

Reutilização de Computadores Obsoletos com a Implementação de um Servidor de Terminais GNU/Linux

GISANE APARECIDA MICHELON¹

GIANCARLO FERNANDO HILD²

PALAVRAS-CHAVE

Terminais Leves, GNU/Linux, Debian, Opensuse, LTSP, Diskless.

RESUMO

A cada dia surgem novos programas e sistemas operacionais que demandam uma maior capacidade de processamento. A reutilização de hardware obsoleto é uma necessidade devido à rápida evolução da informática, às dificuldades de aquisição de equipamentos e da liberação de um item de patrimônio público. Para atender essa necessidade, uma solução seria a utilização de servidores de terminais que transformam computadores antigos em estações de trabalho úteis praticamente sem acrescentar gastos. Essa abordagem torna-se atraente para instituições que possuem um parque de máquinas obsoletas e não dispõe de recursos para investir em novos computadores.

Este artigo aborda o reaproveitamento de computadores com hardware antigo por meio da implementação de um servidor de terminais no sistema operacional GNU/Linux, com LTSP (Linux Terminal Server Project) em um laboratório de informática de uma universidade pública. Além disso, são descritos o processo de instalação e configuração para a montagem do laboratório onde os terminais se conectam a um servidor. Os terminais utilizam boot remoto e se conectam ao servidor por uma rede local. Foram estudados diversos mecanismos para realizar o boot remoto, sendo a alternativa mais efetiva o uso de terminal diskless (sem disco rígido), utilizando uma memória ROM inserida na placa de rede. Testes de desempenho foram realizados com o objetivo de avaliar a viabilidade da implementação do LTSP.

1. INTRODUÇÃO

A evolução rápida da informática causa alguns problemas quando o assunto é reutilização de hardware antigo. Os novos programas e sistemas operacionais exigem um potencial cada vez maior de processamento e de memória, depreciando rapidamente o hardware. Uma solução para este problema é a utilização de servidores de terminais como o LTSP, que podem transformar computadores antigos e sem uso em estações de trabalho úteis com um custo mínimo, necessitando apenas da implementação de um servidor de terminais leves e algumas configurações para iniciar o funcionamento. Este custo mínimo também é devido ao LTSP ser uma aplicação apoiada em software livre com licenças GPL – GNU Public License.

¹ E-mail: gisane@unicentro.br

² E-mail: gian_hild@hotmail.com

O LTSP [Ltsp05] é um projeto voltado para a criação de um servidor de terminais, utilizando o sistema operacional GNU/Linux. Seu objetivo principal é a reutilização de computadores obsoletos ou a utilização de computadores novos e baratos. Esses computadores, denominados estações de trabalho, terminais leves ou terminais remotos, conectam-se ao servidor GNU/Linux com LTSP, utilizando o sistema operacional e os aplicativos instalados no servidor, sendo que os terminais acabam servindo basicamente de interface de entrada e saída aos usuários. Resumidamente, o servidor GNU/Linux com LTSP é um computador que pode ser acessado por várias pessoas por meio dos terminais.

O uso do GNU/Linux é um dos principais atrativos, pois não há um custo adicional com licenças, o que reduz drasticamente o preço de se implementar um laboratório de informática ou de se ter estações de trabalho em um ambiente público. Segundo uma pesquisa do Instituto Nacional de Tecnologia da Informação (ITI) [Info04], sobre custos entre softwares, o uso de software livre mostra uma economia de 97% em relação a softwares similares de propriedade privada. Ainda de acordo com o ITI, o software livre reduz os gastos com licenças e a remessa de royalties para o exterior. Além disso, permite aos usuários se apropriar das tecnologias livres e incentiva a conquista da autonomia no país na área de tecnologia da informação.

Existem várias vantagens em se utilizar servidores de terminais LTSP [Gasp05], por exemplo, o aproveitamento de velhos computadores, incapazes de rodar um bom programa, baixo custo e menor complexidade de manutenção e administração necessária apenas no servidor. Além disso, não há necessidade de pagamento de licenças de software na estrutura do LTSP, pois todos os softwares utilizados são gratuitos.

Outra importante vantagem dessa abordagem é a possibilidade de se utilizar estações de trabalho sem disco rígido ou CD-ROM, as chamadas estações diskless [Mori04]. Esse tipo de estação tem a vantagem de reduzir, de forma significativa, os problemas que o disco rígido traz consigo como o consumo de energia e as falhas frequentes, além de ser um componente sensível, que não possui uma vida útil muito longa. Essas vantagens tornam o uso de servidores de terminais interessante para instituições que possuem um parque de máquinas obsoletas e não dispõem de recursos para investir em novos computadores.

Dentro desta perspectiva, este estudo centrou-se na busca de soluções para aproveitamento de máquinas obsoletas em órgãos públicos e utilização de softwares livres para a diminuição de gastos com a sua aquisição. Como resultado, contribui para a implantação de laboratórios de informática em universidades públicas, que possibilitam uma boa perspectiva de utilização, buscando uma melhora do processo ensino-aprendizagem, no que diz respeito ao uso de computadores para a aquisição do conhecimento.

Este artigo está organizado da seguinte forma: o próximo tópico apresenta conceitos de software livre e o GNU/Linux, na seção 3 são descritos os métodos e ferramentas utilizadas, a seção 4 mostra a configuração dos terminais remotos, a seção 5 apresenta os testes de boot e a seção 6 descreve os testes de desempenho realizados. Por fim, na seção 7 apresentam-se as conclusões e trabalhos futuros.

2. SOFTWARE LIVRE E O GNU/LINUX

O termo software livre, como definido pela FSF (Free Software Foundation) [Foun09], baseia-se em idéias filosóficas. A filosofia de software livre se refere a liberdade e não explicitamente gratuidade. Basicamente, um software para ser considerado livre leva em conta a liberdade do usuário de poder utilizá-lo sem restrições, criar cópias, estudá-lo, modificá-lo, redistribuí-lo e melhorá-lo de acordo com suas necessidades. Para um software ser considerado

livre, ele deve conter, em sua distribuição, o código fonte, já que este é necessário para que o usuário possa modificá-lo ou melhorá-lo.

O Linux é o kernel de um sistema operacional livre, sendo seu nome originado pela contração das palavras Linus e Minix. O Linux teve seu início em 1991, criado por Linus Torvalds, um estudante de Ciência da Computação da Universidade de Helsinque, na Finlândia. Seu criador resolveu desenvolver um kernel melhor que o sistema limitado que usava em seu computador, o Minix. Torvalds e alguns dos primeiros programadores do Linux adaptaram o código para os componentes do sistema GNU e assim nasceu o GNU/Linux como se conhece hoje, um sistema operacional livre, completo, que tem evoluído muito.

Existem hoje muitas variações ou distribuições do sistema operacional GNU/Linux usando o kernel Linux. Por isso, o GNU/Linux é frequentemente chamado de “Linux” (é o kernel – núcleo de um sistema operacional) [Gnu09]. Apesar do nome Linux originalmente referenciar apenas o kernel, ele foi popularizado como nome de sistemas operacionais [Alec04].

A Free Software Foundation advoga que a combinação do núcleo Linux, conjunto de utilitários e aplicativos livres, que incluem aqueles desenvolvidos pelo seu projeto GNU, formam o sistema operacional que deve ser chamado de GNU/Linux, e não simplesmente de Linux. A questão tem sido objeto de intensos debates há anos, sem que um posicionamento geral e definitivo seja alcançado [Brli09a], [Brli09b].

O sistema operacional GNU/Linux adota a licença livre GPL (General Public License – Licença Pública Geral), a qual impõe que todos os interessados podem usá-lo e redistribuí-lo [Gnu09]. Aliado a inúmeros outros softwares livres, como o GNOME, o KDE, o Firefox, os softwares do sistema GNU e o OpenOffice.org, o Linux constitui um ambiente moderno, seguro e estável para os mais diferentes usos como desktops, servidores e até mesmo sistemas embarcados.

3. MÉTODOS E FERRAMENTAS

A metodologia utilizada para a implementação deste trabalho foi a simulação e testes de um laboratório com um servidor Gnu/Linux e estações obsoletas, todas com as mesmas configurações de hardware, que já se encontravam em uso por acadêmicos da universidade, mas em situação precária. Na implementação do LTSP, foi utilizada a distribuição do sistema operacional Debian versão 4.0 chamada Etch e OpenSUSE 10.2. Essas distribuições foram escolhidas por serem estáveis, possuírem grande quantidade de documentação, além de serem amigáveis, com suporte totalmente grátis e de resposta rápida da própria comunidade [Debi06].

Para a implementação do servidor de terminal, foi utilizado o LTSP (Linux Server Project), um conjunto de pacotes para sistemas operacionais GNU/Linux. Pode-se pensar no LTSP como um sistema operacional independente, usando de base para seu funcionamento qualquer distribuição GNU/Linux [Mcqu04].

O computador utilizado como servidor do LTSP tem as seguintes especificações técnicas:

- Processador Core Duo 1.66GHz;
- Memória RAM de 1 GB;
- HD Toshiba 120 GB 5400 RPM;
- Placa de rede Fast Ethernet 100Mbps;
- Drive de DVD-ROM.

Poderia ser utilizado um servidor com características de hardware um pouco inferior, mas, neste caso, buscou-se uma opção mais robusta a fim de o laboratório ter um tempo de vida maior.

Assim como todos os demais recursos, deve-se analisar o perfil das aplicações, entretanto observou-se uma fórmula simples de consumo do LTSP por terminal:

Para a instalação do sistema operacional, são necessários, aproximadamente, 2GB de espaço em disco, sendo que o tamanho restante do disco deve ser utilizado para os aplicativos de usuários instalados [Gasp09].

Quanto à memória RAM, 512MB são suficientes para um bom funcionamento do sistema operacional do servidor + $\langle \text{nr terminais} \rangle * 32 \text{ MB}$. Entretanto, ao observar os aplicativos que rodarão no servidor, não devemos apenas calcular o quanto um usuário ocupa de RAM e multiplicarmos pelo número de estações. Deve ser levado em consideração, principalmente, o número de aplicativos diferentes que serão utilizados e não somente o número de usuários [Gasp2009].

Para a instalação do LTSP, utilizou-se o comando “ltspadmin”. Após esta etapa deve ser realizada a configuração do servidor.

Antes de iniciar a configuração do LTSP, é necessário deixar o computador com o IP fixo, já que este será o servidor de imagens de sistema operacional para as estações diskless.

O próximo passo é configurar o servidor com o utilitário “ltspcfg”. Este utilitário fornece uma interface gráfica que facilita a configuração. Além de ajudar na configuração, o “ltspcfg” também exibe um relatório que mostra o status de todos os serviços e arquivos de configuração necessários ao LTSP (FIG. 3.5).

A função de configuração do “ltspcfg” pode ser visualizada na FIG. 3.6.

```

ghild@servidor: ~
Arquivo Editar Ver Terminal Abas Ajuda
ltspcfg v0.16 The Linux Terminal Server Project (http://www.LTSP.org)

Interface IP Address Netmask Network Broadcast Used
eth1 10.1.1.29 255.0.0.0 10.0.0.0 10.255.255.255 <-----

Service Installed Enabled Running Notes
dhcpd no no no
tftpd no no no Not installed !!!
portmapper Yes Yes Yes
nfs Yes Yes Yes
xdmcp Yes no no gdm Using: gdm

File Configured Notes
/etc/hosts no
/etc/hosts.allow no
/etc/exports no
/opt/ltsp/i386/etc/lts.conf Yes

Configured runlevel: 2 (value of initdefault in /etc/inittab)
Current runlevel: 2 (output of the 'runlevel' command)

Installation dir...: /opt/ltsp

Press <enter> to return to the main menu... █
    
```

Figura 3.5: Status dos serviços e arquivos de configuração

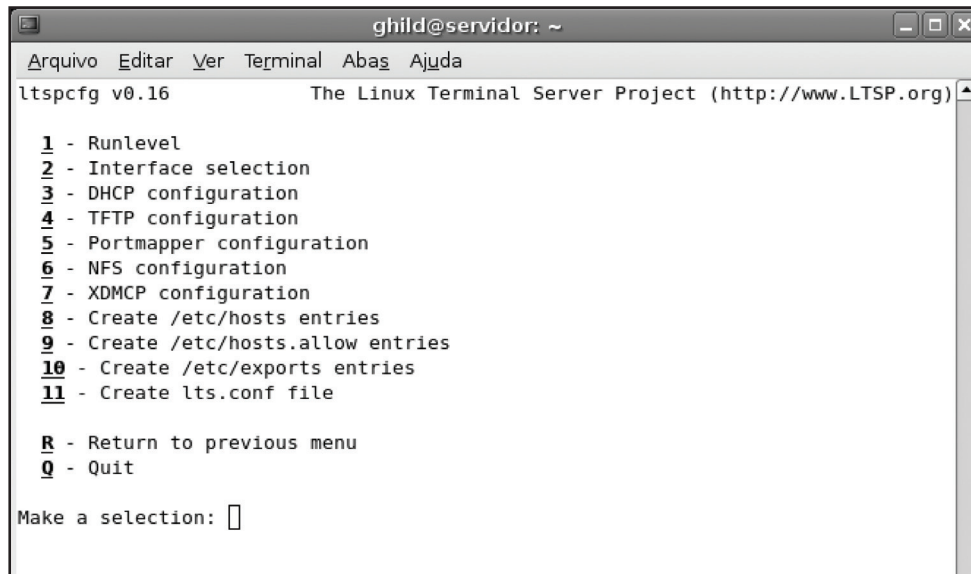


Figura 3.6: Configuração dos serviços

Os onze itens da lista representam os itens que devem estar configurados para que o LTSP funcione corretamente. Mais detalhes sobre eles podem ser encontrados em [Mcqu04].

Para desenvolver as estações de trabalho que utilizem o LTSP como sistema operacional, precisa-se, basicamente, de quatro serviços habilitados e configurados em um servidor, são eles: DHCP, TFTP, NFS e XDMCP.

DHCP (Dynamic Host Configuration Protocol - Protocolo de Configuração de Host Dinâmico): é um protocolo que visa atribuir endereços IP e configurações de forma dinâmica, já que cada computador deve possuir endereços IP distintos e/ou configurações diferentes [Come95], [Tane03].

A instalação do servidor DHCP no Debian é iniciada com o comando “sudo apt-get install dhcp3-server”. No arquivo “/etc/dhcp3”, gerado após a instalação deve-se alterar a opção “subnet” de acordo com a rede onde o servidor está e a opção “range” para uma faixa de IPs livres na rede.

TFTP (Trivial File Transfer Protocol - Protocolo Simples de Transferência de Arquivos): neste trabalho, o protocolo TFTP foi utilizado para obter a imagem do kernel Linux para que a estação diskless carregue-o na sua memória e possa ser iniciada.

Para a instalação do TFTP é utilizado o comando “sudo apt-get install tftpd-hpa”. Após instalado, deve-se alterar o arquivo de configuração do servidor TFTP para que este possa ser executado quando o servidor for iniciado. Para isso, deve-se modificar a opção: “RUN DAEMON” para “yes”.

NFS (Network File System – Sistema de Arquivo em Rede): este protocolo permite que a árvore de diretórios local seja montada nas estações diskless. O protocolo também é utilizado caso seja definido no arquivo “\lts.con” a opção para uso de swap (extensão da memória RAM) remoto. O NFS é instalado juntamente com a instalação do Debian GNU/Linux e configurado com o utilitário “ltspcfg”.

XDMCP (X Display Manager Control Protocol – Protocolo de Controle do Gerenciador de Exibição X): é um protocolo responsável pelo controle do compartilhamento da interface gráfica do servidor LTSP com as estações remotas. A FIG. 1 apresenta um exemplo do funcionamento do XDMCP.

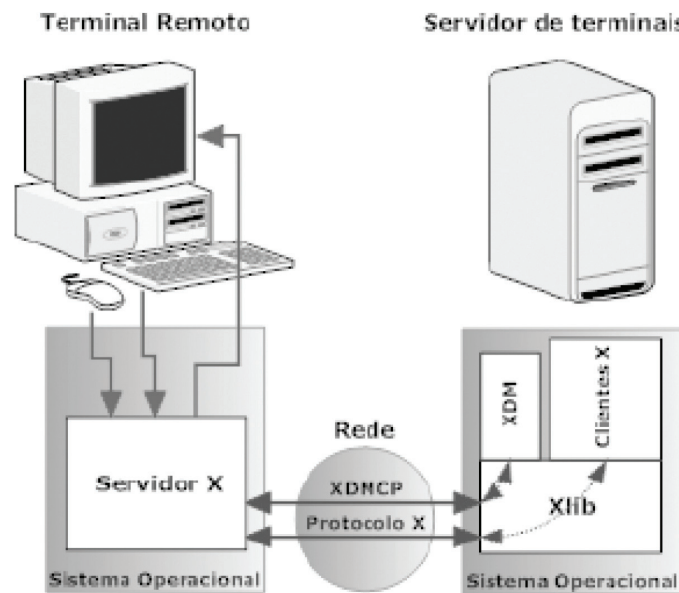


Figura 1: Exemplo de funcionamento do XDMCP

O protocolo X foi desenvolvido com uma arquitetura cliente-servidor em mente, onde as aplicações são os clientes (clientes X) e o servidor (servidor X) é o responsável por exibir as imagens em um monitor e obter a entrada de dados a partir de dispositivos de entrada como o teclado e o mouse. O protocolo de comunicação entre o servidor e os clientes (protocolo X) é transparente na rede, ou seja, o servidor e os clientes podem ser executados no mesmo computador ou em computadores diferentes. Os clientes não se comunicam diretamente com o servidor X; ao invés disso, utilizam uma biblioteca chamada Xlib. Essa biblioteca é responsável pela tradução de comandos em alto nível provenientes dos clientes para comandos em baixo nível enviados ao servidor [Manr05]. A FIG. 2 demonstra a arquitetura do X Window System.

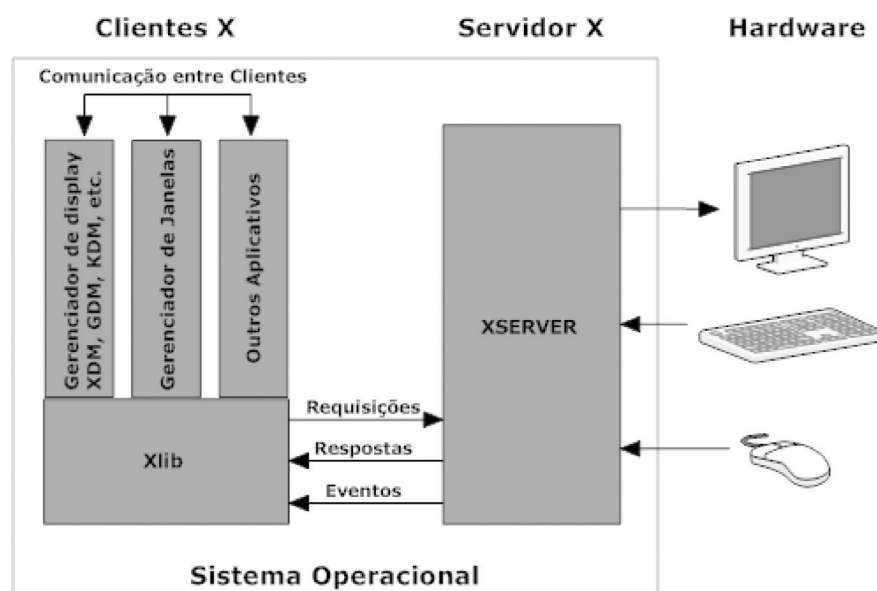


Figura 2: X windows systems

Na FIG. 2, podemos observar a presença de um cliente X chamado gerenciador de login (display manager). Um gerenciador de login é um programa que permite o início de uma sessão em um servidor X, local ou remota. Localmente, o gerenciador inicia um servidor X, e este exibe uma tela de login para que possa ser iniciada uma sessão em algum ambiente desktop (Gnome, KDE, XFCE, entre outras) disponível no computador.

Remotamente, um servidor X é iniciado e utiliza o protocolo XDMCP para se conectar em um gerenciador de display remoto e obter uma tela de login [Chao07]. Atualmente existem vários gerenciadores de login disponíveis, sendo os mais famosos o XDM (X Display Manager) que é o gerenciador-padrão do X, o GDM (Gnome Display Manager), o gerenciador do Gnome, e o KDM (KDE Display Manager), o gerenciador do KDE.

O protocolo XDMCP vem incluso no conjunto de pacotes “ambiente desktop” no Debian GNU/Linux que foi previamente instalado. Apenas é necessário ativar o protocolo XDMCP no gerenciador de login, no caso do Debian, o GDM. Contudo a ativação do protocolo XDMCP é realizada com o utilitário “ltspcfg”.

4. CONFIGURAÇÃO DOS TERMINAIS REMOTOS

A configuração dos terminais pode ser dividida em duas maneiras: configurações locais, onde são efetuadas configurações no hardware ou software do terminal remoto, e configurações remotas, onde são efetuadas configurações no software do servidor de terminais que modificam o comportamento do terminal remoto.

Um servidor de terminais tem o objetivo de oferecer serviços a diversas estações; devido a isso, seu hardware deve ser robusto o suficiente para atender as estações que ele se propõe a gerenciar.

4.1 CONFIGURAÇÕES LOCAIS

A configuração local dos terminais geralmente envolve a remoção de dispositivos de hardware. É recomendável a remoção de dispositivos que não serão utilizados para que seja possível a redução dos gastos com energia elétrica e manutenção. Para se obter um terminal remoto de baixo custo e que seja ao mesmo tempo veloz, o mínimo necessário em termos de hardware é:

1. Placa-mãe preferencialmente com suporte a processadores Pentium com, no mínimo, 100Mhz;
2. 8 MB de memória RAM;
3. Placa de rede Fast Ethernet de 100Mbps;
4. Placa de vídeo PCI.

4.2 CONFIGURAÇÕES REMOTAS

A configuração remota dos terminais consiste na edição de arquivos de configuração contidos no servidor, como o arquivo “\dhcpd.conf” e o arquivo “\lts.conf”. Nesses arquivos, é possível definir configurações comuns a todas as estações remotas e também é possível definir configurações específicas para cada terminal.

No arquivo “\dhcpd.conf”, é possível definir qual IP o terminal irá receber e qual servidor deverá usar para obter a imagem do sistema operacional entre outras opções.

Cada terminal pode ter uma entrada no arquivo “dhcpd.conf” como mostra o Quadro 1.

Quadro 1: Configuração de um *host* no “dhcpd.conf”

```
host ws002 {  
    hardware ethernet 00:E0:7D:D7:AE:84;  
    fixed-address 192.168.1.2; }  
}
```

O arquivo “\lts.conf” é um arquivo utilizado pelo LTSP para definir algumas configurações essenciais dos terminais remotos.

A entrada “host” no arquivo “dhcpd.conf” define a configuração para um terminal específico. O servidor DHCP verifica o endereço MAC de um terminal que está requisitando um endereço IP e o compara com a entrada “hardware ethernet” dos hosts definidos no arquivo “dhcpd.conf”. Se a entrada existir, o servidor DHCP oferece ao terminal o IP contido na entrada “fixed-address”. Se o endereço MAC da placa de rede não for encontrado em nenhuma entrada “host”, o servidor DHCP utilizará a configuração comum a todos os terminais para definir o IP a ser atribuída ao terminal.

A utilização da entrada “host” para cada terminal que se conectará ao servidor de terminais facilita a manutenção do LTSP. Como cada terminal usará sempre o mesmo IP, descobrir problemas de rede em um determinado terminal fica muito mais fácil.

O arquivo “lts.conf” é um arquivo utilizado pelo LTSP para definir algumas configurações essenciais dos terminais remotos. Quando o kernel do LTSP está sendo iniciado em um terminal remoto, ele busca algumas entradas no arquivo “lts.conf” para configurar o terminal. Assim como no arquivo “dhcpd.conf”, o arquivo “lts.conf” pode conter entradas comuns a todos os terminais ou entradas específicas. A vantagem de se utilizar entradas específicas para cada terminal reside no fato de que, em um mesmo ambiente de rede (um laboratório de informática, por exemplo), podem existir terminais com peças de hardware diferentes. No Quadro 2, pode-se observar um exemplo do arquivo de configuração “lts.conf” criado automaticamente pelo utilitário “ltspcfg”.

Quadro 2: Exemplo de arquivo ‘lts.conf’

```
[Default]  
SERVER          = 10.1.1.6  
XSERVER         = auto  
X_MOUSE_PROTOCOL = “PS/2”  
X_MOUSE_DEVICE  = “/dev/psaux”  
X_MOUSE_RESOLUTION = 400  
X_MOUSE_BUTTONS = 3  
SCREEN_01       = startx
```

A entrada “[Default]” é a entrada de configuração comum a todos os terminais remotos.

Se nenhuma entrada específica for definida para um terminal remoto, o LTSP utilizará todas as informações da entrada “[Default]”. Há um manual que descreve todos os tipos de entradas possíveis no arquivo “lts.conf.readme” encontrado no diretório “/opt/ltsp/i386/etc/”. A entrada “SERVER” se refere ao IP de qual computador o servidor X do terminal deverá enviar uma requisição XDMCP para obter uma tela de login.

A entrada “XSERVER” define qual servidor X o terminal irá usar. O valor “auto” desta entrada define que o LTSP escolherá automaticamente qual servidor X usar. As entradas “X_MOUSE” definem as configurações para o tipo de mouse que a estação possui (serial, PS2, USB etc.). A entrada “SCREEN 1” define o tipo de sessão (shell, gráfica etc.) que será utilizada pelo terminal, e o valor “startx” define que uma sessão gráfica será iniciada.

Para definir configurações específicas para cada terminal, deve-se criar uma entrada para cada um dos terminais, como mostra o exemplo do Quadro 3. Os nomes a serem utilizados em cada sessão dependem do IP definido para o terminal. O arquivo “hosts” configurado pelo LTSP, associa o nome que será utilizado por um terminal a um endereço IP.

Quadro 3: Exemplo de configuração específica no “lts.conf”

```
[Default]
SERVER          = 10.1.1.6
XSERVER         = auto
X_MOUSE_PROTOCOL = "PS/2"
X_MOUSE_DEVICE  = "/dev/psaux"
X_MOUSE_RESOLUTION = 400
X_MOUSE_BUTTONS = 3
SCREEN_01       = startx
#Terminal com mouse serial
[ws002]
X_MOUSE_PROTOCOL = "Microsoft"
X_MOUSE_DEVICE   = "/dev/ttyS0"
#Terminal usando o XFree86 3.3.6
[ws003]
XSERVER          = XF86_SVGA
```

Quando um terminal remoto estiver sendo iniciado, o LTSP verificará a entrada “[DEFAULT]” e se existe uma entrada específica para o terminal. Se existir uma entrada, o LTSP usará essas configurações, sobrescrevendo as entradas que sejam iguais à entrada “[DEFAULT]”.

5. TESTES DE BOOT

Este item descreve os testes de *boot* realizados com o LTSP. *Boot* é o procedimento pelo qual um sistema simples carrega outro sistema, geralmente mais complexo [Abne02]. O GRUB (*GRand Unified Bootloader*), LILO (*Linux LOader* - carregador semelhante ao GRUB), NTLDR (*NT Loader* - carregador do Windows NT e derivados) e o *Etherboot* são exemplos de sistemas com essa capacidade.

O *Etherboot* é um projeto que desenvolve código para que computadores possam iniciar seus sistemas operacionais pela rede (*boot* remoto). O código do *Etherboot* que carrega os sistemas operacionais pode ser armazenado em diversos dispositivos, como memórias ROM (*Read Only Memory*), disquetes, CD-ROM, discos rígidos entre outros [Proj07]. O projeto *Etherboot* mantém uma página na internet – (www.rom-o-matic.net) na qual é possível a criação dinâmica de imagens do sistema *Etherboot* para que estas possam ser gravadas em dispositivos capazes de iniciar o processo de boot. Este programa, chamado *gPXE*, é um carregador de *boot open source* (GPL). Atualmente é possível gerar código compatível com 289 NICs (*Network Interface Card* – placa de rede) diferentes, em 10 formatos distintos.

Para gerar uma imagem no ROM-o-matic, é necessário selecionar o modelo da placa de rede utilizada pela estação remota e selecionar qual o tipo de formato desejado. A placa de rede utilizada para os testes com o *Etherboot* é uma Realtek RTL8139.

A caixa de seleção NIC/ROM do ROM-o-matic lista as placas de rede suportadas. Cada placa de rede disponível está listada como mostra o Quadro 4.

Quadro 4: Lista do ROM-o-matic

Família:nome da rom – [PCI ID do fabricante:PCI ID do dispositivo]
--

Neste trabalho foi realizado o boot por ROM pela placa de rede. A grande maioria das placas de rede atuais possui a capacidade de armazenar um código executável capaz de iniciar o processo de boot. Dessa forma, deve-se obrigatoriamente selecionar a placa de rede da lista que contenha a PCI ID do fabricante e a PCI ID do dispositivo correspondentes à placa de rede onde a EPROM será inserida. Caso contrário, a placa de rede não reconhecerá o código do *Etherboot* e o boot pela placa de rede não irá funcionar [YaGu03].

A grande maioria das placas de rede atuais, sejam elas placas off-board (placas que podem ser inseridas em uma placa-mãe) ou on-board (placas embutidas nas placas-mãe) possuem a capacidade de armazenar um código executável capaz de iniciar o processo de boot. A diferença entre elas, é que as placas on-board já possuem uma memória disponível, que normalmente vem com um código de boot pronto para utilizar.

As placas de rede off-board geralmente possuem um soquete DIP (Dual Inline Package - pacote em linha dupla), onde pode ser inserida uma memória ROM (PROM, EPROM, EEPROM ou Flash) que contenha o código de boot.

5.1 MÉTODO DE GRAVAÇÃO POR HOT-SWAP

Para a gravação do código *Etherboot* em um tipo de memória ROM, é necessária a utilização de um dispositivo capaz de gravar dados nessas memórias. O tipo de gravador utilizado dependerá do tipo de ROM que se deseja gravar. Como um dos objetivos do trabalho foi de diminuir os custos da implementação de um servidor LTSP e seus terminais remotos, as memórias utilizadas para a gravação são memórias removidas de placas-mãe antigas ou sem funcionamento, onde a BIOS está gravada. Entretanto deve-se buscar memórias que sejam do tipo DIP (tipo de encapsulamento retangular utilizado em chips, com uma fileira de pinos em cada lado), pois é o mesmo tipo de encapsulamento utilizado pelo soquete das placas de rede. Normalmente, placas-mãe antigas utilizam este tipo de encapsulamento com uma configuração

de 32 pinos (DIP32). Algumas placas de rede possuem a capacidade de gravar memórias ROM nativamente, não sendo necessária a utilização de um gravador. Porém o custo destas placas de rede é superior ao custo das placas sem esta funcionalidade.

Para se identificar uma memória extraída de uma placa-mãe deve-se observar as inscrições sobre a sua parte superior e buscá-las, em uma base de dados de datasheets (termo técnico usado para identificar um documento relativo a um determinado produto), como por exemplo, a página da Internet DatasheetCatalog.net (<http://www.datasheetcatalog.net/>).

É importante verificar também a voltagem necessária para gravar a memória Flash. Algumas placas-mãe possuem um jumper (interruptor móvel que conecta dois pontos em um circuito) próximo ao soquete da memória da BIOS que possibilita a escolha da voltagem da gravação. É importante escolher a voltagem certa, para que a gravação funcione corretamente. Se a voltagem de gravação escolhida para uma memória for maior que a especificada no seu datasheet, é possível que a memória seja danificada. Se a placa-mãe não possuir uma identificação da voltagem utilizada para a gravação da memória, é possível verificar no datasheet da memória da BIOS atualmente instalada a voltagem utilizada pela placa-mãe.

Os gravadores profissionais de memórias ROM têm um custo; assim, foi utilizado o método de hot-swap (troca quente) que utiliza a capacidade de gravação das placas-mãe para a gravação de outras memórias. O método consiste em iniciar o computador que será utilizado como gravador com um sistema operacional, como o DOS, carregar o programa de gravação e remover com cuidado a memória da BIOS do computador do soquete onde ela está inserida com o auxílio de uma ferramenta, preferencialmente plástica, para evitar curto-circuito nos componentes da placa-mãe. O computador funciona normalmente após a remoção da memória, já que o código da BIOS não é acessado enquanto o programa de gravação está sendo executado. Deve-se, então, inserir no soquete uma memória Flash extraída de outra placa-mãe que será utilizada para receber os dados a ser gravados. Concluída a gravação, pode-se retirar a memória Flash.

5.2 UTILIZAÇÃO DO MÉTODO DE GRAVAÇÃO

No presente trabalho, uma placa-mãe Pc-Chips modelo M525 foi utilizada como gravadora. A placa de rede Realtek RTL8139 foi utilizada como placa de teste para a memória gravada. A placa de rede em questão possui um soquete DIP32. Entretanto algumas das placas de rede que utilizam o mesmo chipset RTL8139 possuem um soquete do tipo DIP28. A memória Flash utilizada para a gravação foi uma “Macronix \mx28f1000ppc-12c4” com um encapsulamento DIP32 retirada de uma placa-mãe sem uso. Segundo o datasheet dessa memória, ela possui 128 KB de espaço disponível para gravação e utiliza uma voltagem de gravação de 12 Volts.

É necessário obter a imagem do Etherboot a ser gravada na memória, pela ferramenta ROM-o-matic [Romo07]. A imagem a ser gerada é do tipo “Binary ROM Image(.zrom)”, um arquivo binário que é o tipo de imagem para a gravação em memórias Flash.

Depois da imagem gerada, o arquivo foi renomeado para “rtl8139.rom” e o seu tamanho ficou com 32KB. Para que seja possível a gravação da imagem de 32KB em uma memória Flash de 128KB, é necessário completar os 96KB restantes do arquivo, caso contrário o programa de gravação não irá gravar a imagem na memória. O restante do arquivo pode ser completado com “1’s”, “0’s” ou até mesmo com cópias da imagem binária concatenadas até atingir os 128KB necessários.

O comando no Quadro 5 mostra como gerar uma imagem de 128KB (“img128KB.rom”) concatenando a imagem “rtl8139.rom” de 32KB com 98034 Bytes (96KB) de FF (valor hexadecimal correspondente a 11111111 em binário). A imagem gerada conterá o código do

Etherboot nos seus primeiros 32KB, e o restante do espaço será ocupado por “1’s” [YaGu03].

Quadro 5: Escrevendo 1’s em um arquivo binário

```
(perl -e 'print "\xFF" x 98034'; cat rtl8139.rom) > img128KB.rom
```

O passo seguinte para a gravação é copiar a imagem gerada no processo anterior para um disquete com o sistema operacional DOS e o programa de gravação UniFlash. O UniFlash é um programa de gravação universal que pode ser utilizado para gravar memórias Flash em uma grande variedade de placas-mãe. A FIG. 3 mostra a tela inicial do programa UniFlash depois de executado. Em seguida, deve-se remover a memória Flash da BIOS do computador e inserir a memória a ser gravada. Existe uma marcação em uma das laterais das memórias Flash, que indica o posicionamento correto da memória no soquete. A marcação da memória deve coincidir com a marcação do soquete, caso contrário a memória será danificada.

```
UniFlash v1.40 (c) 2005 Rainbow Software (http://www.uniflash.org)
Original version by Pascal van Leeuwen & Galkowski Adam

(FFFF,FFFF) Flash ROM chip: UNKNOWN
              Organisation: N/A (Is write protect disabled?)
              PCI chipset: Intel AGPSet 440BX/ZX
              Last write status: not available

              Write backup BIOS image to file
              Flash BIOS image INCLUDING bootblock
              Flash backup BIOS image to Flash ROM
              Redetect Flash ROM
              CMOS submenu »
              ADVANCED submenu »

              Quit

ROM base: FFFE0000, memory dump at FFFE0000-->(87BD80F9)
Memory: XMSPos=00310000, XMSLeft=67107840
Exit UniFlash
```

Figura 3: Tela principal do UniFlash

Depois de inserida a memória a ser gravada, seleciona-se a opção “Redetect FlashROM” (redetectar memória Flash ROM) do menu do “UniFlash” para verificar se a memória foi inserida corretamente. Se a memória não foi inserida corretamente, o item “FlashROM chip” (chip de memória Flash) do UniFlash exibirá a mensagem “UNKNOWN” (desconhecido). Se a memória inserida for detectada corretamente, o item “FlashROMchip” exibirá o nome da memória Flash. O próximo passo consiste na gravação da memória por intermédio da opção “Flash BIOS image INCLUDING bootblock” (gravação da imagem de BIOS, incluindo bloco de boot). Seleciona-se o arquivo a ser gravado e, em seguida, o UniFlash pede a confirmação da ação. Se confirmada, uma barra exibe o progresso da gravação da imagem na memória Flash. Se a imagem for gravada com sucesso, o item “Last write status” (status da última gravação) exibirá a mensagem “SUCCESS” (sucesso).

5.3 HABILITANDO A PLACA DE REDE PARA O BOOT REMOTO

Com a memória Flash já gravada com o código do Etherboot, o passo seguinte consiste em configurar a placa de rede para que ela possa iniciar o processo de boot. A configuração da placa de rede utiliza um programa chamado RSET8139.

O programa de configuração RSET8139 pode ser encontrado na sessão de drivers para a placa RTL8139 na página da Realtek (www.realtek.com.tw). Nesse programa, é possível habilitar o boot usando-se a memória ROM inserida na placa de rede. A opção “Set Up New Configuration” (ajustar nova configuração) do menu exibe uma nova tela com algumas opções de ajuste, inclusive a opção “Boot ROM Size”. Por padrão, o valor da opção “Boot ROM Size” é “No Boot ROM” (sem ROM de boot). Selecionando esta opção para configuração, uma janela surge com a opção “Enable Boot ROM” (habilitar ROM de boot) para habilitar o boot pela memória ROM. Quando esta opção é habilitada, uma nova janela surge, desta vez para definir o tamanho da memória que será inserida no soquete da placa de rede. O tamanho selecionado deverá ser igual ou maior ao tamanho do código do Etherboot gravado na memória Flash (esse código refere-se ao código antes da inserção de “1’s”).

Depois de configurada a placa de rede, ocorreu a inserção da memória Flash gravada no soquete da placa de rede.

6. TESTES DE DESEMPENHO

Este tópico descreve os testes de desempenho realizados no laboratório de informática da UNICENTRO para avaliar o servidor de terminais com o LTSP em duas distribuições GNU/Linux. A primeira delas é o Debian 4.0 (codinome Etch) e a segunda distribuição utilizada para os testes é a OpenSUSE 10.2. Os testes mostrados aqui se referem à distribuição Debian.

Os terminais utilizados para os testes têm a seguinte configuração:

- Processador Pentium 166Mhz;
- Memória RAM de 48MB;
- HDs com tamanhos variados;
- Placa de rede Fast Ethernet 100Mbps.

O software utilizado para as medições de desempenho é o Gnome System Monitor que serve para monitorar os processos e recursos do sistema [Te04]. Na aba “recursos” do software, há três gráficos, sendo o primeiro gráfico o histórico do processador, o segundo, o histórico de uso da memória RAM e da memória swap e o terceiro, o histórico do uso da rede.

6.1 TESTE DE LOGIN

O primeiro teste consistiu em verificar o desempenho quando todos os terminais efetuaram login (caracteres solicitados para o usuário acessar um sistema computacional) no sistema. Todos os dez terminais efetuaram login no Gnome ao mesmo tempo.

Como mostra a Figura 4, o uso da rede teve inúmeras oscilações, mas a porcentagem de uso de CPU (CPU1 - 7% e CPU2 - 5%), memória RAM (60%) e memória swap (0% - não necessitou a utilização) demonstraram que o servidor suportou o processamento das estações ao realizarem login.

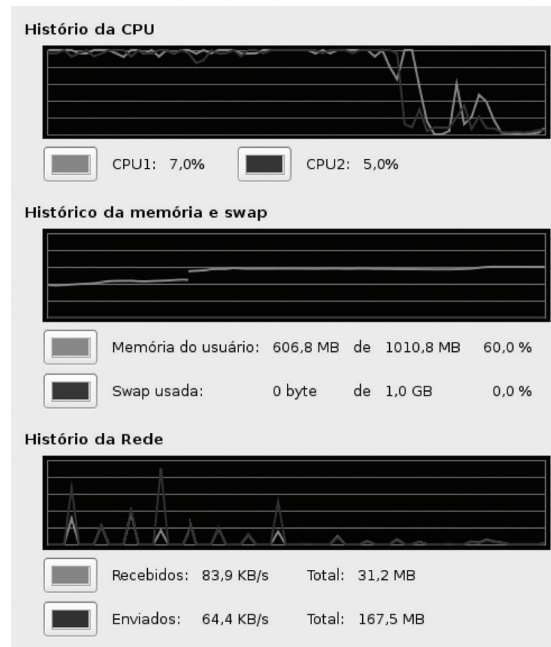


Figura 4: Login dos terminais no sistema

6.2 TESTE COM EDITOR DE TEXTO

O segundo teste utilizou um editor de texto, um software muito utilizado no cotidiano de laboratórios de informática.

O editor de texto do *OpenOffice* foi executado nos dez terminais, e as cargas de processador, memória e rede foram medidas depois que todos os terminais terminaram de carregar o *software*.



Figura 5: OpenOffice depois de carregado

Observando-se a FIG. 5, nota-se que o uso da CPU1 ficou em 3,1%, da CPU2, 2,4%, da memória RAM, 75,2% de memória *swap* agora necessitou ser utilizada e ficou com taxa de utilização de 6,9%. O uso da rede manteve-se constante neste teste.

6.3 TESTE COM ARQUIVO PDF

No último teste realizado, abriu-se um arquivo PDF de 1MB no programa Gpdf.

Observando-se a Fig. 6, pode-se perceber que o uso da rede na abertura do arquivo PDF foi mais intenso comparado ao uso da rede ao utilizar um editor de texto.

O uso da CPU1 ficou em 2%, da CPU2 7,7%, memória RAM 67,5%, e a memória *swap* não precisou ser utilizada.

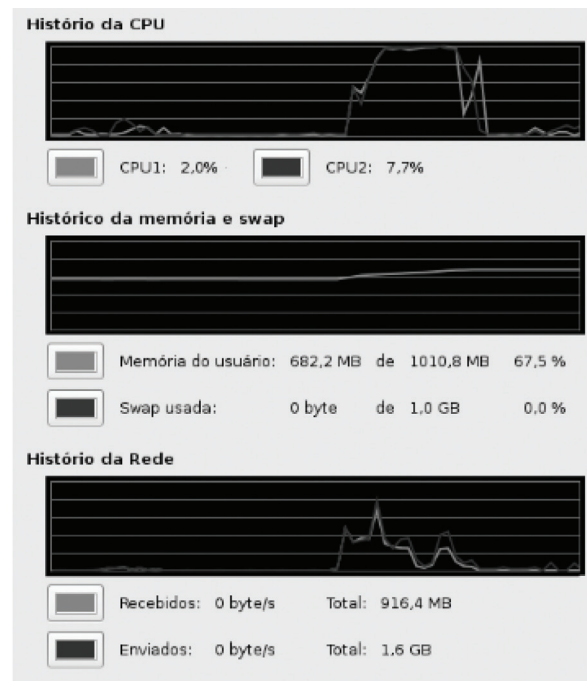


Figura 6: Abrindo um arquivo PDF

7. CONCLUSÃO

Este artigo demonstrou a montagem e configuração de terminais leves, utilizando-se um servidor LTSP. Após os testes de desempenho realizados, comprovou-se que o LTSP com apenas um servidor suporta vários terminais e possibilita uma maior sobrevivência de computadores obsoletos. Dessa maneira, a implantação de um servidor de terminais leves com LTSP é válida para transformar computadores antigos em estações de trabalho úteis praticamente sem acrescentar gastos. Esse estudo pretende servir de base para outras universidades e órgãos públicos que possuem máquinas obsoletas, as quais estão sem utilização, e transformá-las em potências laboratórios de informática visando ao aumento de produtividade e troca de conhecimento.

Como trabalho futuro, pretende-se estudar a possibilidade de utilização do LTSP para a implementação de um laboratório de multiterminal (um computador capaz de ser utilizado por múltiplos usuários ao mesmo tempo localmente) ou uma rede de computadores que tenham a

capacidade de processamento distribuído utilizando, por exemplo, o openMosix com o objetivo de aumentar o desempenho de computadores.

KEYWORDS

Thin Terminals, GNU/Linux, Debian, Opensuse, LTSP, Diskless.

ABSTRACT

New programs and operating systems appear everyday. They require greater processing capacity. The reuse of obsolete hardware is a necessity due fast evolution of computing, difficulties in acquiring equipment and release of public's property item. In order to supply this necessity, the use of terminal servers that turn old computers in useful workstations almost without adding expenses would be a solution. This approach becomes attractive to institutions that have a great number of obsolete machines and lack of resources to invest in new computers.

This article presents the reuse of computers with old hardware through implementation of a terminal servers in the GNU/Linux operating system, with LTSP (Linux Terminal Server Project) in a computer laboratory of a public university. Furthermore, installation and configuration process are described related to the setting of a laboratory where the terminals connect to the serve through a local network. Several mechanisms were studied to achieve the remote boot being the most effective alternative the use of diskless terminal (without hard disk) through the use of a memory ROM inserted into network card. Tests for performance were made with the aim of evaluating the feasibility of LTSP implementation.

REFERÊNCIAS BIBLIOGRÁFICAS

- [Abne02] ABNEY, S. *Bootstrapping*. Disponível em <http://www.vinartus.net/spa/02a.pdf>. Acesso em: 15/08/07.
- [Alec06] ALECRIM, E. A história do GNU/Linux. Disponível em: <http://www.infowester.com/gnu/linux5.php>. Acesso em: 09/05/06.
- [Brli09a] BR-LINUX. A controvérsia: 'Linux' X 'GNU/Linux'. Disponível em: http://br-linux.org/linux/a_controversia_linux_x_gnulinux. Acesso em: 12/06/09 a.
- [Brli09b] BR-LINUX. O que é Linux. Disponível: em <http://br-linux.org/faq-linux/>. Acesso em: 12/06/09 b.
- [Chao07] CHAO, T. GNU/Linux XDMCP HowTo. Disponível em: <http://tldp.org/HOWTO/XDMCP-HOWTO/index.html>. Acesso em: 0/07/07.
- [Come95] COMER, D. E. Interligação em Redes com TCP/IP - Princípios, Protocolos e Arquitetura – Trad. ARX Publicações. 3. ed. Lisboa: ARX Publicações, 1995.
- [Debi06] DEBIAN. Razões para escolher o Debian. Disponível em: http://www.debian.org/intro/why_debian. Acesso em: 03/04/06.
- [Gasp05] GASPARY, L. P. Linux Server Project - LTSP. Disponível em: <http://www.inf.unisinos.br/paschoal/arqsgerenciaredes/trabs/gradtec/ltsp.pdf>. Acesso em: 24/03/06.
- [Gnu09] GNU OPERATION SYSTEM. GNU/Linux and the GNU Project. Disponível em: <http://www.gnu.org/gnu/why-gnu-gnu/linux.html>. Acesso em: 20/04/09.
- [Gnu09] GNU LICENÇAS DE SOFTWARE LIVRE. Disponível em: <http://www.gnu.org/licenses/licenses.pt-br.html>. Acesso em: 12/06/09.

- [Foun09] FOUNDATION F. S. What is free software and why is it so important for society? Disponível em: <http://www.fsf.org/>. Acesso em: 20/04/09.
- [Foun06] FOUNDATION F. S. O que é Software Livre? Disponível em: <http://www.gnu.org/philosophy/free-sw.pt.html>. Acesso em: 20/05/06.
- [Info06] INFORMÁTICA T. Uso de Software livre gera economia de até 97%. Disponível em: <http://informatica.terra.com.br/ebusiness/interna/00I399509-I716,00.html>. Acesso em: 04/03/06.
- [Ltsp05] LTSP. Linux Server Project - LTSP. Disponível em: <http://www.ltsp.org>. Acesso em: 01/03/06.
- [Manr05] MANRIQUE, D. X Window System Architecture Overview HOWTO. Disponível em: <http://tldp.org/HOWTO/XWindow-Overview-HOWTO/index.html>. Acesso em: 01/07/07.
- [Mcqu04] MCQUILLAN, J. A. LTSP - Linux Server Project - v4.1. Disponível em: <http://ltsp.mirrors.tds.net/pub/ltsp/docs/ltsp-4.1-ptbr.html>. Acesso em: 22/03/07.
- [Mori04] MORIMOTO, C. E. Entendendo e dominando o GNU/Linux. 3. ed. Digerati, 2004.
- [Proj07] PROJECT E. About EtherBoot. Disponível em: <http://www.etherboot.org/wiki/about>. Acesso em: 15/08/07.
- [Romo07] ROM-O-MATIC. Ferramenta ROM-o-matic. Disponível em: <http://www.rom-o-matic.org>. Acesso em: 03/08.
- [Tane03] TANEMBAUM, A. S. Redes de Computadores. 4. ed. Campinas: Campus, 2003.
- [Team04] TEAM, S. G. D. System monitor manual V2.1. Disponível em: <http://library.gnome.org/users/gnome-system-monitor/2.20/index-info.html.en>. Acesso em: 16/09/07.
- [Wirz06] WIRZENIUS, L. GNU/Linux Anecdotes. Disponível em: http://liw.iki._/liw/texts/gnu/linuxanecdotes.html. Acesso em: 10/05/ de 2006.
- [YaGu03] YAP, K., GUTSCHKE, M. Etherboot user manual. Setembro 2003. Disponível em: <http://www.etherboot.org/wiki/usermanual>. Acesso em: 16/08/07.

SOBRE OS AUTORES

GISANE A. MICHELON

É graduada em Análise de Sistemas pela Unicentro – Universidade Estadual do Centro-Oeste, mestre em Engenharia Elétrica e Informática Industrial da UTFPR – Universidade Tecnológica Federal do Paraná. Desde 2006, é professora do Departamento de Ciência da Computação na UNICENTRO; suas áreas de interesse são sistemas operacionais e redes de computadores.

GIANCARLO FERNANDO HILD

É graduado em Análise de Sistemas pela Unicentro – Universidade Estadual do Centro-Oeste. Suas áreas de interesse são redes de computadores e arquitetura de computadores.