, mas aqui estão algumas coisas a observar.

7.2.1. Verificar conexões

A estação de trabalho está conectada à mesma rede que o servidor está conectado?

Com a estação de trabalho ligada, certifique-se de que as luzes de conexão estão acesas durante toda a conexão.

Se você estiver conectando diretamente a estação e o servidor (sem hub ou switch), certifique-se que você está usando um cabo cross-over. Se você estiver usando um hub ou switch, então certifique-se de estar utilizando um cabo normal straight-thru, entre ambos a estação e o hub, e o hub e o servidor.

7.2.2. O DHCP está sendo executado?

Você precisa determinar se o **dhcpcd** está sendo executado no servidor. Podemos encontrar a resposta de uma grande variedade de modos.

O **dhcpcd** normalmente permanece em segundo plano, ouvindo na porta udp 67. Tente executar o comando **netstat** para ver se existe algo ouvindo nesta porta:

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

Você deverá ver uma saída similar a esta:

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

A quarta coluna contém o endereço IP e a porta, separados por dois pontos (':'). Um endereço composto por zeros ('0.0.0.0') indica que ele está ouvindo em todas as interfaces. Isto é, você pode ter as interfaces **eth0** e **eth1**, e o **dhcpcd** ouvindo em ambas as interfaces.

Só porque o netstat mostra que alguma coisa está ouvindo na porta udp 67, não significa que é definitivamente o **dhcpcd** que está escutando. Pode ser o **bootpd**, mas isto é incomum, porque o **bootp** não é mais incluído na maioria das distribuições Linux.

Para certificar-se que é o **dhcpcd** que está sendo executado, tente executar o comando **ps**.

```
ps aux | grep dhcpcd
```

Você deverá ver algo parecido com o seguinte:

```
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 dhcp
```

A primeira linha mostra que o **dhcpcd** está sendo executado. A segunda linha apenas o comando **grep**.

Se você não vir nenhuma linha mostrando que o **dhcpcd** está sendo executado, então você precisa checar se o servidor está configurado para o runlevel 5, e que o **dhcpcd** está configurado para iniciar no runlevel 5. Em sistemas baseados em Red Hat, você pode executar o comando **ntsysv** e procurar o **dhcpcd** para certificar-se que ele está configurado para iniciar.

Você pode tentar iniciar o **dhcpcd** com este comando:

```
service dhcpcd start
```

Preste atenção à saída, ela pode mostrar erros.

7.2.3. Verifique novamente a configuração do dhcpcd

O arquivo `/etc/dhcpcd.conf` possui uma entrada para a estação de trabalho?

Você deve verificar novamente a configuração 'fixed-address' no arquivo de configuração, para certificar-se que esta casa com a placa de rede na estação de trabalho.

7.2.4. O IPTables ou IPChains estão bloqueando as requisições?

7.2.4.1. Verificando o ipchains

Execute o seguinte comando para ver o que ele diz:

```
ipchains -L -v
```

Se você vir algo como:

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

Então não é o ipchains que está atrapalhando.

7.2.4.2. Verificando o iptables

Execute o seguinte comando para ver o que ele diz:

```
iptables -L -v
```

Se você vir algo como:

```
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
```

Então não é o iptables que está atrapalhando.

7.2.5. A estação de trabalho está enviando a requisição?

Tente monitorar o arquivo `/var/log/messages` enquanto a estação de trabalho estiver iniciando. Você pode fazê-lo com o seguinte comando:

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

Isto 'seguirá' o arquivo de log quando novos registros forem adicionados a ele.

```
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 você vir mensagens como as acima, informando 'no free leases', isto indica que o **dhcpd** está sendo executado, mas este não sabe nada sobre a estação de trabalho que está requisitando um endereço IP.

7.3. Solucionando problemas no TFTP

O Etherboot usa o TFTP para copiar o kernel Linux do servidor. Este é um protocolo muito simples, mas algumas vezes há problemas ao tentar fazê-lo funcionar.

Se você vir uma mensagem similar a esta:

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

com os pontos enchendo a tela rapidamente, isto normalmente indica que o TFTP está funcionando corretamente, e que o kernel está sendo copiado.

Se, ao invés, você não vir os pontos, então há um problema. Problemas possíveis incluem:

7.3.1. O tftpd não está sendo executado

Se o **tftpd** não estiver configurado para executar, então ele certamente não estará apto a responder às requisições da estação de trabalho. Você pode ver se ele está sendo executado, você pode usar o comando **netstat**, como este:

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

Se você não vir nenhuma saída deste comando, então o tftpd não está sendo executado.

Há dois métodos comuns de executar o tftpd, são eles o **inetd** e o novo **xinetd**

O **inetd** usa um arquivo de configuração chamado `/etc/inetd.conf`. Neste arquivo, certifique que a linha que começa com tftpd não está comentada. Isto é como a linha deve parecer:

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

O **xinetd** usa um diretório com arquivos de configuração individuais. Um para cada serviço. Se o seu servidor está usando o xinetd, então o arquivo de configuração para o tftpd é chamado de `/etc/xinetd.d/tftp`. Abaixo está um exemplo:

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

Certifique-se que a linha **disable** não contém **yes**.

7.3.2. O kernel não está onde o tftpd espera encontrá-lo

O kernel precisa estar em um local onde o daemon tftpd possa acessá-lo. Se a opção '-s' do daemon **tftpd** for utilizada, então qualquer coisa que a estação procurar precisa estar relativo ao diretório `/tftpboot`. Então, se a entrada **filename** no arquivo `/etc/dhcpd.conf` está configurada para `/lts/vmlinuz-2.4.24-ltsp-4`, então o kernel precisa estar em `/tftpboot/lts/vmlinuz-2.4.24-ltsp-4`

7.4. Solucionando problemas do sistema de arquivos raiz NFS

Há diversas coisas que podem impedir que o sistema de arquivos raiz seja montado, incluindo as seguintes:

7.4.1. No init found

Se você obter o seguinte erro:

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

Então o mais provável é que você esteja montando o diretório errado como sistema de arquivos raiz ou o diretório `/opt/ltsp/i386` está vazio.

7.4.2. O servidor retornou o erro -13

Se você obter o seguinte erro:

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

Isto indica que o diretório `/opt/ltsp/i386` não está listado no arquivo `/etc/exports`.

Examine o arquivo `/var/log/messages` para ver se há algum indício. Uma entrada como esta:

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

Então isto confirma nossa suspeita que a entrada no arquivo `/etc/exports` não está correta.

7.4.3. Problemas no Daemon NFS (portmap, nfsd & mountd)

O NFS pode ser um serviço complexo e difícil na solução de problemas, mas entendendo o que deve ser configurado e quais ferramentas estão disponíveis para diagnosticar os problemas irá certamente ajudar a tornar isto mais fácil.

Há três daemons que precisam estar em execução no servidor para o NFS funcionar corretamente. **portmap**, **nfsd** e **mountd**.

7.4.3.1. O Portmapper (portmap)

Se você obter a seguinte mensagem:

```
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
```

Isto é causado muito provavelmente porque o daemon do portmap não está em execução. Você pode confirmar se o portmapper está funcionando usando o comando `ps`:

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

Se o portmapper estiver executando, você deverá ver uma saída como esta:

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

Outro teste é usar o **netstat**. O portmapper usa as portas TCP e UDP 111. Tente executar isto:

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

Você deverá ver a seguinte saída:

```
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 você não vir uma saída similar, então o portmapper não está executando. Você inicia o portmapper executado:

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

Então, você deverá certificar-se que o portmapper está configurado para iniciar quando o servidor é iniciado. Execute o **ntsysv** para certificar-se que este está selecionado para executar.

7.4.3.2. O NFS e daemons MOUNT (nfsd & mountd)

O NFS possui 2 daemons que precisam estar executando. O **nfsd** e o **mountd**. Ambos são iniciados pelo script `/etc/rc.d/init.d/nfs`.

Você pode executar o comando **ps** para certificar-se que eles estão em execução.

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

Se isto mostrar que um ou ambos os daemons não estão executando, então você precisará iniciá-los.

Você deve estar apto a executar o script de inicio com o argumento **restart** para causar o inicio de ambos, mas por alguma razão, o script `/etc/rc.d/init.d/nfs` não reinicia o **nfsd** desta maneira. Ele reinicia somente o **mountd** (erro?). Assim, você deve preferivelmente executar a seguinte seqüência dos comandos:

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

Você pode obter erros no comando **stop**, mas isto está OK. O comando **start** deve mostrar **OK** como status.

Se os daemos estiverem executando, mas o NFS continuar não funcionando, você pode verificar se eles se registraram com o portmapper utilizando o comando **rpcinfo**.

```
rpcinfo -p localhost
```

Você deve ver resultados similares aos abaixo:

```
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
```

Isto indica que o **nfs** (nfsd) e o **mountd** estão ambos executando e estão registrados com o portmapper.

7.5. Solucionando problemas no Xserver

Ah garoto!, Provavelmente a única parte a mais difícil de configurar uma estação de trabalho LTSP é ter o servidor X configurado corretamente. Se você estiver usando uma placa de video razoavelmente nova, e esta for suportada pelo Xorg Xservers, e você tiver um monitor razoavelmente novo que possa gerenciar uma grande escala de frequências e definições, então é razoavelmente fácil seguir em diante. Geralmente, nesse caso, se não funcionar, o mais provável é que o servidor X não é o certo para esta placa de vídeo.

Quando um servidor X não funciona com sua placa, é geramente óbvio. Ou o servidor X não inicia, ou a tela estará incorreta.

Quando a estação de trabalho está pronta para iniciar o servidor X, ela chama o script `startx`, o qual inicia o servidor X localmente na estação de trabalho, com a opção `-query` apontando para um servidor, onde um gerenciador de sessão, como o **XDM**, **GDM** ou **KDM** está executando.

Porque o servidor X é iniciado pelo script `startx`, que é iniciado pelo programa `init`, quando ele falha, o `init` tentará executá-lo novamente. O `init` continuará este laço de tentar executar o servidor X 10 vezes, então desiste, porque pensa que está 'respawning' rapidamente. Depois que finalmente desiste, a mensagem de erro do servidor X deve ser deixada na tela.

Esperar o servidor X falhar 10 vezes pode ser irritante, assim uma maneira simples de evitar as falhas repetidas é iniciar a estação de trabalho no runlevel 3, de modo que o servidor X não seja iniciado automaticamente. Ao invés disto, quando você iniciar a estação de trabalho, começará com um shell `bash`. Do shell `bash`, você pode iniciar o servidor X manualmente com o seguinte comando:

```
sh /tmp/start_ws
```

O servidor X tentará iniciar, então quando ele falhar, irá retornar ao shell `bash`, de forma que você poderá ver a razão da falha.

7.6. Solucionando problemas no gerenciador de telas

O gerenciador de telas é o daemon que executa no servidor, esperando que um servidor X o contate. Uma vez que o contato é feito, ele irá mostrar uma tela de login na tela, oferecendo ao usuário a chance de logar no servidor.

Os três gerenciadores de tela mais comuns são:

- **XDM** – Existir para sempre. Este está incluído no X Windows System padrão.
- **GDM** – O 'Gnome Display Manager'. Este é parte do pacote GNOME.
- **KDM** – O 'KDE Display Manager'. Este é parte do K Desktop System.

As distribuições GNU/Linux mais recentes incluem todos os três gerenciadores de tela.

7.6.1. Tela cinza com um grande cursor em forma de X

Isto indica que o servidor X está em execução, mas ele não conseguiu fazer contato com o gerenciador de telas. Algumas das possíveis razões para isto são:

1. O gerenciador de janelas não está sendo executado

Nas versões recentes do Red Hat (7.0 e superiores), o gerenciador de telas é iniciado através do **init**. No arquivo `/etc/inittab`, há uma linha que se parece com esta:

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

O script **prefdm** irá determinar qual gerenciador de telas executar.

O gerenciador de telas padrão depende de quais pacotes foram instalados. Se o GNOME está instalado, então o GDM é o gerenciador de telas padrão. Se o GNOME não estiver instalado, então o script `prefdm` irá checar para ver se o KDE está instalado. Se ele estiver, então o KDM será o gerenciador de telas padrão. Se o KDE também não estiver instalado, então o XDM será o gerenciador de telas padrão.

Usando o comando **netstat**, você deverá conseguir ver se há um gerenciador de telas em execução. No servidor, execute o comando abaixo:

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

Você deverá ver resultados mostrando que há um processo ouvindo na porta xdmcp (177).

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

Isto mostra claramente que o **gdm** está sendo executado com o PID 1493, e está ouvindo na porta xdmcp.

Se você vir uma linha como esta mostrada acima, indicando que definitivamente há um gerenciador de telas ouvindo, então você precisa certificar-se que a estação de trabalho está enviando a consulta XDMCP para o servidor correto.

No arquivo `lts.conf`, você pode ter uma entrada que especifica o endereço IP do servidor que está executando o gerenciador de telas. A entrada é opcional, mas se presente, deve parecer-se com isto:

```
XDM_SERVER = 192.168.0.254
```

Claro que, o endereço IP para a sua rede pode ser diferente do exemplo acima.

Se a entrada 'XDM_SERVER' não estiver presente, esta usará então o valor da entrada 'SERVER', se presente. Se esta não estiver presente, então ela usará **192.168.0.254**.

A qual mesmo que especificada, você apenas precisa certificar-se que o endereço IP é realmente o endereço correto do servidor executando o gerenciador de telas.

2. O gerenciador de telas pode ser configurado para ignorar requisições de hosts remotos.

Se você tiver determinado que o gerenciador de telas está em execução, então é possível que ele fora configurado para ignorar requisições XDMCP de hosts remotos. Você precisará checar os arquivos de configuração do gerenciador de telas particular em execução para determinar se ele está configurado corretamente.

◆ XDM

A configuração padrão para o Red Hat dá desativar a habilidade das estações de trabalho receberem acesso a uma sessão XDM. O script **Ltsp_initialize** tomará cuidado de permitir isto para você, mas se não estiver funcionando, você deve verificar o arquivo `/etc/X11/xdm/xdm-config`. Procure uma entrada que pareça com esta:

```
DisplayManager.requestPort: 0
```

Esta entrada DEVE estar comentada para que o XDM ouça a requisições remotas na porta 1777.

Outro arquivo de configuração também é importante para que o XDM sirva a requisições remotas de login. É um arquivo chamado `/etc/X11/xdm/Xaccess` que DEVE conter uma linha que inicia com um asterisco '*'. A linha é normalmente incluída no arquivo, mas a Red Hat a mantém comentada. O script `ltsp_initialize` irá corrigir a linha para você, mas se o XDM não funcionar, você deve verificar este arquivo. Uma linha válida deve parecer-se com a abaixo:

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

◆ KDM

Novas versões do KDM possuem um arquivo chamado `kdmrc`. Diferentes distribuições Linux mantém este arquivo em locais diferentes. Para o Red Hat 7.2, é `/etc/kde/kdm/kdmrc`. Para as outras distribuições, você deve executar o comando `locate` para localizar onde o arquivo é mantido.

A entrada que controla se estações de trabalho remotas poderão receber a tela de login é a seção `[Xdmcp]`. Certifique-se que a entrada `Enable` está configurada para `true`.

Versões antigas do KDM usam o arquivo de configuração do XDM, localizado em `/etc/X11/xdm`.

◆ GDM

O GDM usa um conjunto de diferentes arquivos de configuração. Eles estão localizados no diretório `/etc/X11/gdm`.

O principal a ser examinado é o arquivo `gdm.conf`. Procure pela seção `[xdmcp]`. Você deve ver uma entrada dentro desta seção chamada 'Enable'. Ela deve estar configurada como '1' ou 'true', dependendo da versão do GDM. Aqui está um exemplo:

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

Note a linha 'Enable=true'. Versões antigas do GDM usam '0' e '1' para ativar ou desativar XDMCP remoto. Novas versões usam 'false' e 'true'.

3. Se o gerenciador de telas estiver definitivamente em execução, e este estiver ouvindo a requisições de estações de trabalho remotas, pode ser o simples problema no qual o gerenciador de telas não consegue mapear o endereço IP para um nome de host. A estação de trabalho precisa estar listada no arquivo `/etc/hosts`, ou esta precisa estar configurada corretamente no DNS.

Capítulo 8. Kernels

Existem algumas decisões que deve ser tomadas sobre o kernel que irá rodar na estação. Você precisa decidir se você quer usar um dos kernels padrões disponíveis para download, ou criar o seu próprio. E, você precisa decidir se você quer mostrar uma tela gráfica, com uma barra de progresso, isso é possível com o **Linux Progress Patch (LPP)**.

8.1. Kernels padrões fornecidos pelo LTSP

O pacote do kernel que é disponível com o LTSP atualmente inclui dois kernels. Um tem o Patch para mostrar a barra de progresso já aplicado e configurado, e o outro não tem esse patch aplicado.

Ambos os kernels tem o patch para Swap por NFS aplicado.

8.2. Construa seu próprio kernel

Existe duas maneiras de configurar um kernel para o LTSP. O método padrão é usar algo chamado 'Initial Ram Disk' (Disco de Ram Inicial), ou **initrd** para simplificar. A imagem **initrd** é um pequeno sistema de arquivos que é adicionado ao kernel. A imagem do sistema de arquivos **initrd** é carregada na memória, e uma vez que o kernel é carregado isso irá montar o ramdisk com seu sistema de arquivos raiz. Existem duas vantagens de se usar uma imagem **initrd**. Primeiro, nós poderemos compilar os drivers de rede como módulos e carregar o módulo correto durante o boot. Isso permite a um simples kernel suportar virtualmente todas as placas de rede. A outra vantagem é que nós podemos rodar um cliente DHCP como programa de usuário melhor que no espaço do kernel. Rodar o cliente como um programa de usuário permite um melhor controle sobre as opções solicitadas e recebidas do servidor. Também, resulta um kernel ligeiramente menor. A outra maneira para configurar o kernel é sem o **initrd**. Construir o kernel sem o **initrd** requer que um específico driver para placa de rede seja linkada dentro do kernel, e isso também requer uma autoconfiguração do IP e "O sistema de arquivos Root em NFS" sejam setados ao se construir o kernel. A vantagem de não usar **initrd** é que o kernel é ligeiramente menor, e carregará mais rápido. uma vez que a estação de trabalho estará rodando, não há virtualmente nenhuma diferença em como a estação de trabalho funciona.

O kernel padrão do LTSP inclui um Ramdisk inicial (**initrd**) que tem o cuidado de detectar a placa de rede, e faz solicitação DHCP em área de usuário. O objetivo principal para a imagem era de fazê-la o menor possível. Então, nós escolhemos a biblioteca de substituição **uClinux libc**, e o **busybox** para a que precisaríamos durante o boot.

Se você quer criar seu próprio kernel, você deve fazer o download do pacote **ltsp_initrd_kit**. Isso contém a hierarquia do sistema de arquivos raiz e o script para criar a imagem.

8.2.1. Obtendo os fontes do kernel

Quando criamos um kernel customizado, é comum e uma boa idéia iniciar com os fontes do kernel atualizados direto do site **ftp.kernel.org**. A razão para isso é que as distribuições como RedHat, aplicam muitos patches aos seus fontes do kernel, deixando você com uma série de código fonte que realente não estão no kernel oficial.

Faça o download do pacote de fontes do kernel de sua escolha, e salve no diretório `/usr/src`. O kernel é localizado no diretório `/pub/linux/kernel` no servidor **ftp.kernel.org**. Você precisa pegar uma versão recente da série 2.4.x, porque você precisa incluir suporte a **devfs**.

Também, se você quer incluir suporte a swap via nfs ou o Patch para a barra de progresso no linux (Linux Progress Patch (LPP)), você precisa ter certeza de pegar os patches e os fontes do kernel que sejam compatíveis. No momento em que escrevo isso, o kernel 2.4.9 é o último que suporta essas características.

Para nosso exemplo, nós iremos usar o kernel 2.4.9. O caminho completo é

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

Descompacte os fontes do kernel no diretório `/usr/src`. Você precisa ter cuidado, porque quando você descompactar o pacote, será criado um diretório chamado `linux`. Você pode já ter um diretório chamado `linux` com um diferente código fonte, e você não deve sobrescrevê-lo. Então verifique a existência desse diretório, e se ele existir, renomeie-o para outro nome antes de descompactar seus fontes.

O pacote de fontes será baixado comprimido com o utilitário de compactação **bzip2**. Então nós precisamos descompactá-lo antes de usar o programa **tar**. Você pode usar o seguinte comando para descompactá-lo:

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

Quando a descompactação terminar, você terá um diretório chamado `linux` contendo a árvore de fontes completa. Nesse ponto, eu normalmente gosto de renomear o diretório para algo mais significativo.

```
mv linux linux-2.4.9
```

Uma vez que o diretório foi renomeado, mudaremos para dentro do novo diretório:

```
cd linux-2.4.9
```

Eu normalmente gosto de modificar o `Makefile` antes de começar a configuração do novo kernel. Perto do início do arquivo existe uma variável chamada **EXTRAVERSION**. Eu seto esta para `'ltsp-1'`, então o número da versão atual do kernel será `'2.4.9-ltsp-1'`, que tornará fácil a identificação do kernel posteriormente. O topo do `Makefile` deverá ficar parecido com isso:

```
VERSION = 2
PATCHLEVEL = 4
SUBLEVEL = 9
EXTRAVERSION = -ltsp-1
KERNELRELEASE=$(VERSION) . $(PATCHLEVEL) . $(SUBLEVEL) $(EXTRAVERSION)
```

8.2.2. Patches do Kernel

Após descompactar o kernel, você pode ter vários patches que deseja aplicar. Por exemplo, o patch para Swap por NFS ou o Patch para barra de progresso do Linux. Esses patches DEVEM ser aplicados antes de configurar o kernel.

8.2.2.1. Patch para Swap por NFS

O patch para Swap por NFS irá possibilitar ao kernel da estação de trabalho usar um arquivo de swap localizado no servidor NFS. Enquanto é normalmente recomendado ter bastante memória na estação para não se fazer necessário swap, algumas vezes pode ser difícil se adicionar mais memória, especialmente em computadores antigos. Então, essa característica de fazer swap via NFS pode fazer um computador sem utilidade realmente usável.

Se o diretório corrente é /usr/src/linux-2.4.9, e o patch está em /usr/src, você pode fazer o seguinte para testar o patch:

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

Isso irá testar o patch, para ter certeza que pode ser aplicado. Se isso terminar sem erros, então você pode aplicar o patch sem a opção **--dry-run**.

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

8.2.2.2. Patch de Progresso do Linux (Linux Progress Patch – LPP)

O patch de progresso do linux (LPP) irá permitir que você configure um logo gráfico para mostrar durante o processo de boot. As mensagens normais de kernel no boot serão redirecionadas para outra tela tty, e instruções especiais serão adicionadas nos scripts de boot que origina uma barra de progresso para refletir quanto do processo de boot já foi feito.

Como o patch para Swap NFS, você pode testar o patch LPP com o comando:

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

Se o teste completou com sucesso, então você pode aplicar o patch com:

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

8.2.3. Opções de configuração do Kernel

Você pode agora executar o programa de configuração do kernel de sua escolha. Opções disponíveis são:

- make xconfig

Isso irá chamar a versão X Windows do utilitário de configuração do kernel.

- make menuconfig

Isso irá chamar a versão baseada em curses do utilitário de configuração do kernel.

- make config

Isso irá chamar a versão baseada em perguntas do utilitário de configuração do kernel.

8.2.3.1. Configuração do kernel para uso com o initrd

Configurando o kernel para uso com o initrd requer que as seguintes opções sejam setadas:

- File systems → /dev filesystem support

Sistema de arquivos /dev deve estar habilitado. Isso é selecionado na seção 'File systems'. Não especifique 'Automatically mount at boot'. A montagem deve ser efetuada pelo script /linuxrc

- Block devices → RAM disk support

Estações LTSP requerem que o kernel suporte um disco RAM (RAM disk). Isso é setado na seção 'Block devices'

- Block devices → Initial RAM disk (initrd) support

Isso também deve ser habilitado.

- Processor type and features → Processor family

Você precisa ter certeza de que o kernel que você está construindo pode executar na CPU da estação. Isso é feito na seção 'Processor type and features'. Você deve também desligar o suporte a SMP a menos que você tenha realmente múltiplas CPUs.

- File systems → Network file systems → NFS Client support

A estação estará montando seu sistema de arquivos raiz via NFS, então o suporte a cliente NFS é necessário.

Deve-se tomar cuidado com as opções requeridas. Você pode também desligar várias características do kernel, para reduzir o tamanho do kernel.

8.2.3.2. Configuração do kernel para uso sem initrd

Configuração do kernel para uso sem initrd defere de um kernel com initrd em algumas maneiras:

- Block devices → RAM disk support

Estações LTSP precisam que o kernel tenha suporte a Ram Disk.

- Block devices → Initial RAM disk (initrd) support

Isso precisa ser desabilitado.

- Networking options → IP:kernel level autoconfiguration

Isso precisa ser habilitado. Isso irá instruir o kernel para configurar automaticamente a interface de rede eth0, baseado nos valores passados na linha de comando do kernel.

Não é necessário especificar opções de DHCP, BOOTP ou RARP porque a imagem de boot Etherboot já tem feito uma requisição de DHCP ou BOOTP, e o fará com os parâmetros de IP disponíveis na linha de comando do kernel. Isso dispensa o kernel do problema de fazer sua própria requisição.

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

Quando não é usado o initrd, você deve escolher o driver da placa de rede específico que combine com sua placa de rede. Isso DEVE ser linkado de forma estática no kernel, porque a interface de rede é necessária antes de montar o sistema de arquivos raiz. Essa é a principal diferença para o kernel com initrd.

- File systems → /dev filesystem support

Com o LTSP versão 2.09pre2, o suporte a **devfs** é necessário. Isso é verdadeiro independente se `initrd` é usado ou não

- File systems → Automatically mount at boot

Ao NÃO usar `initrd`, o sistema de arquivos `/dev` deve ser montado pelo kernel, durante o processo de boot. Então, diga 'Y' aqui.

- File systems → Network file systems → NFS Client support

A Estação irá montar o sistema de arquivos raiz via NFS, então o suporte a cliente NFS é necessário.

8.2.4. Construindo o kernel

Para tornar as coisas mais fáceis, uma cópia do arquivo `.config` é incluído no pacote `ltsp_initrd_kit`. Você pode copiar isso para o diretório `/usr/src/linux-2.4.9`

Uma vez que você terminou de selecionar ou de–selecionar as opções do kernel, você precisa gerar o kernel. Os seguintes comandos devem ser executados para gerar o kernel:

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

Você pode agregá–los todos juntos como abaixo:

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

O E comercial duplo (&) significa que se o primeiro comando terminar com sucesso, então o segundo comando será executado. Se o segundo comando terminar com sucesso, então o terceiro comando será executado, e assim sucessivamente.

Quando a compilação do kernel finalizar, o novo kernel estará situado em `/usr/src/linux-2.4.9/arch/i386/boot/bzImage`.

8.2.5. Marcando o Kernel para Etherboot

Para Etherboot carregar o kernel linux, é necessário que seja preparado. Isso é chamado 'Marcar' o kernel. Este processo adicionará código extra ao kernel que é executado antes do controle ser passado para o kernel. A ferramenta para marcar o kernel é chamada '**mknbi–linux**'.

O `ltsp_initrd_kit` inclui um script shell chamado **buildk** que inclui todos os comandos que você precisa para preparar a imagem do kernel para boot por rede.

Capítulo 9. Entradas em Its.conf

Quando projetamos o LTSP, um dos problemas que sabíamos que teríamos que tratar era com a variedade de configurações de hardware das estações. Certamente, qualquer que seja a combinação de processador, placa de rede e vídeo disponíveis hoje, poderiam não estariam disponíveis em três meses, então queremos adicionar mais estações à rede.

Então, planejamos um modo de especificar as configurações de cada estação. O arquivo de configuração é chamado de `lts.conf` e encontra-se no diretório `/opt/ltsp/i386/etc`.

O formato do `lts.conf` permite configurações 'padrões' e configurações individuais por estação. Se todas as suas estações são idênticas, você poderia especificar todos os seus parâmetros de configuração na sessão '[Default]'.

9.1. Arquivo exemplo lts.conf

Este é um exemplo do arquivo `lts.conf`:

```
[Default]
SERVER           = 192.168.0.254
X_MOUSE_PROTOCOL = "PS/2"
X_MOUSE_DEVICE   = "/dev/psaux"
X_MOUSE_RESOLUTION = 400
X_MOUSE_BUTTONS  = 3
USE_XFS          = N
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. Parâmetros disponíveis no arquivo lts.conf

9.2.1. Parâmetros gerais

Comentários

Comentários iniciam com o sinal '#' e continuam até o final da linha.

LTSP_BASEDIR

Isso indica onde o sistema de arquivos raiz do LTSP está localizado. O valor padrão é `/opt/ltsp`

SERVER

Isso é o servidor que é usado para XDM_SERVER, TELNET_HOST, XFS_SERVER e SYSLOG_HOST, se qualquer um deles não for especificado explicitamente. Se você tem uma máquina que está atuando como um servidor para tudo, então você pode apenas especificar o endereço aqui e omitir os outros parâmetros de servidor. Se este valor não for setado, **192.168.0.254** será usado.

SYSLOG_HOST

Se você quer enviar as mensagens de log para uma máquina que não seja o servidor padrão, então você pode especificar essa máquina aqui. Se este parâmetro não for especificado, então ele irá usar o parâmetro 'SERVER' descrito acima.

NFS_SERVER

Isso especifica o endereço IP do servidor NFS usado quando o sistema de arquivos /home é montado. O padrão é o que quer que tenha sido setado no parâmetro **SERVER**

USE_NFS_SWAP

Configure isso para **Y** se você quer ligar o swap por NFS. O Padrão é **N**

SWAPFILE_SIZE

Isso é como você pode controlar o tamanho o arquivo de swap. O padrão é **64m**.

SWAP_SERVER

O arquivo de swap pode existir em qualquer servidor da rede que tenha capacidade de manipulá-lo. Você pode especificar o endereço IP desse servidor. O padrão é o que quer que tenha sido setado no parâmetro NFS_SERVER.

NFS_SWAPDIR

O diretório no servidor que é exportado por NFS. O padrão é /var/opt/ltsp/swapfiles. Tenha certeza que o diretório está sendo exportado no arquivo /etc/exports.

TELNET_HOST

Se a estação está configurada para ter uma interface baseada em caracteres, então o valor desse parâmetro será usado como o host para o telnet. Se esse valor não for setado, então será usado o valor do parâmetro **SERVER** acima.

DNS_SERVER

Usado para gerar o arquivo resolv.conf.

SEARCH_DOMAIN

Usado para gerar o arquivo resolv.conf.

SCREEN_01 até SCREEN_12

Até 12 scripts de tela pode ser configurado por estação. Isso irá permitir até 12 sessões na estação que serão acessíveis pressionando-se as teclas Ctrl-Alt-F1 até Ctrl-Alt-F12

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

Atualmente os valores possíveis incluem:

- ◆ startx

- ◆ telnet
- ◆ rdesktop
- ◆ shell

Olhe no diretório `/opt/ltsp/i386/etc/screen.d` para mais scripts de tela, ou escreva seus próprios, e coloque-os lá.

MODULE_01 até MODULE_10

Até 10 módulos do kernel podem ser carregados usando essas entradas de configuração. A linha de comando que você usaria executando o `insmod` pode ser especificada aqui. Por exemplo:

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

Se o valor desse parâmetro é o caminho absoluto, então **insmod** será usado para carregar esse módulo. Senão, **modprobe** será usado.

RAMDISK_SIZE

Quando a estação inicializa, isso criará um ramdisk e montará no diretório `/tmp`. Você pode controlar o tamanho do sistema de arquivos com esse parâmetro. Especifique isso em unidade de kbytes (1024 bytes). Para criar um ramdisk de 1 megabyte, especifique **RAMDISK_SIZE = 1024**

Se você modificar o tamanho do ramdisk aqui, você precisará mudar também o tamanho do ramdisk dentro do kernel. Isto pode ser compilado dentro, ou se você está usando Etherboot ou Netboot, você dirá para o kernel o tamanho do ramdisk quando você rotular o kernel com `mknbi-linux`.

O valor padrão para isso é 1024 (1 mb)

RCFILE_01 thru RCFILE_10

Scripts RC adicionais podem ser executados pelo script `rc.local`. Apenas coloque o script no diretório `/etc/rc.d`, e especifique o nome do script em uma dessas entradas.

SOUND

Se o pacote de som LTSP está instalado, você precisa configurar essa entrada para **Y** e isso irá executar o script **rc.sound** para configurar a placa de som e o daemon. O padrão é **N**.

9.2.2. Parâmetros X-Windows

XDM_SERVER

Se você quer apontar o XDM para uma máquina diferente do servidor padrão, então você deve especificar o servidor aqui. Se o parâmetro não for especificado, então será usado o parâmetro `SERVER` descrito acima.

XSERVER

Isso define qual servidor X a estação irá executar. Para placas de video PCI e AGP, este parâmetro não é necessário. O script `rc.local` poderá auto detectar a placa. Você pode também configurar esse valor para **auto** para indicar que ele irá tentar auto detectar a placa.

Para placas de vídeo ISA, ou para forçar um servidor X específico, você pode especificar o nome do driver ou Xserver nesse entrada.

Se o valor iniciar com **XF86_**, então o XFree86 3.3.6 será usado. Senão, X.org 6.7.0 será usado. O valor padrão para isso é **auto**.

X_MODE_0 through X_MODE_2

Até 3 Modelins ou resoluções podem ser configuradas para a estação. Esta entrada pode ter dois diferentes tipos de valores. Podendo ser uma resolução ou um completo modeline.

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

Se nenhuma das entradas **X_MODE_x** forem especificadas, então serão usadas as geradas em modelines, e as resoluções de 1024x768, 800x600 and 640x480.

Se uma ou mais entradas **X_MODE_x** forem especificadas, elas cancelarão completamente qualquer uma gerada em modelines.

X_MOUSE_PROTOCOL

Qualquer valor que funcione no protocolo de ponteiros do X.org pode ser colocado aqui. Valores típicos incluem "Microsoft" e "PS/2". O valor padrão para isso é "**PS/2**".

X_MOUSE_DEVICE

Isto é o dispositivo onde o mouse está conectado. Se é um mouse serial, isto será uma porta serial, como **/dev/ttyS0** ou **/dev/ttyS1**. Se é um mouse PS/2, o valor será **/dev/psaux**. O valor padrão para isso é **/dev/psaux**.

X_MOUSE_RESOLUTION

Este é o valor 'Resolution' no arquivo **XF86Config**. O valor típico para um mouse serial é **50** e o valor típico para um mouse PS/2 é **400**. O valor padrão para isso é **400**.

X_BUTTONS

Isso informa ao sistema quantos botões o mouse possui. Normalmente é setado para **2** ou **3**. O valor padrão para isso é **3**.

X_MOUSE_EMULATE3BTN

Isso informa ao servidor X para copiar um mouse de 3 botões aceitando um clique de ambos, os botões da esquerda e da direita simultaneamente. O valor padrão é **N**.

X_MOUSE_BAUD

Para um mouse serial, isso define a taxa de baud. O valor padrão para isso é **1200**.

X_COLOR_DEPTH

Isto é o número de bits usados para a profundidade de cores. Valores possíveis são **8, 15, 16, 24** e **32**. 8 bits teremos 256 cores, 16 teremos 65536 cores, 24 teremos 16 milhões de cores e 32 bits teremos 4.2 bilhões de cores! Nem todos os servidores X suportam todos esses valores. O Valor padrão para isso é **16**

USE_XFS

Você tem a escolha de executar um servidor de fontes X (XFS) ou ler as fontes através de um sistema

de arquivos NFS. O servidor de fontes deve oferecer uma maneira simples de manter todas as fontes em um lugar, mas existem alguns problemas quando o número de estações cresce aproximadamente para mais de 40. Os 2 valores para essa opção são: **Y** e **N**. Se você quer utilizar um servidor de fontes, então você deve usar a entrada **XFS_SERVER** para especificar qual host irá atuar como um servidor de fontes.

XFS_SERVER

Se você está usando um servidor de fontes X para fornecer fontes, então você pode usar essa entrada para especificar o endereço IP do host que está atuando como um servidor de fontes. Se isso não for especificado, será usado o servidor padrão, que está especificado na entrada **SERVER** descrita acima.

X_HORZSYNC

Isto configura o parâmetro de configuração **HorizSync** do X.org. O padrão é "**31-62**".

X_VERTREFRESH

Isto configura o parâmetro de configuração **VertRefresh** do X.org. O padrão é "**55-90**".

XF86CONFIG_FILE

Se você quer criar seu próprio arquivo XF86Config completo você pode fazer isso colocando-o no diretório **/opt/ltsp/i386/etc**. Então, o que quer que você decida chamar é necessário ser colocado como valor para essa variável de configuração. Por exemplo:

```
XF86CONFIG_FILE = XF86Config.ws004
```

9.2.3. Parâmetros para telas de toque (Touch screen)

USE_TOUCH

Se você está conectando uma touch screen na estação, você pode habilitá-la configurando esta entrada para **Y**. Se habilitada, as entradas de configurações adicionais irão configurar os aspectos específicos da touch screen. O valor padrão é **N**.

X_TOUCH_DEVICE

Uma touch screen funciona como um mouse e normalmente é interfaceada com a estação através de uma porta serial. Você pode especificar qual porta serial com essa entrada. Por exemplo, você pode configurá-la igual a **/dev/ttyS0**. Não há valor padrão para essa entrada.

X_TOUCH_MINX

Entrada de calibração para um touch screen EloTouch. Padrão é **433**.

X_TOUCH_MAXX

Entrada de calibração para um touch screen EloTouch. Padrão é **3588**.

X_TOUCH_MINY

Entrada de calibração para um touch screen EloTouch. Padrão é **569**.

X_TOUCH_MAXY

Entrada de calibração para um touch screen EloTouch. Padrão é **3526**.

X_TOUCH_UNDELAY

Entrada de calibração para um touch screen EloTouch. Padrão é **10**.

X_TOUCH_RPTDELAY

Entrada de calibração para um touch screen EloTouch. Padrão é **10**.

9.2.4. Parâmetros para aplicações locais

LOCAL_APPS

Se você quer a capacidade de executar aplicações localmente na estação, configure esta variável para **Y**. Diversos passos adicionais devem ser feitos no servidor para habilitar aplicações locais. Verifique a seção 'Local Apps' no manual do LTSP para mais informações. O valor padrão é **N**.

NIS_DOMAIN

Se você configurou *LOCAL_APPS*, então você deve ter um servidor NIS na rede. A entrada *NIS_DOMAIN* é onde você especifica o nome do domínio NIS. É necessário combinar o nome do domínio com o que foi definido no servidor NIS. Isso não é o mesmo que um domínio Internet. O valor padrão é **ltsp**.

NIS_SERVER

Configure isso com o endereço IP de seu servidor NIS se você não quiser enviar um broadcast que irá localizar o servidor NIS.

9.2.5. Parâmetros de teclado

Todos os arquivos de suporte de teclado são copiados dentro da hierarquia */opt/ltsp/i386*, para configurar suporte a teclado internacional é simplesmente uma questão de configurar o *X.org*. Existem vários parâmetros de configuração para isso.

Os valores para os parâmetros abaixo são da documentação do *X.org*. O que é válido para o *X.org* é válido para estes parâmetros.

Gostaríamos de adicionar documentação para mostrar quais valores são necessários para cada tipo de teclado internacional. Se você trabalha com isso e quer configurar seu teclado internacional, dê um retorno para o grupo de desenvolvimento do LTSP que será muito valorizado.

XkbTypes

O valor padrão para isso é a palavra **'default'**.

XkbCompat

O valor padrão para isso é a palavra **'default'**.

XkbSymbols

O valor padrão para isso é **'us(pc101)'**.

XkbModel

O valor padrão para isso é **'pc101'**.

XkbLayout

O valor padrão para isso é **'us'**.

9.2.6. Parâmetros de configuração de impressoras

Até três impressoras podem ser conectadas a uma estação diskless. A combinação de impressoras seriais e paralelas pode ser configurada através das seguintes entradas no arquivo **lts.conf**:

PRINTER_0_DEVICE

O nome do primeiro dispositivo de impressão. Nomes como **/dev/lp0**, **/dev/ttyS0** ou **/dev/ttyS1** são permitidos.

PRINTER_0_TYPE

O tipo de impressora. Opções válidas são **'P'** para Paralela, e **'S'** para Serial.

PRINTER_0_PORT

A número de porta TCP/IP a ser usado. Por padrão, será usado **'9100'**

PRINTER_0_SPEED

Se a impressora é serial, essa é a configuração para selecionar a velocidade (baud rate). Por padrão, **'9600'** será usado.

PRINTER_0_FLOWCTRL

Para impressoras seriais, o controle de fluxo pode ser especificado. Use **'S'** para controle de fluxo por software (XON/XOFF) ou **'H'** para controle de fluxo por Hardware (CTS/RTS). Se nenhum deles for especificado, **'S'** será usado.

PRINTER_0_PARITY

Para impressoras seriais, a Paridade pode ser especificada. As opções são: **'E'**–Even, **'O'**–Odd ou **'N'**–None. Se não for especificada, **'N'** será usado.

PRINTER_0_DATABITS

Para impressoras seriais, o número de data bits pode ser especificado. As opções são: **'5'**, **'6'**, **'7'** e **'8'**. Se não for especificado, **'8'** será usado.

PRINTER_1_DEVICE

Nome do segundo dispositivo de impressão

PRINTER_1_TYPE

Tipo do segundo dispositivo de impressão

PRINTER_1_PORT

Porta TCP/IP do segundo dispositivo de impressão

PRINTER_1_SPEED

Velocidade (baud rate) da segunda impressora (serial)

PRINTER_1_FLOWCTRL

Controle de fluxo da segunda impressora (serial)

PRINTER_1_PARITY

Paridade da segunda impressora (serial)

PRINTER_1_DATABITS

Bits de dados da segunda impressora (serial)

PRINTER_2_DEVICE

Nome da terceira impressora

PRINTER_2_TYPE

Tipo da terceira impressora

PRINTER_2_PORT

Porta TCP/IP da terceira impressora

PRINTER_2_SPEED

Velocidade (baud rate) da terceira impressora (serial)

PRINTER_2_FLOWCTRL

Controle de fluxo da terceira impressora (serial)

PRINTER_2_PARITY

Paridade da terceira impressora (serial)

PRINTER_2_DATABITS

Bits de dados da terceira impressora (serial)

Capítulo 10. Aplicações Locais

Em um ambiente LTSP, você tem a opção de rodar as aplicações localmente na estação, ou remotamente no servidor.

De longe, a maneira mais fácil de configurar um ambiente LTSP é rodar as aplicações no servidor. Isto é, a aplicação cliente roda no servidor usando CPU e memória do servidor, enquanto é mostrado a saída de tela na estação e usando o teclado e mouse da estação.

Isso é uma capacidade fundamental do X Windows. A estação funciona como um terminal X Windows padrão.

Para que um usuário execute uma aplicação na estação, esta precisa de algumas informações sobre o usuário. Informações sobre o seguinte:

- Id de usuário
- Grupo primário que o usuário pertence
- O diretório home do usuário

LTSP confia no Network Information Service NIS, (chamado anteriormente de *Yellow Pages*) para tornar as informações de usuários e grupos disponíveis nas estações.

10.1. Benefícios da execução de aplicações localmente

Há benefícios da execução de aplicações na estação.

- Reduz a carga do servidor. Em grandes redes com o uso intensivo de memória, tais como Netscape, rodando a aplicação na estação é possível obter um desempenho melhor, quando a estação é potente o suficiente para fazê-lo
- Aplicações em execução não afetarão outros usuários.
- O suporte de som é muito fácil de configurar quando a aplicação que tocará o som esta rodando na estação.

10.2. Questões com a configuração do suporte de aplicações locais

Configurar a capacidade de executar aplicações localmente necessita de muito mais.

- Uma maior demanda na estação. Isso necessita de mais RAM e de uma CPU mais poderosa. É um ponto de partida muito bom 64mb de ram na estação.
- NIS – Para executar aplicações na estação, você primeiramente deve indentificar você mesmo para a estação. Isso é, a estação precisar saber quem você é. Isso necessita de alguma forma de autenticação por senha. NIS foi escolhido como o método de autenticar os usuários na rede.
- Diretórios adicionais precisam ser exportados da estação para serem montados via NFS.
- A partida das aplicações é mais lenta, porque necessitam ser lidas via NFS, causando um aumento da atividade de rede. Também, porque cada cópia do programa é executada em sua própria CPU, você não terá a vantagem da capacidade do Linux/Unix de compartilhar segmentos de código entre múltiplas instâncias do mesmo programa, que reduziria o tempo para segundos e execuções sucessivas do programa.

10.3. Configuração do servidor para aplicações locais

10.3.1. Entradas no `lts.conf`

Algumas entradas devem ser configuradas no arquivo `lts.conf`:

LOCAL_APPS

Isto deve ser setado para **Y**. Fará com que os seguintes passos aconteçam durante a processo de boot da estação.

1. O diretório `/home` no servidor será montado via NFS.
2. O `/var/yp/nicknames` será criado na estação.
3. O **portmapper** será iniciado na estação.
4. **xinetd** será iniciado na estação.
5. O arquivo `/etc/yp.conf` será criado na estação.
6. O comando **domainname** será executado com o valor da entrada **NIS_DOMAIN** do arquivo `lts.conf`.
7. O **ypbind** irá executar na estação.

NIS_DOMAIN

Com NIS, todos os pontos na rede que querem ser associados a um servidor NIS específico precisam pertencer ao mesmo domínio NIS (isso não é relacionado de forma alguma a domínio DNS). Você usa a entrada **NIS_DOMAIN** para especificar o nome do domínio NIS que a estação irá pertencer.

NIS_SERVER

NIS irá tentar conectar com um servidor NIS específico, ou irá enviar um broadcast para a rede, procurando por um servidor. Se você quer escolher um servidor específico, então entre com o endereço IP do servidor na entrada **NIS_SERVER**.

10.3.2. Network Information Service – NIS

NIS é um tipo de serviço Cliente/Servidor. No servidor, há um daemon rodando, que irá aceitar requisições de clientes (estações de trabalho). O daemon no servidor é chamado **ypserv**.

Na estação, existe um processo chamado **ypbind**. Quando a estação precisa localizar uma informação sobre um usuário, tais como a verificação de uma senha ou localização do diretório home de um usuário, usará o **ypbind** para estabelecer uma conexão com o **ypserv**, no servidor.

Se você já está rodando NIS em seu ambiente de rede, então não será necessário configurar o servidor LTSP para rodar **ypserv**. Você pode apenas configurar as entradas **NIS_DOMAINNAME** e **NIS_SERVER** no `lts.conf` para combinar com seu esquema NIS atual.

Se você não está rodando NIS em sua rede, então você deverá configurar o servidor para rodar o **ypserv**.

Para uma completa informação na configuração de um servidor NIS, consulte o documento HOWTO no site LDP chamado *The Linux NIS(YP)/NYS/NIS+ HOWTO*. Consulte a lista de outras fontes de informação no final deste documento.

10.4. Configuração de Aplicações

Para configurar uma aplicação para executar na estação, você precisa colocar todos os componentes da aplicação em um lugar onde a estação possa vê-la.

Com as versões antigas do LTSP (2.08 e anteriores), vários diretórios eram exportados no servidor e montados na estação. Diretórios como `/bin`, `/usr/bin`, `/lib` e `/usr` foram expostos para a estação.

O problema com esse esquema é que isto só funciona se a estação e o servidor forem da mesma arquitetura. Na verdade, mesmo com as diferenças, como o servidor é um Pentium II (i686) e a estação um Pentium Clássico (i586) isso pode ser um problema, porque o servidor irá ter as bibliotecas i686 e não as bibliotecas i386, i486 ou i586.

Então, a maneira mais clara de resolver isto é ter uma árvore completa com todos os binários e bibliotecas que a estação irá precisar, independente das bibliotecas e binários do servidor.

Configurando uma aplicação para execução local requer a colocação de todas as peças necessárias na árvore. Um desses pacotes disponíveis para download no site do LTSP é o pacote do netscape local, que instala muitos dos arquivos no diretório `/opt/ltsp/i386/usr/local/netscape`. Coisas como classes java, arquivos de Help, arquivos binários executáveis e scripts são colocados ali.

Netscape não precisa de quaisquer bibliotecas de sistema adicionais, então não há nada a adicionar no diretório `/opt/ltsp/i386/lib`. Muitas aplicações requerem bibliotecas adicionais.

Então, como você pode determinar quais bibliotecas serão necessárias? Isto é o que o comando **ldd** torna acessível.

Vamos pressupor que você precisa configurar uma certa aplicação para rodar localmente. Escolheremos **gaim** como exemplo. **gaim** é um cliente de Mensagens Instantâneas da AOL, isso irá permitir que você se comunique com outras pessoas nos fóruns da AOL.

A primeira coisa que você vai precisar fazer é localizar o arquivo binário executável do **gaim**. Em um sistema Redhat 7.2, ele está localizado no diretório `/usr/bin`.

Uma vez localizado o binário do **gaim**, você pode executar **ldd** nele:

```
[jam@server /]$ ldd /usr/bin/gaim
libaudiofile.so.0 => /usr/lib/libaudiofile.so.0 (0x40033000)
libm.so.6 => /lib/i686/libm.so.6 (0x40051000)
libnsl.so.1 => /lib/libnsl.so.1 (0x40074000)
libgnomeui.so.32 => /usr/lib/libgnomeui.so.32 (0x4008a000)
libart_lgpl.so.2 => /usr/lib/libart_lgpl.so.2 (0x4015d000)
libgdk_imlib.so.1 => /usr/lib/libgdk_imlib.so.1 (0x4016c000)
libSM.so.6 => /usr/X11R6/lib/libSM.so.6 (0x40191000)
libICE.so.6 => /usr/X11R6/lib/libICE.so.6 (0x4019a000)
libgtk-1.2.so.0 => /usr/lib/libgtk-1.2.so.0 (0x401b1000)
libdl.so.2 => /lib/libdl.so.2 (0x402df000)
libgdk-1.2.so.0 => /usr/lib/libgdk-1.2.so.0 (0x402e3000)
libgmodule-1.2.so.0 => /usr/lib/libgmodule-1.2.so.0 (0x40319000)
libXi.so.6 => /usr/X11R6/lib/libXi.so.6 (0x4031d000)
libXext.so.6 => /usr/X11R6/lib/libXext.so.6 (0x40325000)
libX11.so.6 => /usr/X11R6/lib/libX11.so.6 (0x40333000)
libgnome.so.32 => /usr/lib/libgnome.so.32 (0x40411000)
```

```

libgnomesupport.so.0 => /usr/lib/libgnomesupport.so.0 (0x40429000)
libesd.so.0          => /usr/lib/libesd.so.0 (0x4042e000)
libdb.so.2          => /usr/lib/libdb.so.2 (0x40436000)
libglib-1.2.so.0    => /usr/lib/libglib-1.2.so.0 (0x40444000)
libcrypt.so.1       => /lib/libcrypt.so.1 (0x40468000)
libc.so.6           => /lib/i686/libc.so.6 (0x40495000)
libz.so.1           => /usr/lib/libz.so.1 (0x405d1000)
/lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)

```

A listagem acima mostra todas as bibliotecas que o programa **gaim** é dinamicamente linkado em posição contrária.

A maioria dos programas usam bibliotecas compartilhadas, confie no carregador dinâmico **ld-linux** para localizar e carregar cada uma das bibliotecas compartilhadas. Entretanto, alguns programas, no entanto, carregam as bibliotecas manualmente com chamadas a função **dlopen()**. Para essas aplicações, **ldd** não irá mostrar as bibliotecas. Neste caso, **strace** deve ser usado para rastrear a execução desse programa, e você deverá ver as chamadas ao **dlopen()**, com o nome da biblioteca listada em seus argumentos.

Uma vez que a lista de bibliotecas for coletada, as bibliotecas necessárias deverão ser copiadas para seus locais apropriados na árvore `/opt/ltsp/i386`.

10.5. Executando aplicações locais

No X Windows, os programas funcionam normalmente em relação ao lugar onde o gerenciador de janelas está sendo executado. Isto é, se o gerenciador de janelas está executando no servidor, mostrando sua saída na estação, então todos os programas que serão chamados, serão executados no servidor, enviando sua saída para a estação.

O truque é o servidor dizer para a estação executar o programa. Isso é feito tipicamente com o comando **rsh**.

Aqui está um exemplo de como executar o programa **gaim** na estação:

```

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

```

O exemplo acima pode ser digitado em uma janela **xterm**, ou pode ser colocado em um shell script, e ser executado por um ícone na área de trabalho.

De modo similar é feita a execução local do Netscape, mas uma variável de ambiente adicional deve ser setada antes de executar o programa.

```

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

```

Capítulo 11. Exemplos de Configuração

Quase todos os aspectos de uma estação podem ser configurados com entradas no arquivo `lts.conf`, que geralmente está localizado no diretório `/opt/ltsp/i386/etc`.

11.1. Mouse Serial

Aqui está um exemplo das entradas no arquivo `lts.conf` para um mouse serial padrão de 2 botões:

```
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 com roda

Aqui está um exemplo das entradas no arquivo `lts.conf` para o uso de um 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. Impressora USB em um ThinkNic

A estação ThinkNIC possui uma porta USB, que pode ser usada para conectar uma impressora local. Aqui está um exemplo das entradas necessárias no arquivo `lts.conf`:

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

11.4. Forçando a estação a carregar o Xserver XFree86 3.3.6

Por padrão, X.org 6.7.0 será usado para a estação. Se você quer forçar a estação a usar o antigo Xserver XFree86 3.3.6, você deve primeiramente carregar o pacote Xserver 3.3.6 apropriado. Então, você precisa adicionar a entrada no arquivo `lts.conf`. Este é um exemplo da especificação do **SVGA** Xserver:

```
XSERVER = XF86_SVGA
```

Capítulo 12. Outras fontes de informação

12.1. Referências Online

1. The LTSP home page

www.LTSP.org

2. Diskless–Nodes HOW–TO document for Linux

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

3. Etherboot Home Page

etherboot.sourceforge.net

4. The Rom–O–Matic site

www.Rom–O–Matic.net

5. X.org Mouse Support

www.xfree86.org/current/mouse.html

6. XFree86–Video–Timings–HOWTO

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

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

www.linuxdoc.org/HOWTO/NIS–HOWTO.html

12.2. Publicações Impressas

- 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

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)