

CLEMENTE SCARATTI

**CUSTOMIZAÇÃO DE UMA DISTRIBUIÇÃO LINUX LIVE-CD PARA
SER UTILIZADA NOS CURSOS DE CIÊNCIA DA COMPUTAÇÃO E
SISTEMAS DE INFORMAÇÃO**

Monografia apresentada à UNOCHAPECÓ
como parte dos requisitos para a obtenção do
grau de Bacharel em Ciência da Computação.
Orientador: Prof. Elton Luís Minetto

Chapecó - SC, dez. 2007

CUSTOMIZAÇÃO DE UMA DISTRIBUIÇÃO LINUX LIVE-CD PARA SER UTILIZADA NOS CURSOS DE CIÊNCIA DA COMPUTAÇÃO E SISTEMAS DE INFORMAÇÃO

CLEMENTE SCARATTI

Esta Monografia foi julgada para obtenção do título de Bacharel em Ciência da Computação, na área de concentração e aprovada pelo curso de Ciência da Computação

ORIENTADOR: Prof. Elton Luís Minetto

COORDENADORA DO CURSO: Prof. Mônica Tissiani de Toni Pereira

BANCA EXAMINADORA

PRESIDENTE: Prof. Mônica Tissiani de Toni Pereira

Prof. Luciano Frosi

Prof. Radamés Pereira

AGRADECIMENTOS

Agradeço ao meu orientador Elton Luís Minetto pela paciência, incentivo e a disposição em tornar este projeto possível.

Agradeço aos meus familiares pela oportunidade que tenho de concluir esta graduação.

Agradeço aos amigos e colegas de trabalho pelo incentivo e ajuda prestados.

SUMÁRIO

1. INTRODUÇÃO.....	12
1.1 Organização do Projeto.....	12
2. LINUX.....	14
2.1 Software Livre.....	14
2.2 Sistema Operacional	15
2.3 Introdução ao Linux.....	18
2.4 Distribuições Linux.....	20
2.4.1 Principais Distribuições.....	21
2.4.2 Qual é a melhor distribuição?.....	23
2.4.3 Interfaces Gráficas do Linux.....	25
3. LINUX LIVE-CD.....	28
3.1 Características.....	29
3.2 Vantagens.....	32
3.3 Desvantagens.....	34
3.4 Principais distribuições Live-CD.....	34
3.5 Ferramentas para customizar uma distribuição.....	37
3.5.1 Introdução.....	37
3.5.2 Linux From Scratch.....	38
3.5.3 Remasterização.....	40
3.5.4 MKDistro Maker.....	41
4. DESENVOLVIMENTO DO PROTÓTIPO.....	44
4.1 Ifslivecd-x86-6.2-pre2-BuildScripts.....	45
4.2 JhaLFS-2.3.1.....	47
4.3 Utilização do Ifslivecd-x86-6.2-pre2-BuildScripts.....	47
4.4 Processo de Remasterização.....	50
4.4.1 Alteração dos scripts de inicialização e arquivos de configuração.....	56
4.4.2 Adição de pacotes.....	57
4.4.3 Desenvolvimento do “Painel do Live-CD”.....	74
4.4.4 Personalização do XFCE.....	78
4.4.5 Personalização do ISSUE.....	80

4.4.6 Personalização do Splash Screen.....	81
4.4.7 Geração da imagem ISO.....	83
5. CONCLUSÕES E TRABALHOS FUTUROS.....	87
5.1 Trabalhos Futuros.....	88
6. REFERÊNCIAS.....	90
7. ANEXOS.....	95

ÍNDICE DE ILUSTRAÇÕES

Ilustração 1: Entrada automatizada no ambiente chroot usando o script mount.sh...	54
Ilustração 2: Desmontagem automatizada do sistema de arquivos utilizando o script umount.sh.....	55
Ilustração 3: Navegador Mozilla Firefox sendo executado no Live-CD.....	61
Ilustração 4: phpMyAdmin sendo executado no Live-CD.....	66
Ilustração 5: Dia Diagrama sendo executado no Live-CD.....	67
Ilustração 6: IDE de desenvolvimento Geany sendo executado no Live-CD.....	69
Ilustração 7: DBDesigner sendo executado no Live-CD.....	70
Ilustração 8: Glade 3 sendo executado no Live-CD.....	71
Ilustração 9: Desenvolvimento da interface de usuário do Painel do Live-CD.....	75
Ilustração 10: Painel do Live-CD - Tela que oferece ao usuário o acesso a aplicativos de modelagem.....	77
Ilustração 11: Painel do Live-CD - Possibilidade de realizar algumas configurações no sistema.....	78
Ilustração 12: Menu do XFCE, após a tradução dos itens.....	80
Ilustração 13: Amostra do Splash Screen durante a emulação do Live-CD no aplicativo VirtualBox.....	82
Ilustração 14: Assistente para criação da Máquina Virtual - Passo 1.....	84
Ilustração 15: Assistente para criação da Máquina Virtual - Passo 2.....	85
Ilustração 16: Tela principal do aplicativo VirtualBox mostrando a lista de Máquinas Virtuais registradas.....	86

Índice de Tabelas / Quadros

Tabela 1: Esquema de partição do HD.....	45
--	----

ABREVIATURAS

AGP	<i>Accelerated Graphics Port</i>
ALFS	<i>Automated Linux From Scratch</i>
BLFS	<i>Beyond Linux From Scratch</i>
CSS	<i>Cascading Style Sheets</i>
CD	<i>Compact Disk</i>
CPU	<i>Central Processing Unit</i>
DVD	<i>Digital Video Disc</i>
GCC	<i>GNU Compiler Collection</i>
GNOME	<i>GNU Network Object Model Environment</i>
GNU	<i>GNU's Not Unix</i>
GPL	<i>General Public License</i>
GUI	<i>Graphical user interface</i>
GTK	<i>The GIMP Toolkit</i>
HD	<i>Hard Disk</i>
HTML	<i>HiperText Markup Language</i>
HTTP	<i>Hypertext Transfer Protocol</i>
ID	<i>Identification Digit</i>

IDE	<i>Integrated Development Environment</i>
IP	<i>Internet Protocol</i>
KDE	<i>K Desktop Environment</i>
LFS	<i>Linux From Scratch</i>
PC	<i>Personal Computer</i>
PHP	<i>Hypertext Preprocessor</i>
RAM	<i>Read Only Memory</i>
RPM	<i>Red Hat Package Manager</i>
SMP	<i>Symmetric Multi Process</i>
UML	<i>Unified Modeling Language</i>
USB	<i>Universal Serial Bus</i>
XML	<i>Extensible Markup Language</i>
XSL	<i>eXtensible Stylesheet Language</i>

RESUMO

Atualmente todo usuário de computador busca facilitar seu trabalho, e para isso faz proveito dos recursos oferecidos pelo sistema computacional. Uma das formas em que isso pode ser obtido é através da customização do sistema operacional, e em base é este o assunto tratado no trabalho, de forma profunda e acompanhando a construção de uma distribuição Linux Live-CD. O protótipo apresentado possui recursos específicos e limitados sendo destinado ao uso de professores e alunos dos cursos de Ciência da Computação e Sistemas de Informação da Unochapecó, mas ao final fornecerá uma visão para atender outras necessidades e até mesmo apelos mercadológicos.

Palavras-chave: Customização; Live-CD; Software livre;

ABSTRACT

Currently every computer user works towards making easier his work, and to accomplish that benefits from the resources provided by the computational system. One of that ways of achieving that goal can be done with the customizing of the operational system, and basically this is the main subject discussed on this paper, carefully detailed and featuring instructions on building of a Linux Live-CD distribution. The prototype presented has specific limited resources made for use by professors and students in Computer Science and Informational Systems courses at Unochapecó, but at the end will provided an overview to attend for other needs and even business' projects.

Keywords: Customization; Live-CD; Free software;

1. INTRODUÇÃO

O que tem se observado nos dias atuais é um crescimento no uso de software livre em empresas privadas, governamentais, no meio acadêmico e muito mais entre usuários domésticos. Isso acontece devido ao avanço e aperfeiçoamento destes sistemas de software, que em muitos segmentos já ultrapassam os proprietários em termos de qualidade, disponibilidade e usabilidade.

A customização de uma distribuição Linux Live-CD tem como objetivo agilizar o trabalho dos professores e alunos para um melhor aproveitamento dos recursos computacionais. Este trabalho enfatiza a personalização de uma distribuição acrescentando recursos úteis para disciplinas que são ministradas em laboratórios de informática e conseqüentemente necessitam de ferramentas, podendo estas ser de livre licença.

1.1 Organização do Projeto

Este projeto está dividido em vários capítulos para facilitar o melhor entendimento do conteúdo proposto. Segue abaixo uma pequena descrição do que vai ser abordado em maiores detalhes no decorrer deste.

- ♦ No capítulo um temos a parte introdutória do projeto que serve para explicar ao leitor de uma forma sucinta o que vai ser retratado ao decorrer do projeto.
- ♦ O capítulo dois trata com brevidade de Software livre, Sistemas operacionais e Distribuições Linux.
- ♦ No terceiro capítulo é apresentado o Linux Live-CD, características,

vantagens e desvantagens deste tipo de distribuição além das principais disponíveis atualmente. Traz também informações referentes à construção e customização de distribuições, mesclando conceitos e exemplos dos métodos mais utilizados.

- ♦ No quarto capítulo tem-se o processo de desenvolvimento do protótipo, a listagem das ferramentas utilizadas, os métodos utilizados em fim, a descrição do que foi feito na prática.
- ♦ No capítulo cinco está presente a descrição das conclusões obtidas com o projeto.
- ♦ O sexto capítulo traz a lista das referências utilizadas para compor este trabalho.
- ♦ Finalmente no capítulo sete estão os anexos.

2. LINUX

Segundo Maxwell (2000, p. 1) Linux é um sistema operacional derivado do Unix¹, sob uma licença GPL². Desta forma é importante definir neste momento o que é Sistema Operacional e Software Livre.

2.1 Software Livre

Software livre, segundo Hexsel (2006), é o software disponível para que qualquer pessoa possa usá-lo, modificá-lo, aperfeiçoá-lo e distribuí-lo, seja em sua forma original ou modificada, gratuitamente ou cobrando uma taxa, para qualquer um e em qualquer lugar, sendo que o código fonte e possivelmente os binários devem permanecer livre e disponíveis. O software livre, ao contrário do software proprietário gratuito, permite ao desenvolvedor abrir o código fonte e adaptar o sistema ao meio em que necessita, sem se preocupar com ilegalidade nesses procedimentos.

Para que essas liberdades sejam reais, elas devem ser irrevogáveis, caso o desenvolvedor do software tenha o poder de revogar a licença, mesmo que não tenha dado motivo, o software não é livre.

No tempo em que não existia a separação entre software livre e não-livre, os desenvolvedores trocavam código-fonte com liberdade, adaptando-os às necessidades. O início da comercialização de licenças de software ocasionou o fim da distribuição do código-fonte junto com o formato binário. Diante disso, usuários e

1 Unix: é um sistema operacional desenvolvido em 1969 por Ken Thompson e Dennis Ritchie nos laboratórios da AT&T Bell.

2 GPL: acrônimo (agrupamento das iniciais de várias palavras) de General Public License: licença que protege o direito de liberdade do software livre. Website: <http://www.gnu.org>.

hackers³ de todas as partes do mundo começaram a mobilizar-se contra esta idéia e alguns movimentos surgiram, como por exemplo, a Free Software Foundation que foi fundada por Richard Stallman no ano de 1983. Este grupo de pessoas defendia a livre execução, modificação, distribuição e redistribuição de software sem restrições, sendo que isso perdura até os dias de hoje. Trata-se em resumo da lei do software livre (ENEC, 2004).

As leis do software livre, segundo Campos (2006) também podem ser definidas como “Leis da Liberdade”:

- ♦ A liberdade de executar o programa, para qualquer finalidade.
- ♦ A liberdade de acesso ao código-fonte para estudar como o programa funciona e adaptá-lo conforme suas necessidades.
- ♦ A liberdade de redistribuir cópias a fim de ajudar a comunidade.
- ♦ Liberdade de aperfeiçoar o programa e lançá-lo de modo que toda a comunidade se beneficie.

A possibilidade de uma comunicação rápida e efetiva dos desenvolvedores e usuários deu-se, em parte, pelo início da popularização da Internet e uma posterior disseminação do modelo de desenvolvimento do software livre. Este foi adotado por empresas, governos e usuários domésticos.

2.2 Sistema Operacional

Segundo Tanenbaum (2004), é difícil reconhecer de modo preciso o que é um sistema operacional, porque ele realiza basicamente duas funções não relacionadas, estender a máquina e gerenciar recursos. Como uma máquina estendida o sistema operacional fornece uma variedade de serviços que os programas podem obter usando instruções especiais, e como um gerenciador de recursos mantém o controle

³ Hackers: indivíduos que elaboram e modificam software e hardware de computadores, seja desenvolvendo funcionalidades novas, seja adaptando as antigas.

sobre quem está usando o sistema, suas requisições e o conflito de requisições que podem ocorrer entre diferentes programas de usuários.

Em outro conceito, “sistema operacional é um programa que atua como intermediário entre o usuário e o hardware de um computador. O propósito de um sistema operacional é propiciar um ambiente no qual o usuário possa executar programas de forma conveniente e eficiente” (SILBERSCHATZ; GALVIN; GAGNE, 2001).

Em resumo, sistema operacional é um conjunto de rotinas executadas pelo processador de maneira semelhante a um software convencional conhecido. A diferença é que em um sistema operacional as rotinas são executadas concorrentemente e podem ocorrer a qualquer momento, diferente de um software comum que executa as rotinas de forma linear.

Todos os sistemas operacionais possuem conceitos básicos, alguns deles estão brevemente descritos abaixo.

- ♦ Processo: “Um processo é basicamente um programa em execução” (TANENBAUM, 2004). Segundo Tanenbaum (2004) processos podem criar um ou mais processos, sendo que todos eles são identificados por um código (PID) e podem se comunicar um com outro.
- ♦ Gerenciamento de memória: É o controle realizado pelo sistema operacional, permitindo que múltiplos programas permaneçam na memória e garantindo a proteção destes.
- ♦ Entrada e saída: Dispositivos de entrada e saída são os periféricos do computador (teclado, monitor, impressora). Estes são gerenciados pelo sistema operacional e permitem ao usuário interagir com o sistema computacional e verificar os resultados depois do trabalho realizado (TANENBAUN, 2004).
- ♦ Arquivos: É responsabilidade do sistema operacional manter a organização dos

arquivos. Em geral é fornecido o conceito de diretório como um modo de agrupar arquivos. Cada diretório pode conter vários outros diretórios, e este conjunto de diretórios, organizado de forma hierárquica, recebe o nome de sistema de arquivos (TANENBAUN, 2004).

Sistemas operacionais de linhagem diferentes tratam os atributos dos arquivos, além de muitas outras características, de maneira diferente. Por exemplo os atributos que definem o dono de um arquivo em um sistema baseado em Unix é diferente dos de sistema Windows.

- ♦ Interpretador de Comandos (Shell): O interpretador de comandos não é parte do sistema operacional, mas faz uso intensivo de muitos aspectos dele. “Ele é também a interface principal entre o usuário à frente de seu terminal e o sistema operacional, a menos que o usuário esteja usando uma interface gráfica de usuário” (TANENBAUM, 2004).

Existem diversos Shells e através deles o usuário pode executar uma infinidade de comandos, como exemplos de Shell para sistemas baseados em Unix podemos citar o *sh* (Bourne SHell), o *bash* (Bourne Again SHell), o *csch* (C shell), o *tsch* (Turbo C Shell) e o *ksh* (the Korn shell). O *sh* é o Shell original ainda usado em sistemas Unix e ambiente relacionados, sendo um programa simples e com poucos recursos. O *bash* emula o *sh* com muitos recursos adicionais e é o Shell padrão do GNU⁴. O *csch* é dirigido à programadores, pois a sintaxe deste Shell é dirigida para a programação C. O *tsch* é uma derivação do *csch*, com a interface mais amigável, uma velocidade maior e com mais recursos. E o *ksh* é um Shell extremamente complicado para usuários iniciantes, baseado no *sh* o *ksh* é apreciado mais por pessoas que viveram o mundo Unix (MANDRIVA CONECTIVA, 2005).

⁴ GNU: acrônimo recursivo para GNU's Not Unix, é um projeto iniciado em 1984 com objetivo de criar um sistema operacional totalmente livre.

2.3 Introdução ao Linux

O Linux originou-se a partir da cultura Unix, este criado em meados do anos 70, quando a era dos computadores Desktop era desconhecida.

Historicamente, o problema do Unix é a dificuldade dos programadores que querem trabalhar com ele fora de um ambiente corporativo ou universitário. Embora atualmente existam versões deste sistema para PC, estas eram menos poderosas e a aquisição de uma distribuição desse sistema certamente era oneroso. Essa dificuldade possibilitou o surgimento do Linux como um sistema operacional amplamente disponível.

Como outras grandes idéias, o Linux teve um começo modesto. Baseando-se no sistema operacional Minix, trabalho de Andrew Tanenbaum, o estudante Linus Torvalds, na Universidade da Finlândia começou a trabalhar em um sistema operacional próprio, e lançou em agosto de 1991 seu primeiro protótipo (NORTON; GRIFFITH, 2002, p. 5).

A possibilidade de examinar e contribuir com o projeto foram dados aos desenvolvedores do mundo inteiro quando Linus Torvalds publicou gratuitamente seus resultados na lista comp.os.minix. Isso permitiu a germinação de um sistema de desenvolvimento de software com código-fonte aberto, onde as alterações eram publicadas na internet, permitindo em pouco tempo aproveitar o intelecto de muitas pessoas (NORTON; GRIFFITH, 2002, p. 5).

A primeira versão oficial do Linux (0.0.2) foi liberada em 5 de outubro de 1991, esta contemplava a execução de programas como o Bash e o GCC. Como já citado, o Bash é um Shell e o GCC⁵ é um conjunto de compiladores de linguagens de programação. A existência de limitações não intimidou os desenvolvedores, e já nas primeiras versões atraía a atenção à medida que o fenômeno Linux continuava crescendo. A primeira versão considerada estável foi a 1.0, e no momento da escrita

5 GCC: acrônimo de *GNU Compiler Collection*.

deste trabalho a versão do Kernel estável é a 2.6.18.1 (NORTON; GRIFFITH, 2002, p. 5).

Atualmente o Linux é um sistema operacional completo, por este motivo torna-se um grande concorrente de sistemas comerciais, e Linus Torvalds mantém o controle sobre o desenvolvimento do Kernel sendo que seu código-fonte está em constante aperfeiçoamento. Novas versões com novos recursos e correções são lançadas em pequenos intervalos de tempo, ao contrário de sistemas proprietários.

Parece que dificilmente um dia se passa sem que se ouça alguma coisa a respeito de uma empresa que anuncia estar trabalhando com Linux. Até parece a corrida do Linux: um estado de rápido crescimento acompanhado de mudanças em um ritmo nunca visto antes. Essas mudanças certamente são boas para o usuário final, oferecendo avanços e oportunidades que não existiam antes (NORTON; GRIFFITH, 2002, p. 6).

Segundo Maxwell (2000, p. 1) o Linux apresenta tudo o que se espera de um sistema operacional moderno, como:

- ◆ Multitarefa realmente preemptiva, incluindo suporte total para múltiplos usuários.
- ◆ Proteção de memória.
- ◆ Memória virtual.
- ◆ Suporte para máquinas de multiprocessamento simétrico (SMP⁶) assim como para máquinas mais comuns equipadas com um único processador.
- ◆ Compatibilidade com o POSIX⁷.
- ◆ Funcionamento em rede.
- ◆ Interface gráfica com o usuário e ambiente de área de trabalho.
- ◆ Velocidade e estabilidade.

O Linux apresenta-se como um software livre de qualidade e como uma alternativa atraente. Pelo fato de estar em constante aperfeiçoamento, o Linux está

6 SMP: acrônimo de *Symmetric Multi Process*: máquinas que possuem duas ou mais unidades de CPU.

7 POSIX: é o nome de uma família de normas relacionadas definidas pelo IEEE e designada formalmente por IEEE 1003. A designação internacional da norma é ISO/IEC 9945.

se tornando um Sistema Operacional para o público em geral e os recursos da interface gráfica também caminham em direção à satisfação do usuário doméstico.

O Linux é o Kernel, o núcleo do sistema operacional. O sistema completo, incluindo ferramentas e interface é geralmente denominado de Distribuição Linux, o que é o assunto da próxima sessão (MAXWELL, 2000, p. 2).

Ainda conforme Alecrim (2004), Kernel é uma conjunto de arquivos escritos em linguagem C e linguagem Assembly, que constituem o núcleo do sistema operacional. No Kernel definem-se as funções para operações com periféricos (mouse, discos, impressoras, memória, etc.), ele representa a camada mais baixa de interface com o hardware, gerenciando de maneira transparente todos os recursos do sistema computacional. “Cabe ao Kernel as tarefas de permitir que todos os processos sejam executados pela CPU e permitir que estes consigam compartilhar a memória do computador” (ALECRIM, 2004).

O Kernel é a parte mais importante do sistema operacional, pois ele permite que as funções de comunicação com os periféricos que estão nele inclusas sejam reaproveitadas sempre que um novo programa é feito, isso economiza muito trabalho e diminui o risco da ocorrência de erros. Na existência de periféricos ou elemento de um sistema computacional que o Kernel não dá suporte, faz-se necessário escrever uma nova interface para eles, os chamados drivers⁸ ou módulos (ALECRIM, 2004).

2.4 Distribuições Linux

Segundo Danesh (1999, p. 8) sistemas operacionais proprietários como o Windows, Mac OS, Solaris contemplam um conjunto de utilitários, aplicativos, drivers, etc. Neste segmento não existe tecnicamente um produto sem o seu conjunto completo de software e ferramentas, pois qualquer componente adicionado

⁸ Drivers: são programas que possibilitam a comunicação entre o sistema operacional e dispositivos periféricos ligados a um computador.

a este pelo usuário não é considerado parte do sistema. Tratando-se de Linux, não encontramos estes conjuntos desta maneira acoplados, esse diferencial possibilitou a criação de conjuntos diversificados, para atender as diferentes necessidades, a isto deu-se o nome de distribuições.

As distribuições, também chamadas de distros, podem ser produzidas com diferentes versões do Kernel, conter ferramentas, aplicativos, utilitários e drivers diversificados e podem oferecer programas de instalação/atualização específicos do sistema (DANESH, 1999, p. 9).

Ao customizar uma distribuição Linux ganha-se tempo reutilizando trabalhos anteriores, aumentando radicalmente a velocidade de desenvolvimento. “Qualquer pessoa ou empresa com tempo e conhecimentos suficientes pode desenvolver uma distribuição. O mais comum é usar uma distribuição já existente como ponto de partida e ir incluindo novos recursos a partir daí” (MORIMOTO, 2006a).

2.4.1 Principais Distribuições

A maioria das distribuições Linux está disponível na internet, podendo ser obtidas gratuitamente a partir de websites HTTP/FTP, ou adquiridas de organizações que as forneça em mídias de CD/DVD ou pendrives.

Existem muitas distribuições Linux disponível atualmente, como exemplo temos o Slackware, Red Hat, Fedora, Debian e SuSE.

- ♦ **Slackware**

Atualmente é a única sobrevivente das distribuições Linux que surgiram entre 1991 e 1993, desta derivaram algumas distribuições conhecidas, como o Vector, Slax e o College.

Criada por Patrick Volkerding, completou em julho de 2006 13 anos. O Slackware tem como objetivo preservar as tradições dos sistemas Unix, provendo

um sistema estável, organizado onde quase tudo pode ser feito alterando diretamente os scripts de configuração ou utilizando ferramentas simples. Isso permite uma flexível customização e controle do sistema (MORIMOTO, 2006a).

O Slax é um Live-CD⁹, o Vector é uma distribuição otimizada para micros antigos e o College é uma distribuição desenvolvida com o objetivo de ser fácil de usar e está focada ao público estudantil (VIEIRA, 2006).

- ♦ Red Hat e Fedora

O Red Hat surge um ano depois do Slackware, com o objetivo de facilitar a configuração e de automatizar do sistema, incluindo várias ferramentas de configuração e controle de pacotes de aplicativo de software.

Segundo Norton e Griffith (2002, p. 10), uma das contribuições mais importantes para a comunidade Linux foi o RPM¹⁰ e também o instalador do sistema operacional que já foi reaproveitado, com algumas pequenas alterações, para servir a outras distribuições.

O Red Hat deu origem ao Mandrake, SuSE, Conectiva, Fedora e uma série de distribuições menores. O Fedora Core, por exemplo, tornou-se símbolo da ruptura da Red Hat quando este anunciou sua entrada no mercado de servidores corporativos e, atualmente é a versão livre do Red Hat.

De acordo com Morimoto (2006a):

As distribuições derivadas do Red Hat são junto com o Debian e derivados as mais usadas em servidores. O Fedora, Red Hat e SuSE possuem também uma penetração relativamente grande nos desktops nas empresas, enquanto o Mandrake tem o maior público entre os usuários domésticos.

A Mandriva, originalmente conhecida como Mandrakesoft, é o resultado da fusão de diversos pioneiros do código aberto como a Mandrakesoft na França, a Conectiva no Brasil, a Edge IT na França e a Lycoris nos Estados Unidos.

⁹ Live-CD: distribuição que roda direto de uma mídia de CD/DVD, será abordada adiante.

¹⁰ RPM: acrônimo de *Red Hat Package Manager*: ferramenta que torna a instalação e a remoção de software em Linux mais fácil e menos sujeita a erros.

- ♦ Debian

Segundo Morimoto (2005a), o Debian deu origem à quase metade das distribuições atualmente em uso, sendo Knoppix, o Lycoris e o Ubuntu exemplos.

Ao contrário de outras distribuições Linux conceituadas, o Debian é produzido por uma equipe de voluntários, não possuindo uma organização comercial patrocinadora (DANESH, 1999, p. 10).

Segundo Morimoto (2006a), as principais características do Debian são a grande quantidade de pacotes disponíveis e o apt-get, um eficiente aplicativo para gerenciamento de pacotes.

A utilização do Debian está reservada a usuários mais avançados, mas algumas distribuições derivadas dele, como o Ubuntu, o Mepis, e o Kurumin são desenvolvidas com foco na facilidade de uso e incluem ferramentas e assistentes de configuração (MORIMOTO, 2006a).

- ♦ SuSE

As primeiras versões foram baseadas no SLS (primeira distribuição Linux que se tem notícia), depois suas ferramentas foram migradas para o Jurix que era baseado no Slackware. A partir de 1998 o SuSE passou a utilizar pacotes RPM do Red Hat entre outras ferramentas. O SuSE possui um painel de controle chamado Yast, que reúne muitas ferramentas de configuração, facilitando bastante a administração do sistema (MORIMOTO, 2006a).

Conforme Morimoto (2006a) em 2003, a SuSE foi adquirida pela Novell, dando origem a uma solução comercial chamada Novell Desktop e a um projeto comunitário chamado de OpenSuSE.

2.4.2 Qual é a melhor distribuição?

Para Campos (2005) a resposta desta questão pode vir de várias formas,

podendo ser genérica ou específica, aberta ou qualificada. O usuário estreante quer saber qual a distribuição adequada para instalar em seu computador pessoal. O administrador de sistema, como um conhecedor de diferentes arquiteturas, quer saber qual é a melhor distribuição para seu servidor. O usuário que não conseguiu fazer funcionar o hardware adequadamente pergunta qual é a melhor para funcionar com periféricos de projeto exótico.

A maior parte das distribuições de Linux consegue disponibilizar o mesmo conjunto de serviços, embora às vezes de maneiras bem diferentes. Algumas já vêm com todos os aplicativos e serviços incluídos nos CDs de instalação, outras exigem downloads e instalações adicionais. Algumas se distinguem por uma ênfase em aspectos específicos do sistema, como a facilidade de configuração, a quantidade de aplicativos, a segurança, a personalização e vários outros (CAMPOS, 2005).

Segundo CAMPOS (2005), não é possível responder de forma ampla qual é a melhor distribuição Linux, a resposta depende do que o usuário pretende fazer com o sistema, do nível de conhecimento e interesse que possui, e até mesmo de sua atitude em relação a algumas questões políticas e filosóficas. A melhor distribuição sempre será a que atender mais perfeitamente as necessidades, para isso o primeiro passo é saber o que recomendam as pessoas/websites/revistas a quem o usuário pretende recorrer na hora da dificuldade, e depois fazer uma lista de perguntas sobre os diversos aspectos que podem ser de seu interesse na hora de escolher uma distribuição. Algumas perguntas que o usuário deve tentar responder são:

- ◆ Esta distribuição suporta todo o meu hardware?
- ◆ Ela inclui os pacotes de software de que necessito?
- ◆ O processo de instalação e configuração está de acordo com minhas aptidões?
- ◆ Ela tem documentação e treinamento em um idioma que eu entendo?
- ◆ O suporte prestado (gratuito ou pago) atende minhas necessidades?
- ◆ Há uma comunidade de usuários da qual eu possa participar?
- ◆ Ela lança atualizações de segurança quando necessário?

- ♦ Ela continuará sendo atualizada?
- ♦ Ela é livre? É grátis? O preço é aceitável?

Sob esses critérios e diversos outros, o usuário pode escolher a distribuição que melhor atenderá suas necessidades.

Um problema encontrado após a instalação de certas distribuições Linux é o uso de espaço em disco que fica sendo utilizado por pacotes que raramente ou jamais serão utilizados. Seria interessante se o usuário pudesse escolher os pacotes a serem incluídos em uma distribuição antes de obtê-la, isso adaptaria a distribuição às reais necessidades do usuário além de economizar tempo e uso da conexão de internet. Com a idéia voltada para este segmento, o website installinux.com lançou um serviço que permite uma seleção customizada de pacotes (SILVA, 2006). O processo funciona como um *wizard*, onde o usuário seleciona uma distribuição (CentOS, Debian, Fedora, OpenSuse, Scientific, SuSE ou Ubuntu), sua versão e arquitetura, configura a interface de rede, o fuso-horário, o idioma, acrescenta um usuário, faz a seleção de pacotes, configura o esquema de particionamento do HD, etc. e ao concluir o processo o usuário tem acesso a um link para fazer o download da imagem ISO, pronta para ser gravada na mídia. Algumas distribuições como o Ubuntu, poderão ser executadas como Live-CD.

2.4.3 Interfaces Gráficas do Linux

Nos dias atuais, quase todo mundo lida com computadores, seja para trabalho, estudo ou lazer. Quando o usuário se posiciona para utilizar o computador, a primeira coisa que ele espera é poder operar em uma interface gráfica, ou seja, interagir com o mouse, iniciar programas clicando nos ícones, manipular janelas e muito mais.

Usuários mais experientes de sistemas operacionais baseados em Unix, têm prática na utilização de comandos digitados (o chamado "modo texto") para executar tarefas no computador, mesmo porque, em alguns casos, somente isso é necessário. No entanto, se você quer ver imagens, vídeos, utilizar o mouse e aplicações gráficas, é necessário o uso de um ambiente gráfico (ALECRIM, 2006b).

“O programa responsável pela parte gráfica do Linux é o XFree, que é um servidor de janelas” (PAULA, 2002). O XFree contém especificações de como criar e manipular as janelas do ambiente, e oferece ferramentas aos gerenciadores de janela para a implementação de uma interface gráfica. Conforme Alecrim (2006b), além do XFree existe outros servidores de janelas, um deles bastante conhecido recebe o nome de X.org.

“Gerenciadores de janelas são programas que controlam como as janelas devem ser apresentadas, a forma como os botões se posicionam, quais são suas funções, o que pode acontecer dentro de uma janela e coisas do gênero” (PAULA, 2002).

Quando o usuário solicita a execução de algum programa no ambiente gráfico, por exemplo, ele não precisa se preocupar com as especificações das janelas pois o XFree abre uma janela e “coloca” o programa dentro dela. Pelo fato do XFree possuir código-fonte aberto, surgiram várias interfaces gráficas para o sistema e, ao contrário do sistema operacional Windows o usuário pode optar pela interface que mais o agrada em termos de beleza e facilidade de uso. (PAULA, 2002). Os gerenciadores de janelas que estão presentes na maioria das distribuições Linux são o KDE e o GNOME, ambos disponíveis sob licença GPL. Existem vários outros, como exemplo o AfterStep, o Enlightenment, o WindowMaker, o IceW, o BlackBox, Fvwm90 e o XFCE.

O KDE foi desenvolvido com base no Qt, um conjunto de bibliotecas gratuito para o desenvolvimento de aplicações gráficas de código-fonte aberto. Além de ser um gerenciador de janelas eficiente, também oferece aplicativos muito interessantes como o editor de texto Kword, o leitor de arquivos em PDF KPDF, etc (ALECRIM,

2006a).

O GNOME é outro ambiente muito utilizado, “um trabalho iniciado por Miguel de Icaza, em 1997. O objetivo desse projeto é oferecer aos usuários do Linux e de outros sistemas baseados no Unix, um ambiente de desktop completo e fácil de usar” (ALECRIM, 2006b). O GNOME é totalmente personalizável” (ALECRIM, 2006b). O usuário pode alterar o idioma, instalar novos temas, além de efetuar alterações em cores, fontes, menus, papel de parede, proteção de tela, enfim.

Conforme Alecrim (2006a) tanto o KDE como o GNOME exigem bons recursos de hardware para uma execução plena de tudo o que oferecem. Para computadores antigos o ideal é utilizar gerenciadores de janelas mais simples, como o BlackBox, WindowMaker ou XFCE.

3. LINUX LIVE-CD

Tradicionalmente, qualquer sistema operacional para ser usado precisa ser instalado no HD¹¹. Para isso é necessário dar o boot¹² por uma mídia de CD/DVD ou outra e efetuar o processo de instalação.

Live-CD é uma distribuição que roda diretamente a partir de uma mídia de CD/DVD, não implicando na instalação propriamente dita do sistema para sua utilização. Mesmo não havendo instalação, o sistema permite a comunicação com todos os dispositivos de hardware, podendo assim o usuário conectar-se á internet, executar aplicações de escritório, etc. Um requisito fundamental para utilizar uma distribuição Live-CD, é certamente que o usuário tenha um leitor de CD/DVD e configure o computador para inicializar a partir deste (MARQUES, 2007).

Uma fato muito importante das distribuições Linux Live-CD é a possibilidade de gerar novas distribuições a partir destas, permitindo que até usuários pouco experientes com o sistema operacional possam criar suas próprias distribuições personalizadas.

As distribuições Live-CD são muito importantes para sociedade desenvolvedora de software livre pois permite que o usuário teste o sistema sem ter que sobrescrever o atual sistema operacional que roda em seu computador, podendo assim despertar novos interesses e auxiliar na quebra da hegemonia do sistema operacional Windows entre usuários domésticos.

¹¹ HD: acrônimo de *Hard Disk* (disco rígido): é a memória física do computador.

¹² Boot: é o processo de inicialização do microcomputador.

3.1 Características

Para Morimoto (2006a), a base de tudo em um Live-CD está no módulo do Kernel chamado *cloop*, um módulo que permite que o sistema rode a partir de um sistema compactado gravado em uma mídia de CD/DVD. A compressão dos arquivos melhora a taxa de transferência e conseqüentemente o desempenho do sistema.

O módulo *cloop* foi originalmente desenvolvido por Andrew Morton, que na época achou que o módulo não teria utilidade e o descartou. Tempo depois ele foi redescoberto pelo Klaus Knopper, que acabou por utilizá-lo como um dos componentes base do Knoppix. “O módulo *cloop* engana o Kernel, fazendo-o pensar que está acessando uma partição no disco rígido” (MORIMOTO, 2006b). Segundo Morimoto (2006b), quando é preciso abrir um arquivo, ele descompacta a porção correspondente da imagem, lê o arquivo e entrega os dados já “mastigados” para o Kernel.

Girando em uma velocidade constante, a taxa de leitura de CD-ROM com velocidade de 40x chega a 3 MB/s¹³ nas trilhas internas e 6 MB/s nas trilhas mais externas. Como a mídia de CD é gravada a partir do centro, na maior parte do tempo a taxa de transferência fica entre 3, 4 e 5 MB/s. Algumas distribuições Live-CD podem ser transferidas e executadas através de um pendrive ou cartão de memória. As distribuições Linux que são executadas a partir de dispositivos USB são chamadas de Live-USB. Em computadores com USB 1.0 a taxa de transferência chega a 2,5 MB/s, e com USB 2.0 pode chegar a 60 MB/s.

“Se não fosse o sistema de compressão, os Live-CDs descendentes do Knoppix seriam três vezes maiores e três vezes mais lentos ao rodar do CD, o que os tornariam sistemas muito menos atrativos” (MORIMOTO, 2006a).

Em geral, ao iniciar uma distribuição Live-CD, é mostrada uma tela de boas

¹³ MB/s: unidade de medida para representar a uma certa quantidade de dados por unidade de tempo.

vindas e um campo de texto onde o usuário pode fornecer parâmetros para o boot, como por exemplo a resolução de vídeo que será utilizada no ambiente gráfico. Logo depois o Kernel fica responsável de inicializar o hardware do computador e criar a *ramdisk* em uma pequena parte de memória RAM, aonde ficará armazenado dentre outros, alguns arquivos de configuração.

Conforme Nilsen (1999), *ramdisk* é uma porção da memória principal alocada para ser usada como uma partição de um disco secundário, ou seja, é um espaço na memória manipulado por utilitários e tratado com se fosse um HD.

Na etapa posterior entra em cena os aplicativos de detecção, que se encarregam de detectar placa de som, vídeo, rede, modem e outros periféricos. Sem a necessidade de intervenção do usuário, em um curto espaço de tempo o sistema detecta, configura e ativa os periféricos suportados no microcomputador, e quando disponível configura também a conexão com Internet.

A utilização da *ramdisk* é importante sempre que se tem determinados arquivos que são utilizados constantemente. Alocando eles na memória principal, desde que esta possua acesso mais rápido que o HD, obtêm-se um considerável aumento no desempenho (NILSEN, 1999).

Como a *ramdisk* é o único repositório de armazenamento dos dados, ela cresce na medida em que novos programas são instalado e configurados, por isso os requisitos de memória RAM para rodar o sistema variam de 128 MB a 256 MB para grande parte das distribuições, e 512 MB para instalação de novos aplicativos. A instalação de novos aplicativos é possível graças a um módulo do Kernel chamado *UnionFS* (MORIMOTO, 2006a).

UnionFS é um sistema de arquivos unificador, que permite a união dos conteúdos atualizados de múltiplos diretórios mas mantém seu conteúdo físico original separado. O website do UnionFS descreve o sistema como útil para gerenciamento unificado da árvore de diretórios, união de conteúdo de dois CDs ou

pacotes de software, datagrids¹⁴, etc. O UnionFS permite a mistura de arquivos sem permissão de escrita com arquivos com permissão, bem como edição em qualquer parte. Para manter a semântica do Unix, o UnionFS trata a eliminação de duplicados, condições parciais de erro, etc (GIBBS, 2005).

No Knoppix, por exemplo, a implementação do UnionFS une a ramdisk com o sistema de arquivos somente-leitura do CD, de forma que o usuário possa modificar qualquer arquivo somente-leitura como se fosse gravável (GIBBS, 2005).

Um módulo do kernel similar ao UnionFS é o SquashFS. Este além da utilização na confecção de Live-CDs, pode ser utilizado como ferramenta de backup, pois permite a compressão de uma grande quantidade de dados. O GoblinX Linux é uma distribuição que faz uso do SquashFS (GOBLINX, 2006b).

Para melhorar o desempenho geral do sistema é recomendável ter disponível uma partição de *swap*, que pode ser criada automaticamente na instalação do sistema ou manualmente utilizando um software de particionamento.

“A memória *swap* (ou memória virtual) é um recurso usado por todos os sistemas operacionais atuais quando não existe memória RAM suficiente” (MORIMOTO, 2006a). “Como não é possível fazer memória *swap* no CD-ROM, o sistema é capaz de detectar e montar automaticamente partições de memória *swap* de outras distribuições Linux” (MORIMOTO, 2006a). Caso o sistema operacional instalado no HD for o Windows, o sistema tentará criar um arquivo de *swap* nesta partição, mas isso somente será possível se o formato da mesma for FAT32¹⁵.

Incompatibilidades podem travar o sistema, mas na maioria das vezes isso pode ser resolvido filtrando a detecção de dispositivos nas opções de boot.

¹⁴ Datagrid (computação em grade): é um modelo computacional capaz de processar uma alta taxa de processamento divididos em diversas máquinas, podendo ser em rede local ou rede de longa distância. Datagrid permite armazenar dados de forma distribuída, sobre uma grid.

¹⁵ FAT32: é o sistema de arquivos usado pelo MS-DOS e outros sistemas operacionais baseados em Windows para organizar e gerenciar arquivos.

3.2 Vantagens

As distros Live-CD possuem muitas utilidades, algumas vantagens estão abaixo listadas.

- ♦ Dispensa a instalação do sistema: O usuário pode usufruir do sistema Linux sem instalar no computador. Como exemplo o Knoppix, pioneiro no formato, que possui até 2 GB de software e nada disso precisa ser obrigatoriamente instalado.
- ♦ Portabilidade: Muitas destas distros podem ter o tamanho de um CD de 8cm, tornando-as compactas e portáteis. Como exemplo podemos citar o Slax, o Damn Small e algumas versões do Kurumin (TUXS, 2006).
- ♦ Mantém os dados do HD intactos: Quando estiver utilizando o Linux em um Live-CD o usuário poderá acessar as partições do HD local e poderá optar pelo acesso somente-leitura, isso garante que nenhum dado do HD seja alterado de maneira indesejada (TUXS, 2006).
- ♦ Utilizado para procedimentos de recuperação de dados e particionamento: Diversas distros tradicionais possuem um chamado CD SystemRescue, que não deixa de ser um Live-CD. Este inclui muitas ferramentas para reparar dados do HD, gerenciar e redimensionar uma grande variedade de partições, além de utilitários para espelhamento e cópia íntegra de discos (TUXS, 2006).
- ♦ Possibilidade do usuário conhecer e testar a distribuição: As distros Live-CD auxiliam na popularização do Linux, que não deixa nada a desejar em relação a distribuições instaláveis e até mesmo proprietárias, quanto a variedade de aplicativos. No Kurumin, por exemplo, o usuário pode testar a distribuição, e se preferir, poderá instalar no HD.
- ♦ Utilizado como ferramenta para quebra de senha: Distribuições específicas, como por exemplo a Austrumi, tem como finalidade zerar senhas do sistema

operacional Windows, inclusive a do usuário administrador, independentemente do tipo do sistema de arquivos (TUXS, 2006).

- ♦ Possibilidade do usuário criar sua própria distribuição: A maioria das distribuições permitem que o usuário personalize o sistema para posteriormente gerar uma nova distribuição.
- ♦ Utilizado como sistema para acesso seguro: Pelo fato do sistema do Live-CD rodar diretamente na memória, os dados são totalmente destruídos após a reinicialização do computador. Por este motivo os usuários podem utilizar um Live-CD para fazer acesso a websites de agências bancárias ou similar com maior segurança.
- ♦ Utilizado para execução de aplicações multimídia: A grande parte das distribuições possuem nativamente uma série de plugins¹⁶ para executar arquivos de audio, vídeo, streaming, CDs e DVDs. Existem distribuições especializadas no assunto, como por exemplo a Movix (TUXS, 2006).
- ♦ Utilizado para jogos: Algumas distribuições são especialmente criadas para o lazer, e trazem uma série de jogos e aplicações. Como exemplo tomamos o Kurumin Games, o Morphix e o GamesKnoppix.
- ♦ Utilizado como ferramenta para remoção de vírus: O usuário poderá bootar com o Live-CD para acessar uma partição do HD e eliminar vírus. Segundo Tuxs (2006), uma distro ideal equipada com anti-vírus é a INSERT¹⁷.
- ♦ Linux no pendrive: algumas distribuições podem ser executadas a partir de um pendrive ou cartão de memória, para isso é necessário que o hardware possua suporte para dar o *boot* através de dispositivos USB. O usuário pode instalar diretamente o sistema no pendrive como se fosse um HD, ou copiar a imagem de um Live-CD e usar o espaço excedente para armazenar arquivos.

¹⁶ Plugin ou plug-in: é um programa de computador que serve normalmente para adicionar funções a outros programas para prover alguma função particular ou muito específica.

¹⁷ INSERT: Acrônimo de INside SEcurity Rescue Toolkit. <http://www.insert.cd/>

- ♦ Montar computadores sem HD: como não é necessário um dispositivo de armazenamento físico para rodar Live-CD, a inclusão do HD é dispensado quando houver interesse de rodar apenas este tipo de distribuição e a quantidade de memória RAM garantir a não necessidade de uma partição de *swap* (MORIMOTO, 2005b).

3.3 Desvantagens

- ♦ O tempo de busca maior: O sistema pode ficar mais lento pois a taxa de leitura do CD geralmente é menor que do HD. Alguns Live-CDs oferecem uma opção que permite aos computadores dotados de grande quantidade de memória, copiar todo o conteúdo do CD para a *ramdisk*, isso melhora radicalmente o desempenho do sistema.
- ♦ Destruição das informações quando o computador é reiniciado: Sempre que o microcomputador for desligado ou reiniciado e o sistema estiver rodando diretamente no CD, todas as informações serão apagadas, e o usuário perderá as alterações do sistema e arquivos. Para que isso não aconteça, é necessário que as informações pessoais sejam gravadas em uma partição no HD ou em repositórios remotos.
- ♦ Mais trabalho para o processador e necessidade de mais memória: o trabalho do processador passa a ser maior, pois as informações devem ser descompactadas antes de processadas, com isso torna-se necessário uma maior quantidade de memória para que haja um desempenho considerável.

3.4 Principais distribuições Live-CD

- ♦ Knoppix: É uma distribuição Linux alemã baseada do Debian criada por Klaus Knopper, que foi a pioneira se tratando de Live-CD. O Knoppix se tornou rapidamente uma das distribuições Live-CD mais usadas e deu origem a um

universo gigantesco de novas distribuições, incluindo o Kurumin (MORIMOTO, 2005a).

Para Morimoto (2006b), as vantagens do sistema Knoppix e derivados em relação a outras distribuições são: 1) Ele detecta e configura o hardware de maneira automatizada. 2) Possui o knx-hdinstall, um aplicativo que permite instalar o sistema no HD mantendo toda a configuração de hardware feita durante o boot. 3) É possível instalar qualquer um dos mais de 15.000 pacotes do Debian, usando o apt-get. 4) Como o conteúdo do CD é compactado, quase 2GB de programas se reduzem a um CD de 700MB, mais que o suficiente para uma distribuição. 5) É possível instalar drivers para vários tipos de hardware, programas binários e, até mesmo rodar aplicativos windows, através de um software de emulação chamado Wine.

Conforme Osxbook (2006), o Knoppix dá seus primeiros passos no Macintosh¹⁸ baseado em Intel, ele já roda nessa arquitetura porém não sabe-se ainda se todos os aspectos de hardware estão funcionando corretamente e o quanto levará para que isso ocorra.

Para quem já se acostumou com a idéia do Live-CD, pode parecer natural rodar, configurar o sistema e até mesmo instalar novos programas contando com a ausência do disco rígido, mas o aparecimento do Knoppix foi um verdadeiro marco dentro do mundo Linux (MORIMOTO, 2006a).

Uma coisa interessante é que o Knoppix mantém a estrutura Debian quase intacta, o que fez com que instalar o Knoppix no HD acabasse tornando-se uma forma alternativa de instalar o Debian. As distribuições derivadas do Knoppix muitas vezes vão além, incluindo novos componentes que tornam o sistema mais adequado para usos específicos (MORIMOTO, 2005a).

- ♦ Kurumin Linux: É a distro brasileira baseada no Debian e Knoppix, desenvolvida por Carlos E. Morimoto e mantida pela equipe do Guia do Hardware¹⁹ e colaboradores. Tornou-se rapidamente uma das distribuições Linux mais usadas

¹⁸ Macintosh ou Mac: é o nome dos computadores pessoais fabricados pela Apple Computer.

¹⁹ Guia do Hardware: <http://www.guiadohardware.net/kurumin>

no país (MORIMOTO, 2006c). Conforme Morimoto (2006c) o Kurumin tem como objetivo a facilidade de uso, sendo que os scripts, ferramentas de configuração, menus e outros são escritos diretamente no idioma Português do Brasileiro, ao invés de traduzidos, permitindo assim uma boa familiarização dos usuários. O Kurumin possui além da documentação completa em português, um conjunto variado de drivers, estes selecionados conforme o hardware comercializado no país. Kurumin Games e Kurumin Developer são exemplos de distribuições que se baseiam no Kurumin Linux.

- ♦ Ubuntu: em essência, Ubuntu significa “Eu sou porque você existe”, podendo existir diversas outras variações. A distro Ubuntu é sul-africana relativamente nova, amigável, moderna e seu principal objetivo é oferecer às pessoas, com qualquer nível de conhecimento, um sistema útil e fácil de usar. Para versões mais antigas do Ubuntu, o disco para instalação e o Live-CD eram distribuídos separadamente, já nas atuais é apenas um disco. Ubuntu é uma distribuição derivada do Debian, sendo que os projetos mais conhecidos que derivaram do Ubuntu são: a)Kubuntu: é a própria distro do Ubuntu, porém com o KDE como ambiente gráfico padrão (KUBUNTU, 2006). b)Edubuntu: foi desenvolvido para uso educacional, seu objetivo é levar o Ubuntu às escolas e futuramente às universidades. Possui ferramentas de escritório, navegador de internet além de uma série de aplicações educacionais (EDUBUNTU, 2006). c)Xubuntu: montada com software leves, objetivando maior desempenho em computadores com pouco poder de processamento e pouca memória (XUBUNTU, 2006).
- ♦ GoblinX: Distribuição Live-CD baseada no Slackware Linux. Caracteriza-se por ser uma distro amigável (GOBLINX, 2006a). O GoblinX tem suporte a vários idiomas e, o mais importante é que para habilitar um deles basta apenas um comando. Como utiliza um sistema de módulo, é permitido ao usuário acrescentar seus próprios programas, seja alterando a ISO original, seja enquanto usa o Live-CD. “Acrescentando módulos o usuário obtém uma distro

ainda mais completa e personalizada, além de poder virtualmente utilizar qualquer programa com o GoblinX” (GLOBLINUX, 2006a).

- ♦ Dreamlinux: “É um sistema modular e moderno, que pode ser utilizado diretamente pelo CD ou instalado, de maneira rápida e fácil, em seu computador” (DREAMLINUX, 2006a).

O Dreamlinux é uma distro baseada no Debian, no Knoppix e no Morphix, dos quais reúne os melhores programas e características para atender as necessidades do usuário no dia-a-dia. Possui ainda ferramentas de desenvolvimento próprias e de geração de distros, é o caso do MKDistro Maker, uma ferramenta simples e automatizada que permite ao usuário personalizar a distribuição.

- ♦ Outras distribuições: Outras distribuições bastante conhecidas: Morphix, Kalango, Famelix, Slax, Damn Small Linux, Big Linux, Mepis, Dizinha. Uma lista completa de distribuições Linux Live-CD pode ser obtida no website: <http://www.frozentech.com/content/livecd.php>.

3.5 Ferramentas para customizar uma distribuição

3.5.1 Introdução

Em muitos casos, o usuário obtém uma distribuição Linux de sua preferência, mas os aplicativos nela inclusos não atendem suas reais necessidades. Por isso muitos profissionais da comunidade software livre contribuem no desenvolvimento de soluções para esse problema.

Existem basicamente três maneiras para customizar uma distribuição:

- ♦ Montando a distribuição do zero.
- ♦ Alterando o conjunto de pacotes de uma distro pronta.

- ♦ Fazendo uso de ferramentas automatizadas.

3.5.2 Linux From Scratch

O LFS (Linux From Scratch) ou “Linux do nada” como também é chamado, é um projeto liderado por Gerard Beekmans que fornece instruções passo-a-passo de como construir distribuições Linux a partir do zero. Estas instruções reunidas formam o Livro Linux From Scratch (SCRATCH, 2006).

Primeiramente o usuário deve fazer download dos pacotes contendo o código-fonte dos programas mais comuns em um sistema Linux (Shell, editor, compilador etc.) e depois deve seguir as instruções do Livro LFS.

Para Beekmans (2005), a construção de um sistema LFS permite entender o funcionamento interno do Linux, como suas partes trabalham junto e como dependem entre si, e ao final fornece habilidade para personalizar o Linux conforme as necessidades.

Usando o LFS, o usuário pode criar um sistema Linux compacto, sem necessidade da inclusão de pacotes dispensáveis para a aplicação do usuário, resultando numa grande economia de espaço em disco e processamento (SCRATCH, 2006)

“O maior benefício da implementação do LFS é permitir que os usuários tenham mais controle sobre o sistema sem ter que confiar na implementação de outra pessoa (BEEKMANS, 2005)”. Outra vantagem é a possibilidade de analisar todo o código-fonte, permitindo que sejam aplicadas correções de segurança, resultando em um sistema adequadamente seguro (BEEKMANS, 2005).

Um sistema LFS é construído usando uma distribuição previamente instalada do Linux. Este é chamado de sistema anfitrião e será usado com ponto de partida, fornecendo os programas necessários, tais como um compilador, editor de vínculos e Shell. Estes programas geralmente já estão disponíveis no sistema anfitrião, caso

contrário, o usuário deverá escolher a opção “desenvolvimento” ou similar durante o processo de instalação do mesmo.

Dois capítulos da versão 6.2 do Livro LFS tratam da construção:

- ◆ Preparando a configuração

Engloba a descrição dos pacotes e patches necessários trazendo os links para obtenção destes, discute a instalação e configuração de um ambiente de trabalho próprio ao processo de montagem do sistema LFS e explica a instalação de certos pacotes que dão forma ao conjunto básico de desenvolvimento que será utilizado para construir o sistema real. A construção deste conjunto básico, também chamado de *toolchain*, é dividido em duas etapas, a primeira construirá um jogo de ferramentas novo e independente do sistema anfitrião e a segunda usa este jogo de ferramentas para acrescentar as demais ferramentas essenciais. Este último é tratado no livro como *dtoolchain*, e é a base de instalação do restante dos aplicativos, não mais dependendo o sistema LFS do sistema anfitrião, exceto do kernel. Ao final, todo este conjunto será descartado.

- ◆ Configurando o sistema LFS

Traz todas as instruções para completar a construção. Faz uso do aplicativo *chroot* que serve para entrar em um ambiente virtual e iniciar um novo Shell cujo o diretório raiz é definido como parâmetro. As instruções deste capítulo são muito precisas, a fim de minimizar as possibilidades de erros, e estas devem ser seguidas estritamente. Antes das instruções de instalação, cada página da instalação fornece informações sobre o pacote, incluindo uma descrição do seu conteúdo, o tempo aproximado da configuração, o espaço em disco requerido durante o processo de configuração, entre outros.

A distro Famelix GNU/Linux foi desenvolvida com a finalidade de oferecer um ambiente similar ao sistema operacional Windows XP, de tal maneira que facilite a migração de usuários do sistema operacional proprietário a um sistema livre. Esta é

exemplo de distribuição gerada através do Linux From Scratch, e utiliza elementos presentes em outras distribuições como Debian, Knoppix e Kurumin.

3.5.3 Remasterização

Quando o usuário já possui uma distribuição, seja em CD, em formato ISO ou já instalada no computador, é possível personalizar esta e gerar uma nova distro. Para isso o usuário precisa seguir uma série de passos, a este processo se dá o nome de remasterização.

De acordo com Morimoto (2006b), para customizar uma distribuição através da remasterização, o primeiro passo é descompactar a imagem em uma partição previamente montada do HD. Para alocar os arquivos da imagem descompacta e os da nova distribuição de 700MB, será necessário 3.5GB de espaço livre podendo variar de acordo com a distribuição. O próximo passo consiste em fazer as modificações desejadas, e isso torna-se simples graças ao comando chroot. “Ele permite transformar a pasta no diretório raiz do sistema, de modo que você possa instalar programas, instalar e remover pacotes e até mesmo abrir o KDE e sair alterando suas configurações” (MORIMOTO, 2006b). A adição e remoção de programas dentro do diretório montado pelo chroot pode ser feita através do aplicativo apt-get, mas isso não impede que o usuário descarte o comando chroot e modifique manualmente os arquivos de configuração e pacotes de software. Mais algumas configurações adicionais e o sistema de arquivos está pronto para ser compactado e transformado em uma imagem ISO. Por fim esta imagem pode ser gravada em um CD e poderá ser utilizada.

O Kurumin Linux é um exemplo onde o processo de remasterização é aplicado. Ele inclui um script que oferece alternativas ao usuário para remasterizar com a distribuição rodando no CD ou simplesmente alterar o conteúdo de uma ISO (MORIMOTO, 2006b).

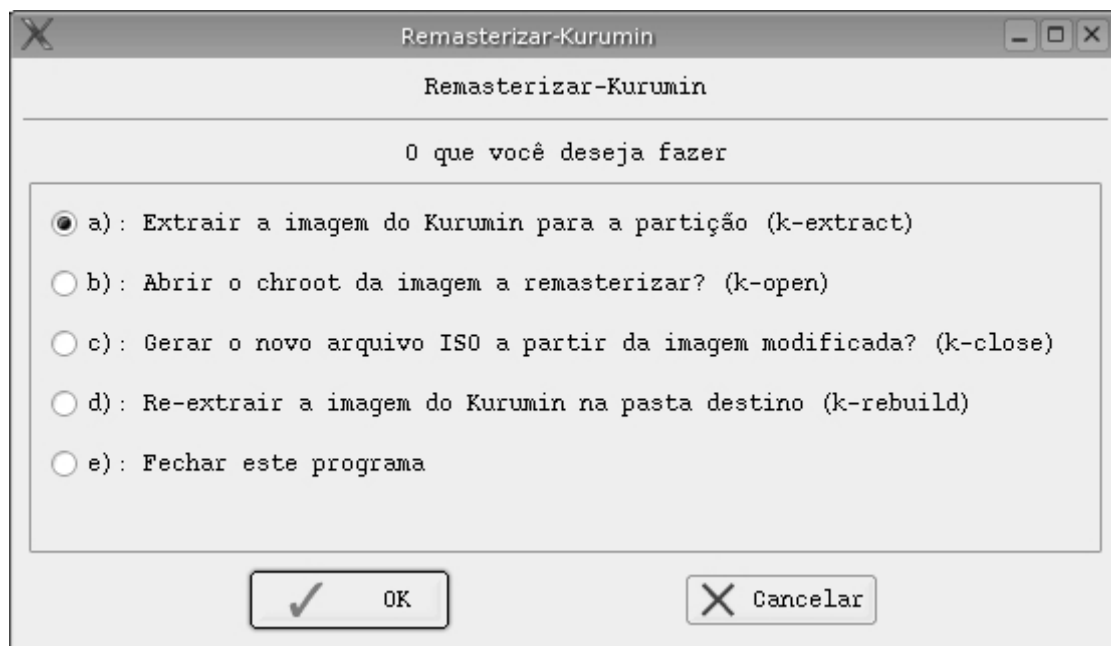


Figura 1: Primeira etapa do processo de remasterização do Kurumin Linux

Fonte: (MORIMOTO, 2006b)

Ainda como exemplo temos o KuruminDeveloper que é uma remasterização do Kurumin, o Ubuntu Games e Nubuntu que são remasterizações do Ubuntu.

3.5.4 MKDistro Maker

Conforme Dreamlinux (2006b), MKDistro Maker é uma ferramenta construtora de módulos e distribuições, desenvolvida por Nelson Gomes da Silveira, inspirada nas idéias advindas do Morphix, Knoppix e Debian, e seu objetivo é facilitar o desenvolvimento para os usuários que querem uma distro adaptadas às suas necessidades.

Conforme Dreamlinux (2006b) os componentes básicos do MKDistro Maker segue a seguinte divisão:

- ◆ MKDistro: é uma ferramenta automatizada e de simples manuseio que permite comandar todo o processo de criação e geração de distribuições, através de telas e menus auto-explicativos.
- ◆ Módulo-Base: é uma seleção básica de programas composto pelo Kernel e

módulos além de programas de detecção e configuração de hardware. Sua função é inicializar o sistema, detectar e configurar todo o hardware disponível no computador.

- ♦ Módulo-Principal-Básico: “é o módulo de trabalho que irá receber as instalações dos programas e as diversas configurações do sistema que estará sendo construído” (DREAMLINUX, 2006b). Consiste em um sistema de arquivos Debian completo, construído com a utilização do *debootstrap*²⁰, mais a inclusão de programas essenciais.
- ♦ Módulo principal da distro: “é o módulo final da distro, ou seja, é praticamente a nova distro, já incorporando todos os trabalhos executados no módulo-principal-básico” (DREAMLINUX, 2006b). É gerado após o encerramento dos trabalhos no Módulo-Principal-Básico, e será anexado ao módulo-base para a construção da imagem ISO.
- ♦ Imagem ISO: é módulo-base unido com o módulo principal da distro no formato de uma imagem ISO. Essa imagem é a nova distribuição, que posteriormente será gravada em um CD para ser utilizado.

O MKDistro Maker funciona como um *wizard*, dispondo de telas auto-explicativas oferecendo diversas alternativas ao usuário, isso o torna muito prático e amigável. Uma tela muito interessante é aquela que permite a seleção da família da distro, isso indica que o mesmo script pode ser utilizados para customizar diversas distribuições. Atualmente o MKDistro Maker oferece suporte às distribuições derivadas do Morphix, Knoppix e Mepis.

²⁰ Debootstrap: software utilizado pelo instalador de distribuições Debian e nele baseadas.

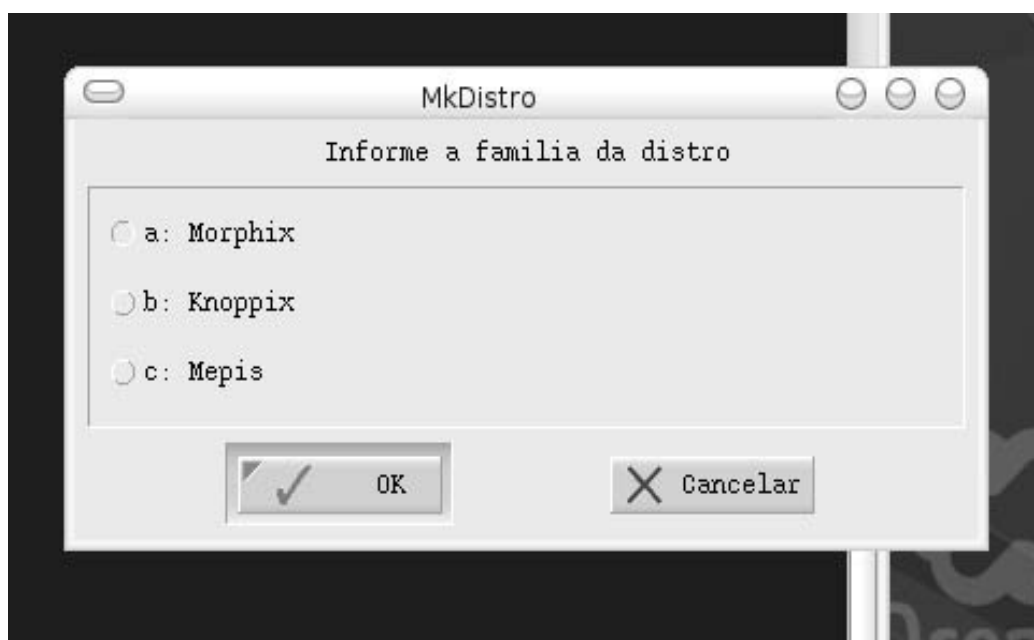


Figura 2: Quinto passo do MKDistro

Fonte: (DREAMLINUX, 2006c)

4. DESENVOLVIMENTO DO PROTÓTIPO

Para o desenvolvimento do protótipo foram utilizadas somente software livres:

- ◆ Distribuição Linux Ubuntu 7.04 - the Feisty Fawn;
- ◆ Distribuição Linux LFS Live-CD 6.2-5 Oficial;
- ◆ Construtor de Interfaces Glade 2.12.1;
- ◆ Linguagem de programação Python 2.5.1;
- ◆ Wrapper Python-gtk2 2.10.4;
- ◆ Máquina virtual innotek Virtualbox 1.4.0;
- ◆ Conjunto de Shell script lfslivecd-x86-6.2-pre2-BuildScripts.tar.bz2;
- ◆ Software de Gravação de CD nautilus-cd-burner

O hardware do microcomputador Desktop utilizado possui as seguintes especificações:

- ◆ Processador: AMD Sempron 1.68 Ghz;
- ◆ Memória RAM: 512 MB de frequência 400 Mhz;
- ◆ HD: 80 GB;
- ◆ Placa de vídeo: ATI Rage XL 8MB barramento AGP;

O particionamento do HD foi realizado em uma etapa anterior ao início do desenvolvimento, através do sistema operacional previamente instalado, ficando disposto da seguinte maneira:

Tabela 1: Esquema de partição do HD.

Dispositivo	Sistema de arquivos	Tamanho (GB)
hda1	ext3	35
hda2	linux-swap	1
hda3	ext3	35

A partição hda1 possui o sistema operacional Ubuntu 7.04 e a partição hda3 ficou disponível para a realização do processo.

Tento em vista o grande espaço de tempo consumido pelo desenvolvedor para construir uma distribuição do zero, surge a necessidade de ferramentas que automatizem e ao mesmo tempo garantam este processo.

Como exemplo de ferramentas que automatizam a construção utilizando o método LFS, podemos citar o *lfslivecd-BuildScripts* e o *JhaLFS*. Ambas as ferramentas assumem que o usuário tenha lido e esteja familiarizado com o processo de construção oferecido no livro do LFS e que o usuário tenha acesso à internet.

4.1 *lfslivecd-x86-6.2-pre2-BuildScripts*

É um pacote de Shell scripts mantido por Jeremy Huntwork e desenvolvido com o auxílio de Chris Lingard's, Jeremy Utley's e contribuições da comunidade. Originalmente, a idéia era apenas manter arquivos de configuração e uma lista de pacotes necessários em um repositório subversion²¹. O que realmente aconteceu foi que um sistema de scripts totalmente automatizado resultou, especificamente para o efeitos de construção do LFS Live-CD, e atualmente serve como uma espécie de framework²², onde os usuários constroem suas distribuições dentro do padrão oficial.

Seu funcionamento é controlado por um arquivo Makefile, ele contém as instruções para dirigir a construção, incluindo a ordem na qual os pacotes são construídos. Ele também contém variáveis que são utilizadas por toda a construção.

²¹ Subversion ou SVN: é um sistema de controle de versão.

²² Framework: é um conjunto de classes com objetivo de reutilização de *design*.

Dentro do pacote existe um diretório chamado *packages* onde ficam sub-diretórios para cada pacote que o Live-CD vai conter. Cada um desses diretórios mantém um Makefile específico com instruções específicas que necessita para compilar. Qualquer patch²³ necessário para o pacote que não está descrito nos livros do LFS e BLFS²⁴, estão incluídos nesses sub-diretório também.

O BLFS é um projeto que fornece instruções para adicionar recursos a uma distribuição construída com o LFS (BEEKMANS, 2007b).

Parte da construção é executado com um usuário sem privilégios chamado *lfs*, e o resto é feito após a entrada no *chroot*. A execução é de forma linear, os scripts Makefile dos pacotes é chamado para fazer download dos arquivos necessários, checar sua integridade com o *sha1sum*²⁵, descompactar os fontes no próprio diretório, se necessário aplicar os patches e por fim compilar e instalar o pacote, gerando um *log*²⁶ deste processo. Ao completar todo do processo, uma imagem no formato ISO²⁷ é gerada e poderá ser gravada em um CD/DVD ou emulada através de algum aplicativo específico.

Antes de iniciar o processo de construção é possível alterar algumas variáveis no arquivo Makefile principal, permitindo assim a configuração da arquitetura do CD que será gerado (x86, ppc ou sparc), do diretório padrão de compilação, do diretório onde serão gravados os pacotes de fontes adquiridos, entre outras. A dependência de pacotes não é implementada neste caso, ou seja, caso o Makefile tenha sido editado e pacotes foram retirados ou substituídos pelo usuário, suas dependências devem ser tratadas manualmente.

23 Patch: é um script que aplica diferenças textuais entre dois arquivos.

24 BLFS: acrônimo para *Beyond Linux From Scratch*.

25 Sha1sum: é um algoritmo de hash.

26 Log: é um arquivo que contém registros relevantes de um sistema operacional ou software.

27 Imagem ISO: um arquivo/imagem ISO contém um sistema de arquivos no formato ISO9660.

4.2 *JhaLFS-2.3.1*

O ALFS²⁸ ou *Automated Linux From Scratch* é um projeto que mantém um framework para construção automatizadas de distribuições LFS, destina-se a tornar o processo mais simples e eficiente, sem deixar de fornecer flexibilidade, permitindo ao usuário controle total para elaboração e gestão de seu sistema. O *JhaLFS* é a implementação oficial atual (BEEKMANS, 2007a).

Originalmente criado por Jeremy Huntwork, mas atualmente desenvolvido e mantido por Manuel Canales Esparcia e George Boudreau, o *JhaLFS* tornou-se um método muito prático de automatizar a construção de uma distribuição LFS. É um Shell script que faz uso do Subversion e Xsltproc²⁹ para fazer o download dos fontes XML do Linux From Scratch (versão em desenvolvimento) e, em seguida, extrair quaisquer comandos necessários, colocando-os em Shell scripts executáveis. Os pacotes necessários são adquiridos automaticamente, e finalmente o *JhaLFS* gera um Makefile que irá controlar a execução, permitindo a recuperação caso aconteça um erro durante a construção. Como o processo é feito diretamente sobre os fontes XML do livro LFS, não há perfis para serem editados.

Ao contrário do *lfs-livecd-BuildScripts*, ao final o *JhaLFS* não gera uma imagem ISO, ficará a cargo do usuário a compactação e construção da ISO, além da configuração a partir do capítulo 8 do livro LFS.

4.3 *Utilização do lfs-livecd-x86-6.2-pre2-BuildScripts*

O conjunto de Shell scripts *lfs-livecd-x86-6.2-pre2-BuildScripts* faz uso de uma versão estável do livro do LFS, permite seleção de pacotes a serem instalados (LFS e BLFS) e além disso gera uma imagem ISO ao final do processo. Estas características foram essenciais na escolha, além de apresentar resultados compatíveis aos esperados para a conclusão deste trabalho.

²⁸ Website: <http://www.linuxfromscratch.org/alfs>

²⁹ Xsltproc: é uma ferramenta de manipulação da linguagem XSL.

Através do sistema Ubuntu foi adquirido a imagem da distribuição oficial do LFS Live-CD 6.2-5 através do website oficial do Linux From Scratch e posteriormente esta foi gravada em uma mídia de CD. Também foi adquirido o pacote de `lfslivecd-x86-6.2-pre2-BuildScripts` e salvo na partição `hda3`.

Após cada boot do CD foi necessário montar a partição de trabalho e de swap, para isso a execução dos comandos abaixo no Shell do sistema foi necessária:

```
$ mkdir /mnt/lfs
$ mount /dev/hda3 /mnt/lfs
$ /sbin/swapon /dev/ hda2
```

O próximo passo foi descompactar o conteúdo do pacote de Shell scripts que já estava disponível no diretório `/mnt/lfs`.

```
$ cd /mnt/lfs
$ tar -xvf lfslivecd-x86-6.2-pre2-BuildScripts.tar.bz2
```

A criação do diretório `lfs-livecd` foi o resultado do comando, e sua estrutura ficou composta da seguinte maneira:

```
doc/*
etc/*
initramfs/*
isolinux/*
packages/*
root/*
scripts/*
uname/*
vars/*
LICENSE
Makefile
README
TODO
```

Avançando com o processo, foi necessário entrar no diretório `lfs-livecd` e, estando logado com o usuário `root`, aplicar o comando `make`. A partir daí o script prepara o ambiente e inicia o download e instalação dos pacotes.

Com o passar do tempo, uma série de pacotes tiveram seus *links* movidos

causando mensagens de erro e conseqüentemente abortagem do script. A cada falha ocorrida foi preciso efetuar correção no arquivo Makefile do pacote e reiniciar o processo de compilação. A lista dos pacotes com *links* quebrados que foram corrigidos está disponível no anexo II.

Pelo fato do processo de compilação ser muito demorado, diversas vezes ele foi interrompido para continuação no dia seguinte. A cada desligamento do microcomputador, o sistema anfitrião, por ser uma distribuição Live-CD, perdia suas configurações personalizadas e a tentativa de continuar a compilação do ponto abortado falhava. Para reverter esta situação, foi criado no diretório /mnt/lfs o arquivo lfs_inicio.sh com o seguinte conteúdo:

```
#!/bin/bash
export LFS=/mnt/lfs
groupadd lfs
useradd -s /bin/bash -g lfs -m -k /dev/null lfs
ln -nsf $LFS/tools /
```

Este Shell script precisou ser executado após cada inicialização da distribuição Live-CD anfitriã e montagem das partições de trabalho (hda2 e hda3). Sua função é configurar o ambiente com as mínimas características necessárias para a execução do lfslivecd-x86-6.2-pre2-BuildScripts.

Após um série de alterações e reinícios, o processo finalmente acaba, e o resultado de tudo isso é uma imagem ISO derivada do LFS Live-CD Oficial. A conclusão é informada com uma mensagem na tela: “The LiveCD, /mnt/lfs/lfs-livecd/lfslivecd-x86-6.2-2.iso, is ready!”. O conteúdo da imagem ISO ficou disposta da seguinte maneira:

```
boot/*
root.sqfs
```

No diretório boot fica o gerenciador de boot ISOLINUX, responsável por carregar o módulo *cloop* do Kernel, interpretar as opções informadas na tela inicial e montar o sistema de arquivos compactado no arquivo root.sqfs.

4.4 Processo de Remasterização

Acrescentar pacotes diretamente no momento da compilação do `lfslivecd-x86-6.2-pre2-BuildScripts`, requer um conhecimento amplo em de Linux e muita habilidade na construção de Shell scripts. Por isso, tentando evitar futuras falhas a decisão tomada foi efetuar a adição de novos pacotes através do método de remasterização.

Antes de iniciar o processo de instalação/remoção é necessário montar o sistema de arquivos do tipo *squashfs* contido no `root.sqfs`, o inconveniente é que este não pode ser montado com permissão de escrita, sendo assim não é possível realizar a remasterização. Este mesmo problema foi enfrentado por outros usuários e a versão 6.2-5 do Live-CD oficial LFS passou a ter um sistema de arquivos diferente, no formato `ext2`. A versão 6.2-5 segue o mesmo livro LFS e traz o sistema de arquivos compactado em `root.ext2`, ou seja, a única diferença entre as versões é o tipo do sistema de arquivos, e ainda, a mais recente permite a montagem com permissão de escrita. Devido a estes fatores a decisão tomada foi substituir a imagem ISO gerada pelo `lfslivecd-x86-6.2-pre2-BuildScripts` pela imagem do LFS Live-CD 6.2-5 oficial e retomar a remasterização a partir deste ponto. O arquivo `root.sqfs` foi abandonado e o `root.ext2` foi extraído da imagem ISO adquirida dos repositórios do Linux From Scratch. Um usuário mais experiente poderá aproveitar o sistema de arquivos compactado em *squashfs* copiando-o para uma partição física e efetuando o processo de remasterização neste.

O processo de remasterização foi efetuado através do sistema anfitrião Ubuntu 7.04 instalado na partição `hda1` do mesmo microcomputador.

Para o processo de remasterização os requisitos do sistema anfitrião são:

- ◆ Pacotes *cdrtools-2.01* e *zisofs-tools-1.0.6* ou em versões recentes;
- ◆ Permissões de super-usuário;

- ♦ Possuir suporte ao loop device³⁰;
- ♦ CD-ROM e suporte a sistema de arquivos ISO9660;
- ♦ Aproximadamente 3GB de espaço livre em disco;

Os pacotes *cdrtools* e *zisofs-tools* possuem utilitários para criação, edição e gravação de CD, e será útil no momento de recriarmos e compactarmos a imagem ISO.

O pacote *coreutils* é um componente nativo do sistema anfitrião e contém utilitários que permitem ver e ajustar as características básicas do sistema, como por exemplo o *chroot* que permite transformar um diretório qualquer em um diretório raiz, podendo este preceder parâmetros que definem variáveis no novo ambiente. Ao executar o *chroot*, o usuário será direcionado para um Shell do sistema e os comandos neste executados afetarão apenas o que estiver sob o ambiente do *chroot*. Alguns procedimentos, como a configuração instalação/remoção de pacotes devem ser executadas nesse ambiente.

Todos os requisitos do sistema anfitrião foram nativamente atendidos. Em adicional, foi acrescentado o software innotek VirtualBox, capaz de emular nossa distribuição Live-CD através da imagem ISO, anulando a necessidade de queimarmos uma mídia de CD/DVD para cada realização de testes.

Para executar a instalação do innotek VirtualBox no sistema Ubuntu, foi executado o seguinte comando:

```
$ sudo apt-get install virtualbox
```

Os procedimentos para realização do processo de remasterização consistiram basicamente em:

- ♦ Definir uma estrutura básica de diretórios;
- ♦ Montagem dos sistemas de arquivos;

³⁰ Loop device ou loopback device: é um dispositivo do sistema operacional que permite que um arquivo de imagem seja montado como se fosse uma partição normal do HD.

- ♦ Acesso a novo ambiente usando chroot;
- ♦ Instalação/remoção de pacotes, configurações, ajustes;
- ♦ Saída do ambiente chroot;
- ♦ Desmontagem dos sistema de arquivos;
- ♦ Recompessão do sistema de arquivos em root.ext2;
- ♦ Recriação da imagem ISO

Todos os procedimentos que seguem neste capítulo foram realizados logado com o usuário root. O diretório escolhido foi /root, e a estrutura de diretórios criada ficou disposta da seguinte forma:

- ♦ /root/build: diretório de acesso comum entre ambiente do sistema anfitrião e ambiente chroot;
- ♦ /root/iso: diretório onde fica a estrutura de diretórios/arquivos que ficará contido na imagem ISO que será recriada após a customização.
- ♦ /root/root: será o ponto de montagem do sistema de arquivos contido no arquivo root.ext2;
- ♦ /root/scripts: contém os Shell scripts para automatizar a montagem, desmontagem e geração da ISO;
- ♦ /root/backups: diretório para realização periódica de backups do arquivo root.ext2;

Para iniciar o processo foi necessário adquirir, montar a imagem ISO da versão 6.2-5 do LFS³¹ e adquirir seu conteúdo:

```
$ mkdir /mnt/tmp
$ mount -t ISO9660 -o loop /root/lfslivecd-x86-6.2-5.iso /mnt/tmp
$ mkzftree -u -F /mnt/tmp/root.ext2 /root/root.ext2
$ cp -Rp /mnt/tmp/boot /root/iso
$ umount /mnt/tmp
```

31 Todos os arquivos utilizados que foram obtidos da internet estão listados no anexo I.

```
$ rm -r /mnt/tmp
```

O próximo passo foi a criação dos scripts `mount.sh`, `umount.sh` e `iso.sh` para automatizar respectivamente a montagem e entrada no `chroot`, para desmontagem e para geração da imagem ISO. Estes ficaram alojados no diretório `/root/scripts` e com o passar do tempo sofreram melhoramentos, a versão final de cada script está listada no que segue.

- ♦ Script *mount.sh*

```
#!/bin/bash

cd /root
mount -o loop root.ext2 root
mkdir root/buil
mount --bind build/ root/build/
mount -t proc proc root/proc/
mount -t sysfs sysfs root/sys/
mount -t devpts devpts root/dev/pts
mount -t tmpfs tmpfs root/dev/shm/
chroot /root/root /usr/bin/env -i \
HOME=/root TERM="$TERM" PS1='\u:\w\$ ' \
PATH=/bin:/usr/bin:/sbin:/usr/sbin \
/bin/bash --login
```

- ♦ Script *umount.sh*

```
#!/bin/bash

umount /root/root/dev/shm
umount /root/root/dev/pts
umount /root/root/sys/
umount /root/root/proc
umount /root/root/build/
rmdir /root/root/build/
rm /root/root/root/.bash_history
dd if=/dev/zero of=/root/root/zeroes
rm /root/root/zeroes
rm /root/root/root/.viminfo
umount /root/root
sync
cd /root/scripts
```

Nota: Uma mensagem informando que o disco está cheio é mostrada na tela durante a execução deste script, ela deve ser ignorada.

♦ Script iso.sh

```
#!/bin/bash

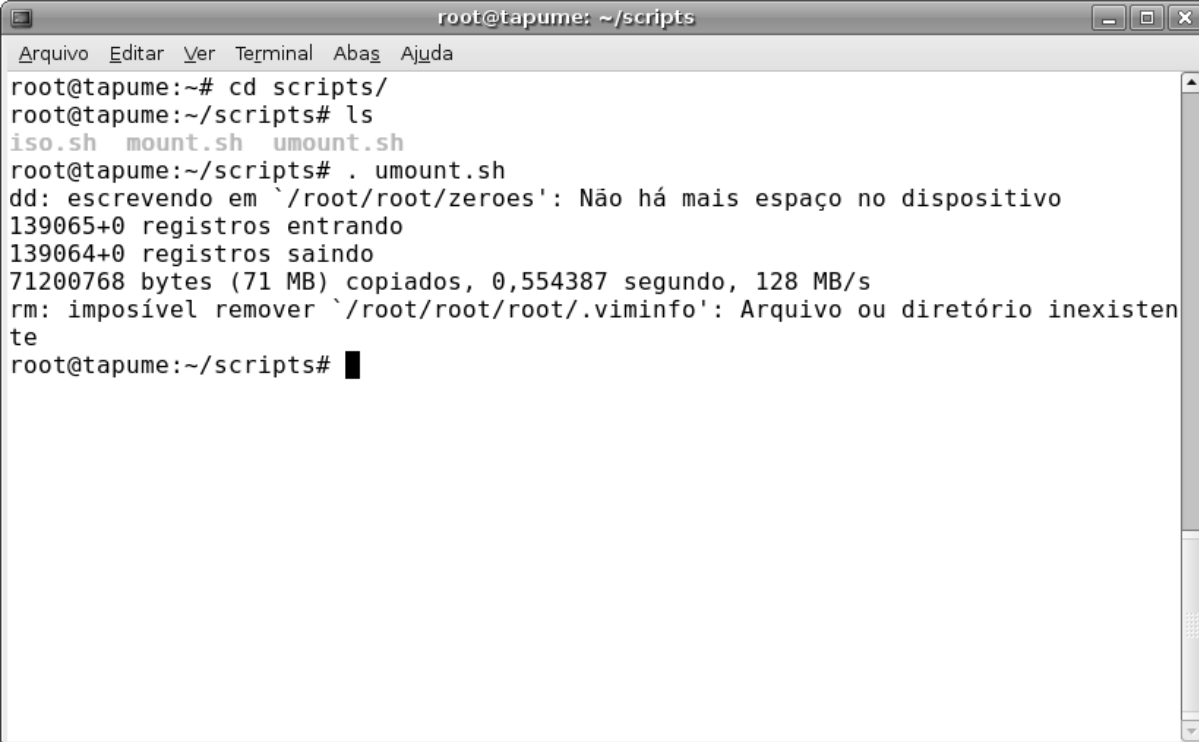
cd /root
rm /root/lfslivecd-x86-6.2-UNOLIVE.iso
rm /root/iso/root.ext2
echo "----- COMPACTANDO ROOT.EXT2 -----"
mkzftree -F root.ext2 iso/root.ext2
cd /root/iso
mkisofs -z -R -l --allow-leading-dots -D -o \
/root/lfslivecd-x86-6.2-UNOLIVE.iso -b \
boot/isolinux/isolinux.bin -c boot/boot.cat \
-no-emul-boot -boot-load-size 4 -boot-info-table \
-V "lfslivecd-x86-6.2-UNOLIVE" ./
chown clemente. /root/lfslivecd-x86-6.2-UNOLIVE.iso
cd /root
echo
du -sch lfslivecd-x86-6.2-UNOLIVE.iso
```

Todo o processo de instalação/remoção/configuração dos pacotes foram efetuados dentro do ambiente *chroot*. Para entrar no ambiente *chroot* o script *mount.sh* foi utilizado.

```
root@tapume: ~/scripts
Arquivo Editar Ver Terminal Abas Ajuda
root@tapume:~# cd scripts/
root@tapume:~/scripts# ls
iso.sh  mount.sh  umount.sh
root@tapume:~/scripts# . mount.sh
tapume:root | Qua 07 Nov 2007 14:39:18 GMT | /
# ls
bin    build  etc    lib          media  opt    root  share  sys  usr
boot  dev    home  lost+found  mnt    proc  sbin  srv    tmp  var
tapume:root | Qua 07 Nov 2007 14:39:20 GMT | /
# █
```

Ilustração 1: Entrada automatizada no ambiente *chroot* usando o script *mount.sh*.

Para cada saída do chroot o comando *exit* foi utilizado e, logo após a execução do script *umount.sh* garantiu a desmontagem dos sistemas de arquivos.



```
root@tapume: ~/scripts
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
root@tapume:~# cd scripts/
root@tapume:~/scripts# ls
iso.sh  mount.sh  umount.sh
root@tapume:~/scripts# . umount.sh
dd: escrevendo em `/root/root/zeros': Não há mais espaço no dispositivo
139065+0 registros entrando
139064+0 registros saindo
71200768 bytes (71 MB) copiados, 0,554387 segundo, 128 MB/s
rm: impossível remover `/root/root/root/.viminfo': Arquivo ou diretório inexisten
te
root@tapume:~/scripts#
```

Ilustração 2: Desmontagem automatizada do sistema de arquivos utilizando o script umount.sh.

O primeiro passo realizado após a primeira entrada no ambiente chroot foi a criação de uma nova imagem do *initramfs*. Esta precisou ser recompilada de acordo com o nome da nova distribuição. Através do navegador de internet do sistema anfitrião, o pacote no formato zip foi adquirido e salvo no diretório */root/buid*, ficando desta forma disponível também para o Shell do ambiente chroot. No ambiente chroot foi preciso descompactar o pacote, remover a linha “include $\$(ROOT)/scripts/functions$ ” do arquivo Makefile e compilar o pacote. Este pacote é também parte do conjunto de Shell scripts *lfslivecd-x86-6.2-pre2-BuildScripts* e para ser executado fora deste contexto foi necessário adaptá-lo, por isso a remoção de tal linha do arquivo Makefile.

```
$ make compile-stage2 VERSION="x86-6.2-UNOLIVE"
```

A saída do comando acima gerou um arquivo chamado `initramfs_data_cpio.gz` que foi utilizado mais adiante, no momento da geração da ISO.

4.4.1 Alteração dos scripts de inicialização e arquivos de configuração

O SysVinit ou simplesmente `init` é um recurso de inicialização baseado no conceito de *run-levels* (níveis de execução), podendo ser completamente diferente em um sistema com relação a outro. As distribuições baseadas no LFS tem sua própria maneira de fazer coisas, mas respeita os padrões geralmente aceitos, possuindo 6 níveis. Cada nível corresponde às tarefas que o sistema supostamente irá executar durante a inicialização.

Existem subdiretórios em `/etc/rc.d` com nome `rc1.d`, `rc2.d`, `rc3.d`, `rc4.d`, `rc5.d`, `rc6.d` e `rcsysinit.d`, onde ficam os links simbólicos que apontam para os scripts.

O nível de execução padrão no Live-CD é o 3, desta forma, durante o boot, o sistema irá executar os scripts apontados pelos links simbólicos existentes no diretório `rcsysinit.d` (diretório comum para todos os níveis) e `rc3.d` (diretório exclusivo do nível 3)³².

Originalmente o sistema executava os scripts *clkconf* e *langconf* durante o boot, estes apresentavam ao usuário uma caixa de seleção para escolha do fuso horário e configurações de teclado respectivamente. Assumindo que na maioria das vezes a escolha, tanto do idioma quanto do teclado, seria o Português Brasileiro, estas configurações foram feitas manualmente nos arquivos de configuração que por eles eram afetados.

A remoção desses scripts da inicialização foi o segundo passo realizado após a entrada no ambiente `chroot`, para isso o seguinte comando foi necessário:

³² O capítulo 7.3 do Livro LFS 6.2 traz informações de forma detalhada sobre os scripts de inicialização


```
$ rm /etc/rc.d/rcsysinit.d/S59clkconf
$ rm /etc/rc.d/rcsysinit.d/S69langconf
```

Os seguintes arquivos foram alterados manualmente e as respectivas linhas de configuração foram a eles acrescentas:

- ♦ /etc/environment

```
TZ=GMT
```

- ♦ /etc/sysconfig/rc

```
TZ=GMT ; export TZ
```

- ♦ /etc/profile

```
TZ=GMT ; export TZ
LANG="pt_BR" ; export LANG
```

- ♦ /etc/sysconfig/clock

```
UTC=0
```

- ♦ /etc/sysconfig/console

```
KEYMAP="br-abnt2"
FONT="lat1-16 -m 8859-1"
```

No arquivo /etc/X11/xorg.conf, uma substituição foi realizada com uso do utilitário sed:

```
$ sed -i "s/\"us\"/\"br-abnt2\"/" /etc/X11/xorg.conf
```

4.4.2 Adição de pacotes

O processo de adição de pacotes consistiu, de maneira geral, em obtê-los a partir dos websites oficiais, obter os patches de correção quando necessários a partir dos repositórios do LFS, compilar os fontes do aplicativos, instalar e efetuar as configurações pós-instalação. Alguns deles serviram apenas como novos recursos do Live-CD e outros, como o Python, o Libglade, o PyGTK serviram para suprir

dependências e permitir a execução do “Painel do Live-CD”. Abaixo segue a listagem dos software que foram acrescentados ao Live-CD e a maneira que foram instalados. Uma relação dos presentes no Live-CD está disponível no arquivo README, alocado no diretório /root e no URL <http://www.unochapeco.edu.br/~clemente/livecd>.

- ◆ Firefox-1.5.0.9

O primeiro passo realizado ao que se refere a instalação/remoção de pacotes, foi a substituição do navegador de internet Seamonkey pelo Mozilla Firefox. O Seamonkey é um projeto mantido pela comunidade com intuito de contribuir no desenvolvimento da suíte Mozilla Firefox (navegador de internet), Mozilla Thunderbird (cliente de e-mail) entre outros aplicativos, mas apresentou muitas instabilidades no Live-CD, principalmente no que se refere a acesso à paginas com execução de requisições AJAX³³. Por este motivo foi substituído pela versão 1.5.0.9 do Mozilla Firefox. Esta versão específica foi a escolhida por estar contida no livro do LFS e ser estável na versão 6.2.

A desinstalação do aplicativo Seamonkey foi realizada com a execução dos seguintes comandos:

```
$ rm -f /usr/{bin,lib/pkgconfig}/seamonkey*
$ rm -f /usr/share/aclocal/nspr.m4
$ rm -rf /usr/{lib,include,share/idl}/seamonkey-1.0.5
```

Para instalação do Firefox-1.5.0.9, foi preciso adquirir os arquivos fontes e o patch de correção necessário devido ao Pango³⁴ estar reinderizando as fontes. Os URLs destes, bem como de todos os outros software estão listados no anexo I.

O arquivo de configuração adicional necessário .mozconfig foi criado no diretório resultado da descompactação do arquivos de fonte (/build/mozilla) com o seguinte conteúdo:

³³ AJAX: acrônimo para *Asynchronous Javascript And XML*: é o uso sistemático de tecnologias providas por navegadores, como Javascript e XML, para tornar páginas mais interativas com o usuário.

³⁴ Pango: é uma biblioteca usada para renderizar texto.

```
. $topsrcdir/browser/config/mozconfig
mk_add_options MOZ_OBJDIR=@TOPSRCDIR@/../../firefox-build
ac_add_options --prefix=/usr
ac_add_options --with-system-zlib
ac_add_options --with-system-png
ac_add_options --with-system-jpeg
ac_add_options --enable-system-cairo
ac_add_options --enable-official-branding
ac_add_options --enable-canvas
ac_add_options --enable-svg
ac_add_options --enable-strip
ac_add_options --disable-tests
ac_add_options --disable-installer
ac_add_options --disable-accessibility
ac_add_options --enable-pango
```

Os comandos que seguem foram necessários para a aplicação do patch, compilação e instalação do Firefox:

```
$ cd /build/mozilla
$ patch -Np1 -i ../../firefox-1.5.0.9-pangoxft-1.patch
$ sed -i "s/^      enum$/& xptinfo_enum_1/" \
    xpcor/reflect/xptinfo/public/xptinfo.h
$ make -f client.mk build
$ make -f client.mk install
$ install -v -m755 -d \
    /usr/lib/firefox-1.5.0.9/chrome/icons/default
$ ln -v -s ../../../../icons/default.xpm \
    /usr/lib/firefox-1.5.0.9/chrome/icons/default
$ chown -v -R root:root \
    /usr/lib/firefox-1.5.0.9/extensions/inspector@mozilla.org/*
$ install -v -m755 -d /usr/include/firefox-1.5.0.9/nss
$ cp -v -Lf ../../firefox-build/dist/{private,public}/nss/*.h \
    /usr/include/firefox-1.5.0.9/nss
```

Uma descrição detalhada dos comandos e do arquivo de configuração .mozconfig pode ser encontrada no capítulo X do BLFS³⁵.

A aceitação do navegador Mozilla Firefox aumenta cada vez mais entre os internautas por ser gratuito, estável e flexível. Na nova distribuição, este foi adequando para atender as necessidades dos alunos, ao fazerem ou não uso em laboratórios desta instituição. Os complementos *Firebug* e *SwitchProxy Tool* foram instalados e configurados além da instalação do Plugin do Flash, personalização da barra de favoritos e alteração da página inicial.

³⁵ <http://www.linuxfromscratch.org/blfs/view/stable/xsoft/firefox.html>

O *Firebug* é uma ferramenta que permite que o usuário edite, depure e controle códigos CSS, HTML e Javascript no momento que acessa qualquer página da internet e o *SwitchProxy Tool* é uma ferramenta que permite o gerenciamento e alternância entre múltiplas configurações de proxy.

A instalação de complementos no Mozilla Firefox consiste basicamente em adquirir os pacotes do website oficial, identificar o ID da extensão, extrair o conteúdo do pacote em um diretório com o nome do ID do pacote e mover este diretório para o diretório `extensions` do Firefox. Assim sendo foram adquiridas as extensões *Firebug* e *SwitchProxy Tool* e o ID foi identificado entre as *tags*³⁶ `<em:id>` e `</em:id>` do primeiro bloco `<Description>` do arquivo `install.rdf`. Posteriormente ambos diretórios foram movidos para o diretório `/usr/lib/firefox-1.5.0.9/extensions/`.

Para configuração do complemento *SwitchProxy Tool* foi preciso a criar o `/usr/lib/firefox-1.5.0.9/defaults/profile/localstore.rdf`, um arquivo XML contendo as configurações de proxy da Unochapecó.

No arquivo `/usr/lib/firefox-1.5.0.9/defaults/profile/bookmarks.html` foram acrescentadas algumas linhas, que equivalem a exibição de atalhos na barra de favoritos para acesso aos websites `http://www.google.com.br` e `http://br.php.net`. As linhas acrescentadas ao arquivo foram:

```
<DT><A HREF="http://www.google.com.br/" ADD_DATE="1193219457"
LAST_CHARSET="UTF-8" ID="rdf:#$DhxKA3">Google</A>
<DT><A HREF="http://br.php.net/" ADD_DATE="1193219491"
LAST_CHARSET="UTF-8" ID="rdf:#$EhxKA3">PHP.net</A>
```

Para a instalação do Plugin do Flash foi preciso obter e extrair o pacote de `install_flash_player_9_linux.tar.gz`, ato seguinte os arquivos `flashplayer.xpt` e `libflashplayer.so` foram movidos para o diretório `/usr/lib/firefox-1.5.0.9/plugins/`.

Algumas configurações, tais como a mudança do URL da página inicial para `http://www.unochapeco.edu.br`, foram realizadas no arquivo `/usr/lib/firefox-`

³⁶ Tags: em um código-fonte de programa, delimitadores indicam o início e fim de um bloco de dados.

1.5.0.9/defaults/profile/prefs.js.

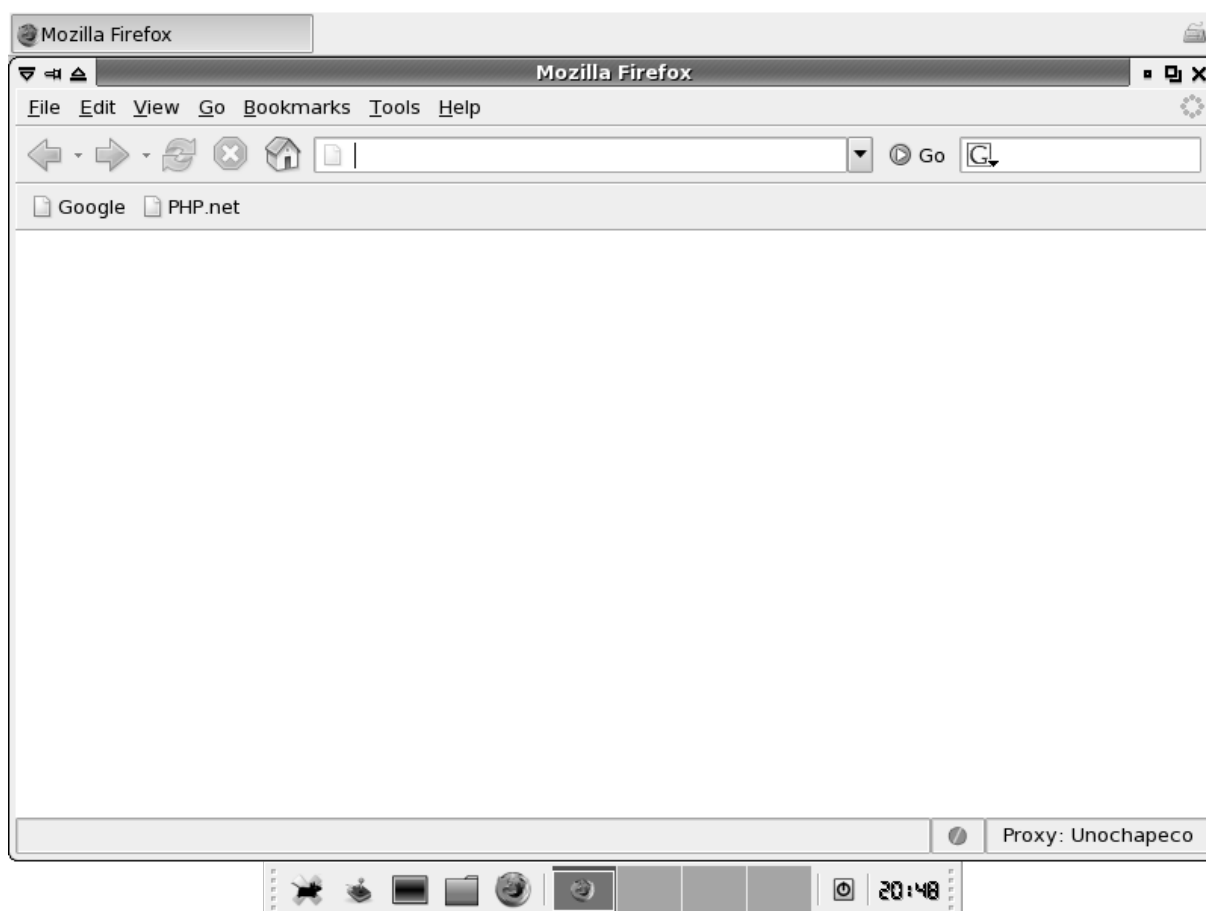


Ilustração 3: Navegador Mozilla Firefox sendo executado no Live-CD.

- ◆ Apache-2.2.2

O Apache HTTP é um projeto da *Apache Software Foundation*, e contém um servidor HTTP de código aberto. Conforme Apache (2007), o projeto é mantido por um grupo de voluntários de várias partes do mundo e é o mais utilizado servidor HTTP livre (APACHE, 2007a).

A instalação do servidor Apache no Live-CD foi realizada com o auxílio o livro BLFS e consistiu no download do pacote de código-fonte `httpd-2.2.2.tar.bz2` e patch de correção `httpd-2.2.2-config-1.patch`, adição do grupo e usuário `apache` no sistema, aplicação do patch de correção, compilação e instalação.

```
$ groupadd -g 25 apache
```

```
$ useradd -c "Apache Server" -d /dev/null -g apache \
-s /bin/false -u 25 apache
$ cd /build/httpd/httpd-2.2.2
$ patch -Np1 -i ../httpd-2.2.2-config-1.patch
$ ./configure --enable-layout=FHS --enable-mods-shared=all
$ make
$ make install
$ chown -v root:root /usr/lib/apache/httpd.exp \
/usr/sbin/{apxs,apachectl,dbmmanage,envvars{,-std}} \
/usr/share/man/man1/{dbmmanage,ht{dbm,digest,passwd}}.1 \
/usr/share/man/man8/{ab,apachectl,apxs,htcacheclean,httpd}.8 \
/usr/share/man/man8/{logresolve,rotatelog,suexec}.8
$ chown -v -R apache:apache /srv/www
```

Após a instalação foi preciso ajustar o arquivo `/etc/httpd.conf`, os seguintes comandos foram aplicados para a realização do ajuste:

```
$ sed -i -e "s/User daemon/User apache/" \
-e "s/Group daemon/Group apache/" \
/etc/apache/httpd.conf
$ sed -i "s/^LoadModule isapi_module/# &/" \
/etc/apache/httpd.conf
```

O Apache é executado como um serviço, e foi posto para iniciar automaticamente durante o boot do sistema. Esse procedimento exigiu a criação do script em `/etc/rc.d/init.d/apache` e de links simbólicos nos referidos diretórios de níveis do init. Isso tudo pôde ser automatizado com o uso do pacote `blfs-bootscripts-20060910`. Após a descompactação, o seguinte comando foi necessário:

```
$ cd /bulid/blfs-bootscripts-20060910 && make install-apache
```

O diretório `/srv/www/htdocs` é a raiz do website criado localmente que pode ser acessado através do URL `http://localhost`.

♦ Python-2.5.1

Python é uma linguagem de programação interpretada de alto nível e apesar de sua sintaxe simples, oferece os muitos recursos disponíveis também em linguagens mais complicadas como Java e C++. Foi desenvolvida por Guido van Rossum e recebe este nome em homenagem ao grupo humorístico inglês Monty Python, adorado por *geeks* de todo o mundo.

O interpretador da linguagem Python será necessário para executar a aplicação “Painel do Live-CD”, sendo assim sua instalação tornou-se pré-requisito para a conclusão do trabalho. O primeiro passo foi adquirir o pacote através do website oficial (<http://www.python.org>) e posterior a descompactação do pacote, foi preciso aplicar os comandos abaixo para concluir a instalação:

```
$ cd /build/Python-2.5.1
$ ./configure
$ make
$ make install
```

♦ MySQL-5.0.21

É um gerenciador de banco de dados muito popular conhecido pela facilidade de integração com a linguagem PHP.

Assim como no Apache, é necessário a criação de um grupo e um usuário do sistema específico para seu funcionamento:

```
$ groupadd -g 40 mysql
$ useradd -c "MySQL Server" -d /dev/null -g mysql -s \
/bin/false -u 40 mysql
```

A compilação e instalação do MySQL foi realizada com a aplicação dos seguintes comandos:

```
$ cd /build/mysql-5.0.21
$ C_EXTRA_FLAGS=-fno-strict-aliasing \
./configure --prefix=/usr \
--sysconfdir=/etc \
--libexecdir=/usr/sbin \
--localstatedir=/srv/mysql \
--enable-thread-safe-client \
--enable-asm \
--enable-local-infile \
--with-unix-socket-path=/var/run/mysql/mysql.sock \
--without-debug \
--without-bench \
--without-readline \
--with-berkeley-db \
--with-extra-charsets=all
$ make testdir=/tmp/mysql
$ make testdir=/tmp/mysql install
```

```
$ rm -rf /tmp/mysql
$ cd /usr/lib
$ ln -v -sf mysql/libmysqlclient{,_r}.so* .
```

Para finalizar, foi preciso transferir um arquivo de configuração e criar uma base de dados padrão entre outras configurações:

```
$ install -v -m644 /usr/share/mysql/my-medium.cnf /etc/my.cnf
$ mysql_install_db --user=mysql
$ chgrp -v mysql /srv/mysql{,/test,/mysql}
$ install -v -m755 -o mysql -g mysql -d /var/run/mysql
```

A senha do usuário padrão (root) para acesso ao banco de dados não foi criada, sendo desnecessário informá-la para o acesso.

Igualmente ao Apache, o MySQL é executado com um serviço e a instalação dos scripts de inicialização foi automatizada com o uso do pacote blfs-bootscripts-20060910:

```
$ cd /bulid/blfs-bootscripts-20060910 && make install-mysql
```

♦ PHP-5.2.4

O PHP³⁷ é uma linguagem de programação de livre licença para desenvolvimento Web, podendo ser mesclada ao código HTML³⁸. O principal objetivo do PHP é permitir a geração de páginas dinamicamente e de forma rápida, mas esta linguagem permite muito mais que isso (ACHOUR, 2007).

Após adquirir o pacote através do website oficial (<http://br.php.net>) e extrair seu conteúdo, foi necessário os seguintes comandos para compilar, instalar e configurar o pacote:

```
$ cd /buid/php-5.2.4
$ ./configure --prefix=/usr \
  --sysconfdir=/etc \
  --with-apxs2=/usr/sbin/apxs \
  --enable-force-cgi-redirect \
  --enable-discard-path \
  --with-config-file-path=/etc \
  --with-zlib \
```

37 PHP: acrônimo para *Hypertext Preprocessor*.

38 HTML: acrônimo para *HyperText Markup Language*: é uma linguagem de marcação.


```

--enable-bcmath \
--with-bz2 \
--enable-calendar \
--enable-dba \
--enable-exif \
--enable-ftp \
--with-gettext \
--enable-mbstring \
--with-ncurses \
--with-readline \
--enable-magic-quotes \
--enable-safe-mode \
--with-mysql \
--with-mysql-sock=/var/run/mysql
$ make
$ make install
$ install -v -m644 php.ini-recommended /etc/php.ini

```

Por padrão a variável *short_open_tag* do arquivo de configuração */etc/php.ini* vem setada com o valor *Off*, dessa forma o início e fim de um bloco de código PHP está limitado entre as respectivas *tags* *<?php* e *?>*. O valor da variável foi manualmente substituído por *On*, assim um bloco de código PHP poderá iniciar também com a *tag* *<?*. Essa alteração também pode ser feita com auxílio do aplicativo *sed*:

```
$ sed -i 's/short_open_tag = Off/short_open_tag = On' /etc/php.ini
```

Para o PHP ser executado no ambiente web, ou seja, o servidor Apache reconhecer o PHP, foi necessário adicionar algumas linhas no arquivo de configuração do Apache (*/etc/apache/httpd.conf*). Assim sendo foi acrescentado a string *AddType application/x-httpd-php .php* logo abaixo da linha *AddType application/x-gzip .gz .tgz* já existente. Dentro do mesmo arquivo foi concatenado a string *index.php* na variável *DirectoryIndex* para que o arquivo *index.php* também seja índice do diretório em que reside (APACHE, 2007b).

♦ phpMyAdmin 2.11.2

O phpMyAdmin é um sistema desenvolvido em PHP para administração do banco de dados MySQL. Ele funciona através de um navegador de internet e desta

forma é possível gerenciar tabelas e informações em um banco de dados remoto através da internet. A instalação do phpMyAdmin consistiu em obter o pacote do website oficial (<http://www.phpmyadmin.net>), descompactar o mesmo e mover seu conteúdo para o diretório `/srv/www/htdocs/phpmyadmin`. O acesso ao phpMyAdmin pode ser feito diretamente no navegador de Internet Mozilla Firefox, através do URL <http://localhost/phpmyadmin>.

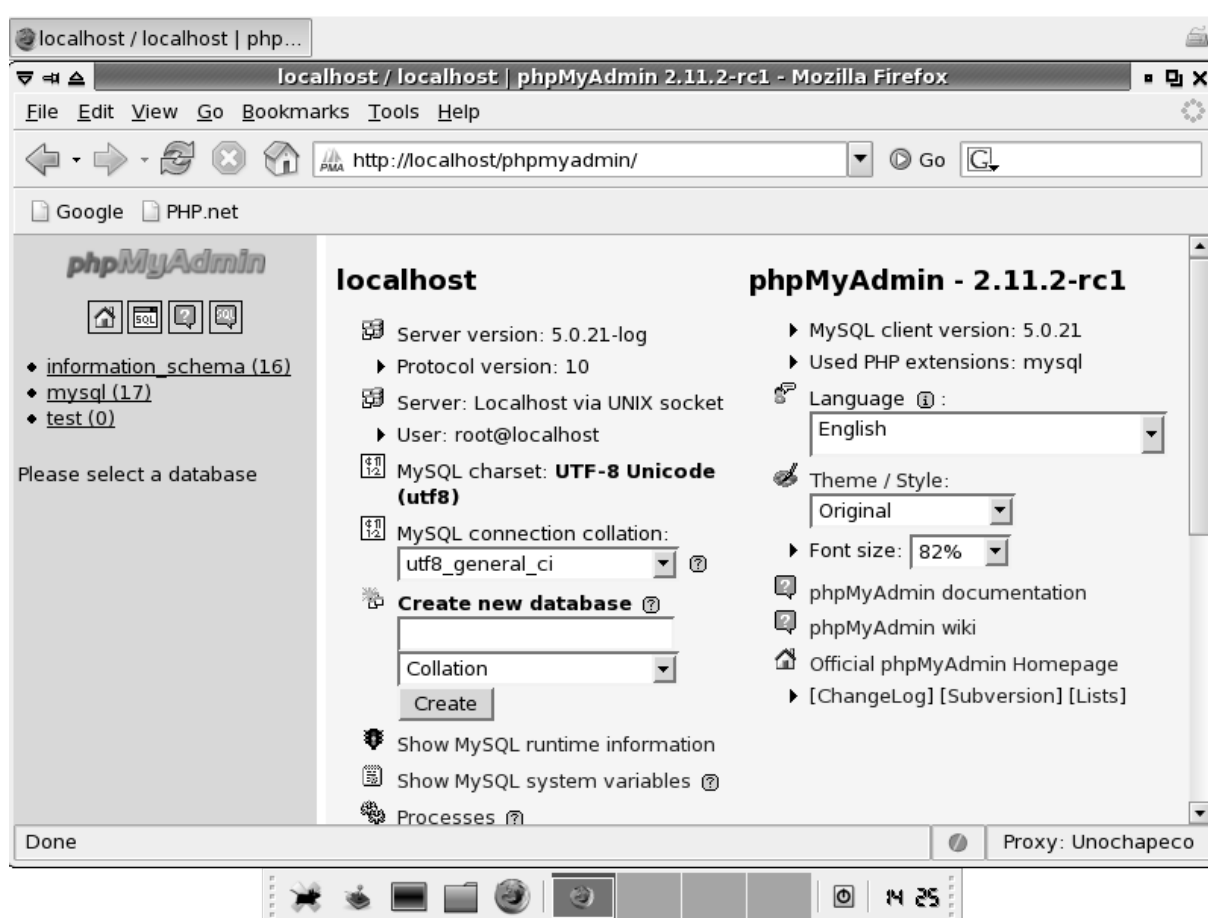


Ilustração 4: phpMyAdmin sendo executado no Live-CD.

- ◆ Frameworks PHP

Um recurso do Live-CD é a possibilidade de desenvolvimento com Frameworks PHP, para isso foram inclusos os Frameworks CakePHP (<http://www.cakephp.org/>), SlimPHP (<http://slimphp.sourceforge.net/>), Symfony PHP (<http://www.symfony-project.com/>) e Zend Framework (<http://framework.zend.com/>).

A instalação destes consistiu basicamente em obtê-los através de seus websites oficiais e extraí-los no diretório raiz do website local (/srv/www/htdocs).

- ♦ Dia-0.96.1

O software para criação de diagramas incluso no Live-CD é o Dia, em sua versão 0.96.1 traz uma variedade muito grande de objetos gráficos categoricamente agrupados, incluindo os de UML³⁹.

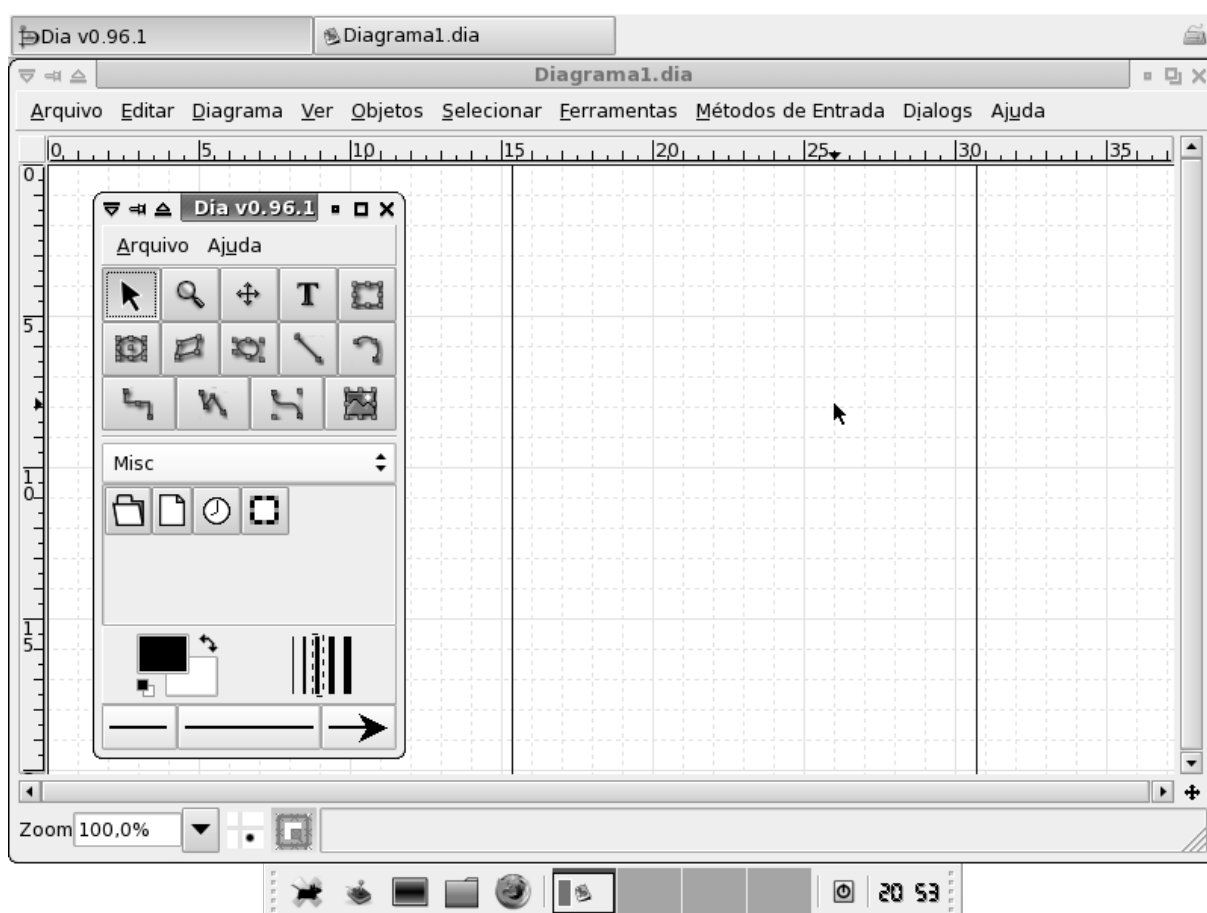


Ilustração 5: Dia Diagrama sendo executado no Live-CD.

O download do pacote das fontes do website oficial foi necessário e para sua compilação e instalação, os comandos abaixo foram requeridos:

```
$ cd /build/dia-0.96.1
```

³⁹ UML: acrônimo para *Unified Modeling Language*, é uma linguagem de modelagem não proprietária de terceira geração.

```
$ ./configure --prefix=/usr
$ make
$ make install
```

♦ Geany-0.11

O Geany é uma IDE⁴⁰ de desenvolvimento simples, rápida e flexível. Suporta highlighting⁴¹ para mais de 20 linguagem, tais como C, PHP, Shell Script, Java, Pearl, Ruby, Assembler, etc. além de permitir a alteração da forma de codificação dos arquivos através do menu principal.

O Geany possui recursos para compilação e execução dos programas editados através de botões localizados na barra de ferramentas ou através de teclas de atalho e também uma “janela de mensagens” onde é exibido separadamente o log das tarefas realizadas, as mensagens do compilador, uma instância do Shell do sistema além de outras funcionalidades.

Para a instalação do Geany-0.11 foi preciso adquirir e descompactar o pacote geany-0.11.tar.bz2. Posteriormente foi necessário compilar, instalar e alterar a forma de condificação padrão para os novos arquivos.

```
$ cd /build/geany-0.11
$ ./configure --prefix=/usr
$ make
$ make install
$ sed -i 's/pref_editor_default_encoding=UTF-8/pref_editor_default_encoding=ISO-8859-1' \ /root/.geany/geany.conf
```

40 IDE: acrônimo para *Integrated Development Environment*: é um software de desenvolvimento que reúne características e ferramentas para agilizar este processo.

41 Highlighting: É um recurso que alguns editores de código-fonte possuem, consiste em colorir a fonte de acordo com a categoria dos itens.

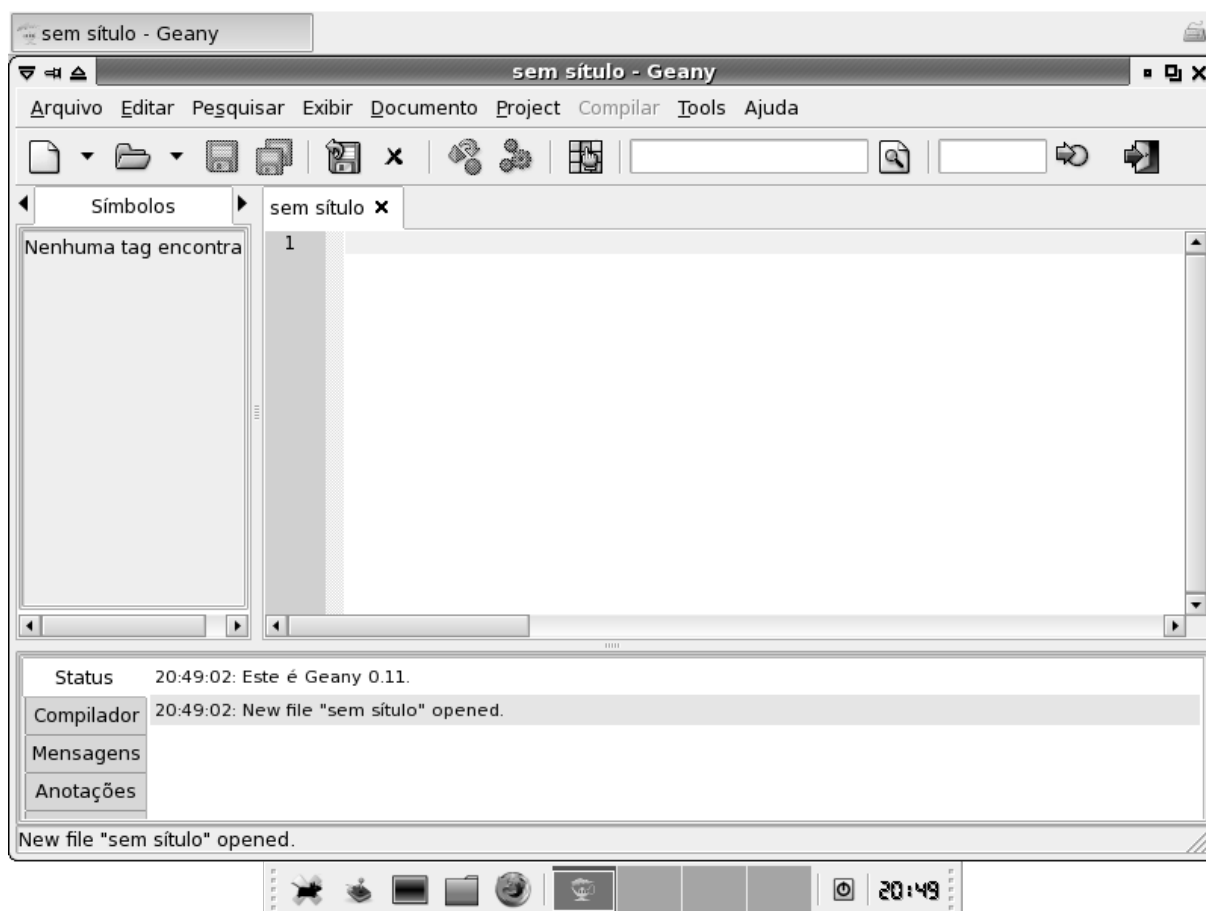


Ilustração 6: IDE de desenvolvimento Geany sendo executado no Live-CD.

- ♦ DBDesigner-4.0.5.4 e kyllixlibs3-borqt

O DBDesigner é um software de licença Open Source para modelagem de banco de dados, suporta MySQL, Oracle, Sqlite, MSSQL.

Para a instalação do DBDesigner-4.0.5.4 foi necessário adquirir o pacote dos fontes e as bibliotecas kyllix-borqt.

Primeiramente foi preciso descompactar ambos os pacotes e posteriormente mover as bibliotecas kyllix e do DBDesigner para os diretórios necessários.

```
$ mv /build/kyllixlibs3-borqt/libborqt-6.9.0-qt2.3.so /usr/lib
$ ln -s /usr/lib/libborqt-6.9.0-qt2.3.so /usr/lib/libborqt-6.9-qt2.3.so
$ cp -r /build/DBDesigner4 /opt
$ sed -i 's/export QT_XFT=true/#export QT_XFT=true/' \
/opt/DBDesigner4/startdbd
$ sed -i 's/export CLX_USE_LIBQT=yes/#export CLX_USE_LIBQT=yes/' \
/opt/DBDesigner4/startdbd
```

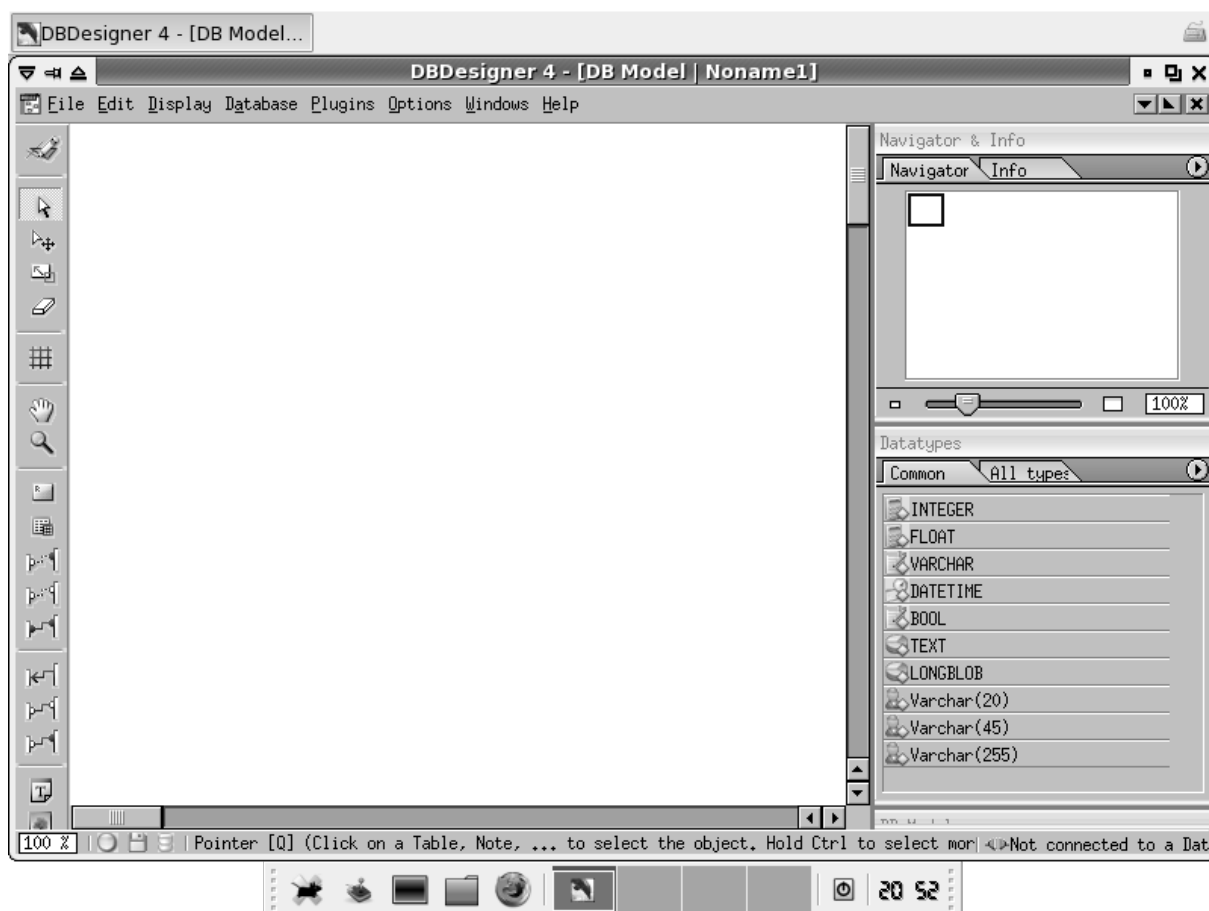


Ilustração 7: DBDesigner sendo executado no Live-CD.

- ◆ Glade3-3.0.0

O Glade é uma ferramenta de rápido desenvolvimento de interfaces de usuário. As interfaces de usuário desenhadas nesta são guardadas em um arquivo XML e usando as bibliotecas do libglade elas podem ser dinamicamente carregadas por aplicações conforme o necessário. Podem ser usadas em várias linguagens de programação, incluindo C, C++, Java, Perl, Python etc. (GLADE, 2007).

Para instalar o Glade foi necessário obter o pacote através do website oficial (<http://glade.gnome.org/>) e executar os comandos que segue após a descompressão:

```
$ cd /build/glade3-3.0.0/
```

```
$ ./configure --prefix=/usr
$ make
$ make install
```

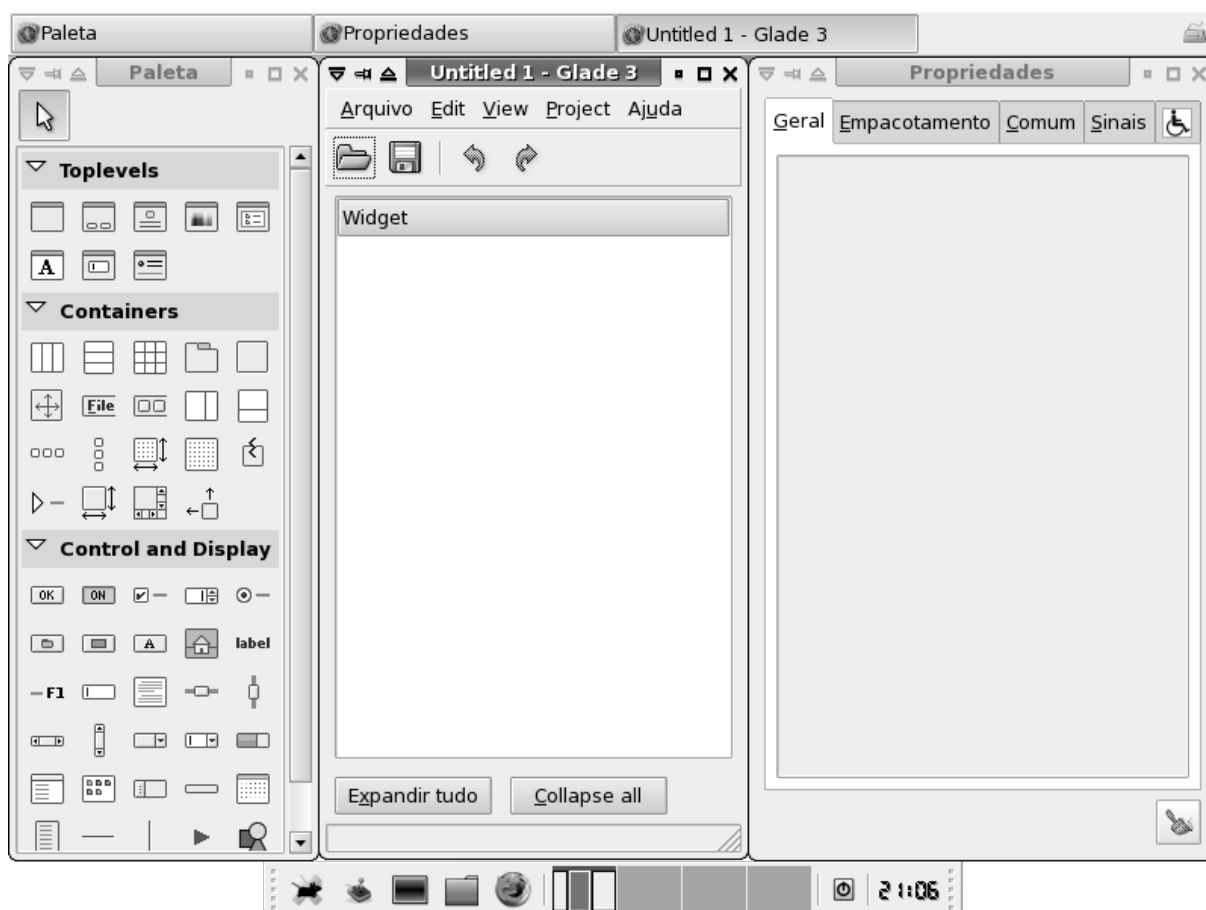


Ilustração 8: Glade 3 sendo executado no Live-CD.

- ◆ Pygobject-2.12.1, Pycairo-1.2.2, Libglade-2.6.0 e PyGTK-2.10.6

Para resolução de dependências e para detecção das bibliotecas, a compilação e instalação dos pacotes necessitou ser feita na ordem que segue.

O PyGObject fornece um *wrapper* para o GObject+, biblioteca para uso em programas Python, e trata de muitos detalhes como gerenciamento de memória. Quando combinado com PyGTK, PyORBit e gnome-python, todos os seus recursos podem ser utilizados em aplicações do Gnome (SLACKWARE, 2007).

Instalação do Pygobject-2.12.1:

```
$ cd /build/pygobject-2.12.1
$ ./configure --prefix=/usr
$ make
$ make install
```

De acordo com Slackware (2007b) o Pycairo é um conjunto de bibliotecas utilizado para permitir ao Python a utilização da bibliotecas gráficas do Cairo⁴².

Instalação do Pycairo-1.2.1:

```
$ cd /build/pycairo-1.2.2
$ ./configure --prefix=/usr
$ make
$ make install
```

Libglade é uma pequena biblioteca que permite que um programa carregue sua interface do gráfica a partir de um arquivo XML(interface do usuário gerada no Glade) em tempo de execução (HENSTRIDGE, 2002).

Instalação do Libglade-2.6.0:

```
$ cd /build/libglade-2.6.0
$ ./configure --prefix=/usr
$ make
$ make install
```

O PyGTK é um *wrapper* que permite a criação de programas com uma interface gráfica utilizando linguagem Python, de maneira fácil. Ele usa o *toolkit*⁴³ GTK+ e oferece um conjunto abrangente de elementos gráficos e outros meios úteis de programação como manuseio de texto estrangeiro, etc. (PYGTK, 2007).

Instalação do PyGTK-2.10.6:

```
$ cd /build/pygtk-2.10.6
$ ./configure --prefix=/usr
$ make
$ make install
```

⁴² Cairo: é uma biblioteca usada para prover design gráfico baseado em vetores.

⁴³ Toolkit: biblioteca de rotinas que define reutilização de uma solução de arquitetura (*design*).

♦ Net-tools-1.60

Net-tools é um pacote que contém uma coleção de programas para controlar o subsistema de rede do Kernel. Como exemplo de programas contidos podemos citar o `ifconfig` (é o principal utilitário de configuração de rede), `netstat` (usado para listar estatísticas as conexões, tabela de roteamento das interfaces de rede) e `route` (usado para manipular a tabela de roteamento de IP);

Para instalação do Net-tools-1.60 foi preciso adquirir o pacote de fontes `net-tools-1.60.tar.bz2` através do website oficial e os patches de correção através dos repositórios do LFS. Os seguintes comandos⁴⁴ foram executados para a aplicação do patches de correção e instalação dos aplicativos:

```
$ cd /build/net-tools-1.60
$ patch -Np1 -i ../net-tools-1.60-gcc34-3.patch
$ patch -Np1 -i ../net-tools-1.60-kernel_headers-2.patch
$ patch -Np1 -i ../net-tools-1.60-mii_ioctl-1.patch
$ yes "" | make config
$ sed -i -e 's|HAVE_IP_TOOLS 0|HAVE_IP_TOOLS 1|g' \
-e 's|HAVE_MII 0|HAVE_MII 1|g' config.h
$ sed -i -e 's|# HAVE_IP_TOOLS=0|HAVE_IP_TOOLS=1|g' \
-e 's|# HAVE_MII=0|HAVE_MII=1|g' config.make
$ make
$ make update
```

♦ Java jdk1.6.0_01

Java é uma linguagem de programação orientada a objeto que, diferentemente das linguagens convencionais, não é compiladas para código nativo, a linguagem Java é compilada para um *bytecode* que é executado por uma máquina virtual. Desenvolvida inicialmente na empresa Sun Microsystems por volta de 1991, atualmente a tecnologia é encontrada em redes e dispositivos que vão desde a Internet e supercomputadores científicos a laptops e telefones celulares (JAVA, 2007).

A instalação do Java no Live-CD foi muito simples, bastou executar o pacote

⁴⁴ Uma descrição detalhada dos comandos pode ser encontrada no capítulo X do livro BLFS.

binário `jdk-6u1-linux-i586.bin`, mover o diretório criado para `/opt` e registrar a variável de ambiente `JAVA_HOME`.

```
$ cd /build
$ ./jdk-6u1-linux-i586.bin
$ mkdir /opt/java6
$ mv jdk1.6.0_01 /opt/java6
```

Os comandos acima foram necessários para a instalação, mas para que ele fosse reconhecido pelas aplicações foi necessário acrescentar três linhas ao arquivo `/etc/profile`:

```
JAVA_HOME=/opt/java6/jdk1.6.0_01
PATH=$PATH:$JAVA_HOME/bin
export JAVA_HOME
```

4.4.3 Desenvolvimento do “Painel do Live-CD”

O desenvolvimento de uma aplicação para facilitar o acesso aos recursos do Live-CD como proposto, foi realizado no sistema anfitrião e ao final transferido para o Live-CD. O aplicativo chamado de “Painel do Live-CD” foi desenvolvido em linguagem Python com uso do *wrapper* PyGTK e sua interface gráfica foi construída com o aplicativo Glade 2.12.1.

O Python e as bibliotecas do PyGTK necessárias acompanham a distribuição Ubuntu 7.04, restando apenas a instalação do Glade:

```
$ sudo apt-get install glade
```

A facilidade de interagir com o sistema operacional, a simplicidade da linguagem e a necessidade de uma interface gráfica amigável foram os principais motivos da utilização desta linguagem/ferramentas.

O “Painel do Live-CD” oferece ao usuário uma interface para acesso à IDEs de desenvolvimento e software de modelagem de banco de dados além de possibilitar algumas configurações no sistema e acesso a URLs de documentação.

O intuito desta aplicação é auxiliar os usuários iniciantes a realizar tarefas simples, para isso o aplicativo foi dividido em abas e o conteúdo de cada assunto foi organizado separadamente.

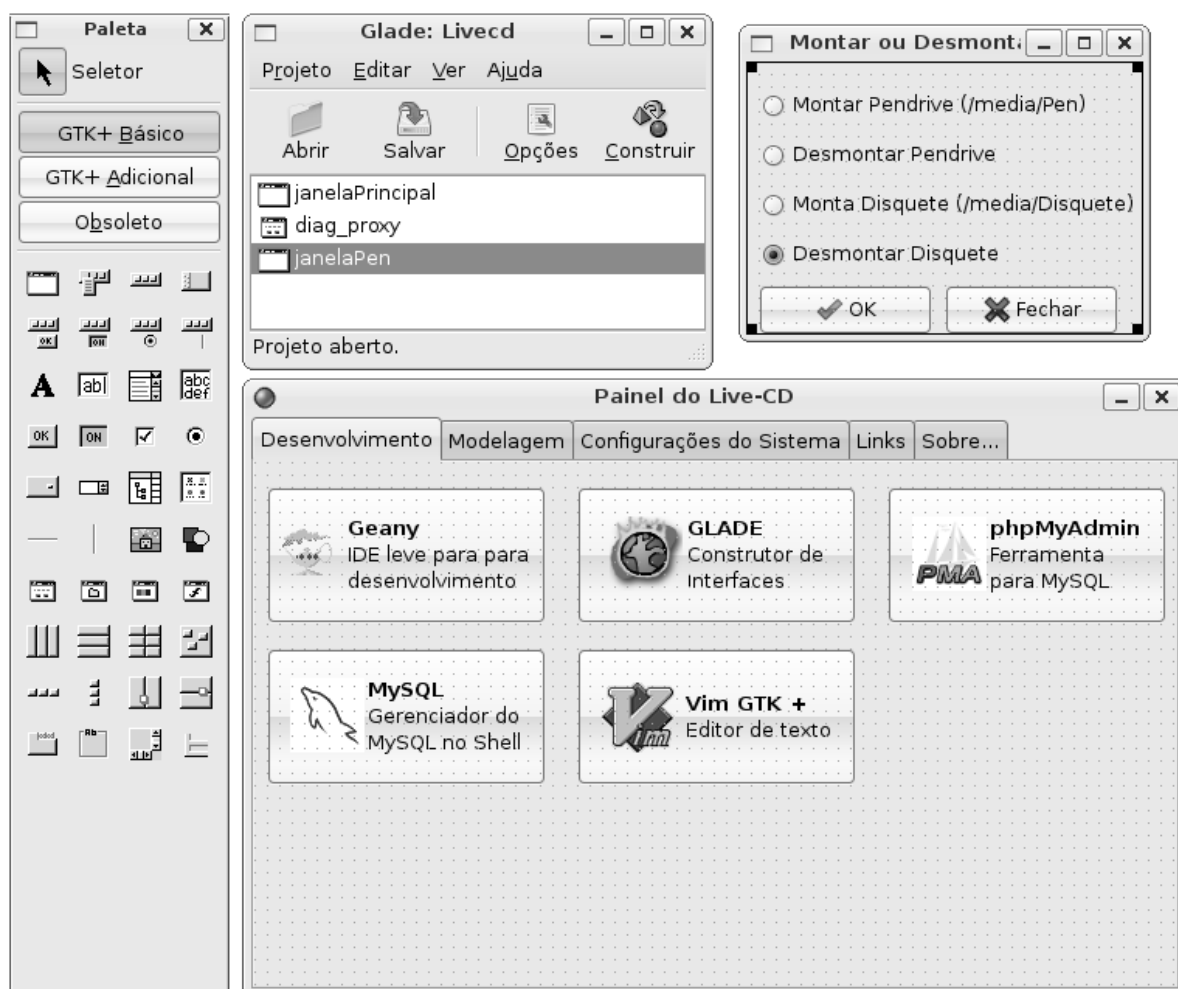


Ilustração 9: Desenvolvimento da interface de usuário do Painel do Live-CD

Durante o processo de desenvolvimento os arquivos criados foram mantidos no diretório Painel da pasta pessoal do usuário no sistema anfitrião, trata-se do arquivo *livecd.glade* que contém o código XML que será interpretado pelo Python/Libglade para exibição da interface gráfica, do arquivo *livecd.py* que contém o código-fonte Python (classe principal e métodos) da aplicação e dos arquivos de imagens usados como ícone na aplicação.

O trecho de código abaixo mostra a incorporação das biblioteca do PyGTK entre outras e o método construtor da classe *Aplicacao*. O código-fonte completo da aplicação pode ser conferida no anexo III ou adquirido através do URL <http://www.unochapeco.edu.br/~clemente/livecd/>.

```
#!/usr/bin/python
#Arquivo: livecd.py
#Autor: Clemente Scaratti clementesc@gmail.com

import pygtk
pygtk.require("2.0")
import gtk, gtk.glade, sys, os

class Aplicacao:

    def __init__(self):
        """
        Metodo Construtor da classe
        """
        #Carrega a interface a partir do arquivo glade
        self.Painel = gtk.glade.XML('livecd.glade')

        #Associa os widgets a variaveis
        self.janela = self.Painel.get_widget('janelaPrincipal')
        self.dproxy = self.Painel.get_widget('diag_proxy')
        self.janPen = self.Painel.get_widget('janelaPen')

        #Conecta Sinais aos Callbacks
        self.Painel.signal_autoconnect(self)

        #Exibe toda interface
        self.janela.show_all()

        #Inicia o loop principal de eventos (GTK MainLoop)
        gtk.main()
```

O acesso aos aplicativos de desenvolvimentos acontece através da aba “Desenvolvimento”, onde reúnem-se IDEs, editores entre outros.

A aba “Modelagem” permite acesso a aplicativos de modelagem de banco de dados.

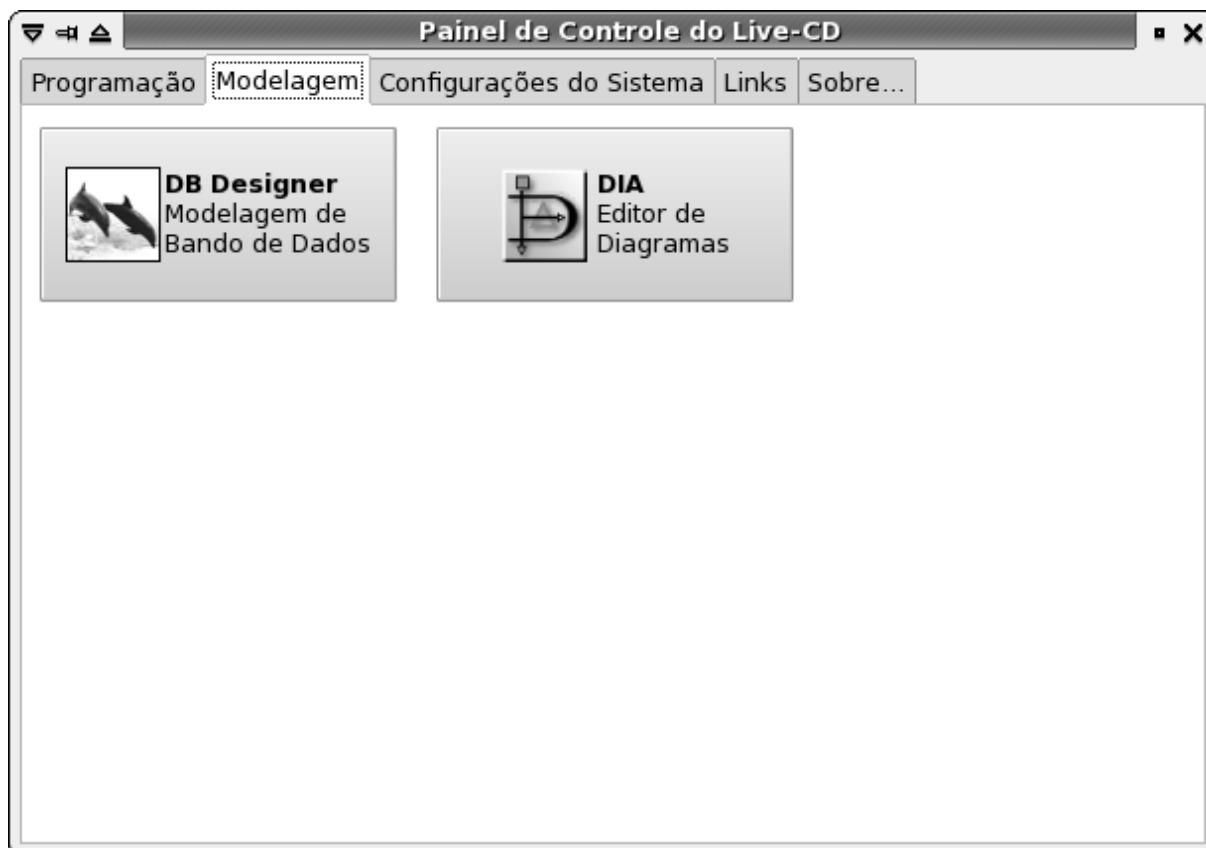


Ilustração 10: Painel do Live-CD - Tela que oferece ao usuário o acesso a aplicativos de modelagem.

Na aba “Configurações do Sistema” encontram-se reunidos alguns recursos que permitem iniciar/parar serviços do sistema operacional, efetuar a configuração completa de IP, exportar configurações de proxy para o Shell além da montagem e desmontagem de dispositivos removíveis (disquetes e pendrives).

As abas “Links” e “Sobre” permitem o acesso a websites oficiais dos recursos do Live-CD e de documentação, e a segunda traz os créditos ao desenvolvedor.

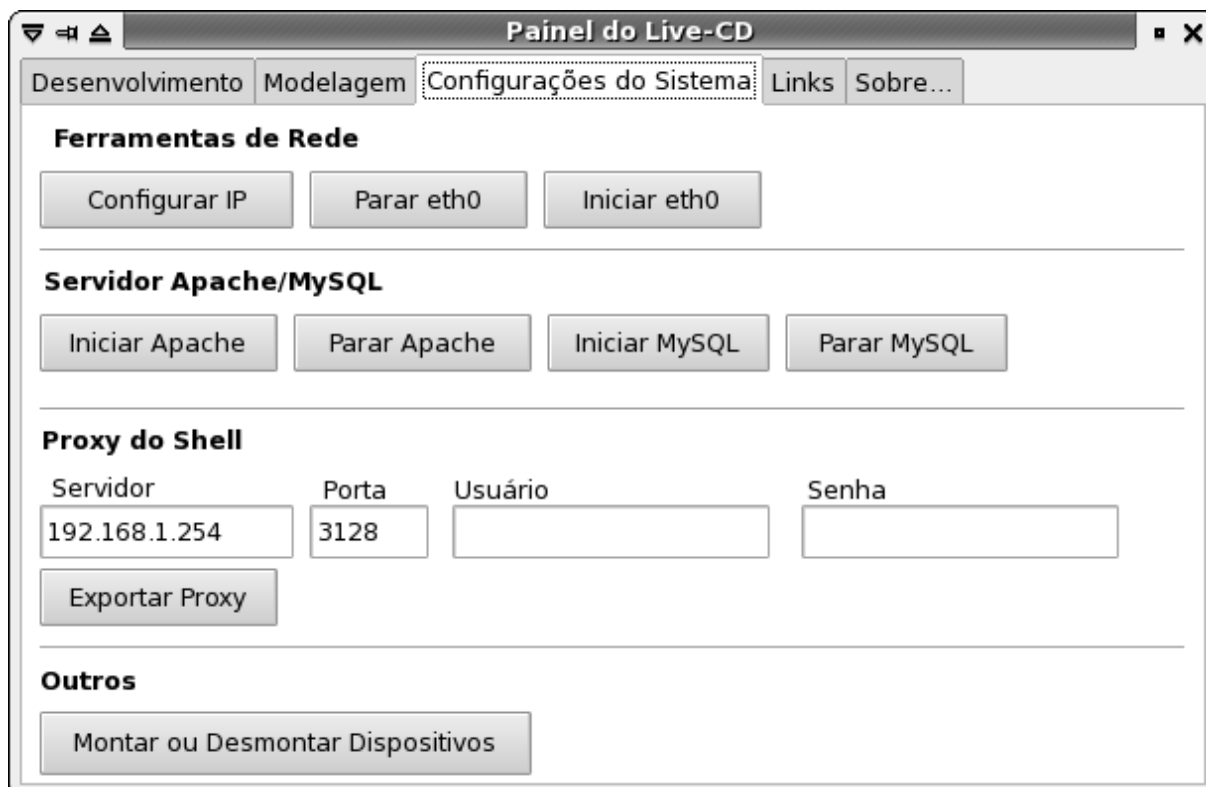


Ilustração 11: Painel do Live-CD - Possibilidade de realizar algumas configurações no sistema.

O ambiente de funcionamento do Painel do Live-CD, bem como outros aplicativos desenvolvidos com esta tecnologia, é orientado a eventos, sendo que uma ação pode ocorrer a qualquer momento. Desta forma, um método da classe *Aplicacao* é executado ao clique do mouse sobre um botão, por exemplo.

Após o término do desenvolvimento a aplicação foi testada e o diretório Painel foi copiado para o diretório `/root/build` (diretório de acesso comum entre o sistema anfitrião e o ambiente chroot no processo de remasterização) para ser também testada no Live-CD.

4.4.4 Personalização do XFCE

O XFCE foi o ambiente desktop escolhido, por se muito leve, tornou-se ideal para o Live-CD, além de já estar presente na distribuição LFS Oficial.

De acordo com Xfce (2007), o ambiente é leve e rápido e ao mesmo tempo estável e atraente. Ele disponibiliza uma série de aplicações incluindo editor de texto, software para gravação de CD para o usuário final, sempre objetivando rapidez na execução, além de permitir a gerência de múltiplos painéis para adição de menus e atalhos, preferências de aplicações e plugins adicionais.

O ambiente XFCE teve que ser personalizado para atender as alterações e instalações de novos aplicativos no sistema. Essas modificações ocorreram na área de trabalho, menu e painel do XFCE, basicamente para gerenciamento dos atalhos.

- ♦ Papel de Parede

A alteração do papel de parede o XFCE foi realizada com a edição do arquivo `desktop.xml`, localizado no diretório `/etc/xdg/xfce4/mcs_settings`. O valor da variável `imagepath_0_0` passou a conter o caminho completo da imagem agora exibida como papel de parede. Esta imagem é parte do XFCE e já residia no Live-CD.

- ♦ Painel do XFCE

O XFCE possui um painel que permite o acesso à aplicativos, configurações, tarefas, etc. Neste painel foi acrescentado um ícone para acesso ao “Painel do Live-CD”. Foi também removido deste painel o atalho para as aplicações `xfce-settings-show` e `xfhelp4`, e alterado a linha de comando para a execução do navegador Mozilla Firefox. Estas alterações foram feitas através da edição do arquivo `/etc/xdg/xfce4/panel/contents.xml`. Ainda no painel do XFCE, existe um botão responsável por mostrar o “menu do XFCE”, este faz a leitura do arquivo `/etc/xdg/xfce4/desktop/menu.xml.pt_BR` que possui configurações para acesso a diversas aplicações e configurações do ambiente XFCE. Neste arquivo modificações foram realizadas para habilitar o atalho de abertura do navegador de internet Mozilla Firefox.

- ♦ Atalhos do XFCE

Alguns itens do menu do XFCE estavam em idioma estrangeiros e foram

traduzidos. Esta etapa consistiu em conferir todos os arquivos residentes no diretório `/usr/share/applications` e editar os necessários.

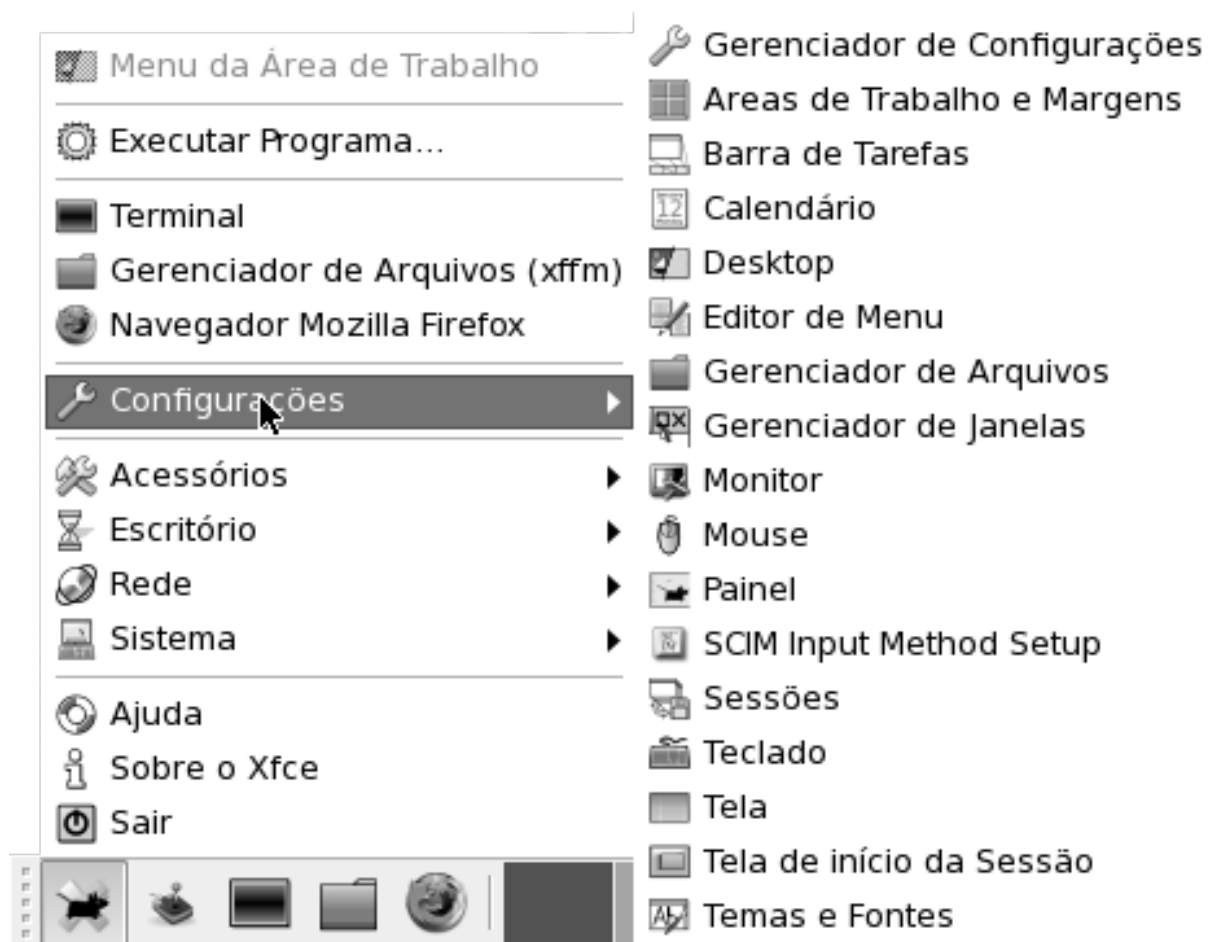


Ilustração 12: Menu do XFCE, após a tradução dos itens.

4.4.5 Personalização do ISSUE

O arquivo `/etc/issue` contém informações sobre a distribuição e é exibido na abertura de cada sessão do console do sistema. O conteúdo deste arquivo ficou disposto da seguinte maneira.

Esta é uma derivação do Oficial Linux From Scratch Live-CD.
É a implementação prática do trabalho de conclusão de curso:

CUSTOMIZAÇÃO DE UMA DISTRIBUIÇÃO LINUX LIVE-CD PARA SER UTILIZADA

NOS CURSOS DE CIÊNCIA DA COMPUTAÇÃO E SISTEMAS DE INFORMAÇÃO DA UNOCHAPECÓ

Orientando: Clemente Scaratti

Orientador: Elton Luís Minetto

Website: <http://www.unochapeco.edu.br/~clemente/livecd>

 Para Configurar o IP de forma estática execute: `net-setup`
 Para Iniciar o ambiente gráfico execute: `startx`

4.4.6 Personalização do Splash Screen

Em geral ao iniciar uma distribuição Linux Live-CD é mostrado ao usuário uma tela de apresentação, podendo esta conter informações de configurações para a inicialização e/ou uma imagem personalizada.

Esta imagem corresponde a um arquivo na árvore de diretório de CD: `/boot/isolinux/splash.lss`. Este arquivo é uma imagem de 16 cores compactada (RYSANEK, 2003).

De acordo com Rysanek (2003), a edição direta desta imagem não é permitida, portanto teve que ser transformada para o formato ppm. Tanto o processo de conversão quanto o de edição foi realizado fora do ambiente do chroot. Para a transformação do arquivo de imagem foram necessários os aplicativos *lss16toppm* e *ppmtolss16*, ambos adquiridos com a instalação do pacote *netpbm*. Para instalar o seguinte comando foi executado no sistema anfitrião:

```
sudo apt-get install netpbm
```

Agora a conversão da imagem para um formato editável:

```
$ lss16toppm1 < splash.lss> splash.ppm
```

A saída do comando acima foi um arquivo chamado `splash.pmm` nas dimensões de 640 x 480 pixels. Após a edição com o Gimp 2.2 ela foi novamente

convertida, agora para o formato original (LSS). Esta conversão foi realizada com a aplicação do comando:

```
$ ppmtolss16 <splash .ppm> splash.lss
```

Durante a edição o padrão de cores foi mantido, caso contrário neste momento seriam truncadas as cores que estivessem fora do formato RGB 6 bits.

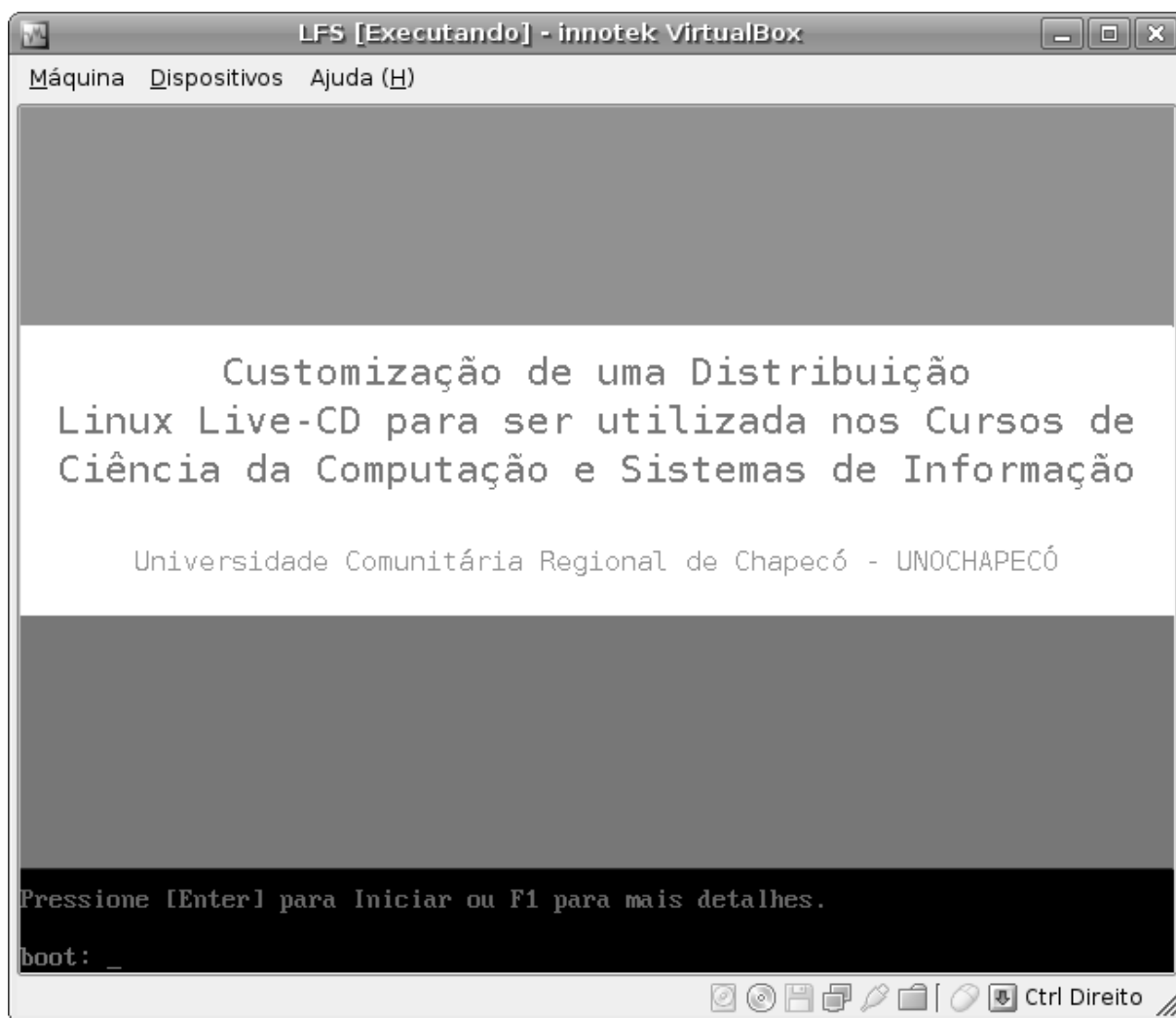


Ilustração 13: Amostra do Splash Screen durante a emulação do Live-CD no aplicativo VirtualBox.

4.4.7 Geração da imagem ISO

O primeiro passo realizado após a primeira entrada no ambiente chroot foi a criação de uma nova imagem `initramfs_data_cpio.gz` personalizada. Esta fará parte do gerenciador de boot e precisou substituir a original no diretório `/root/iso/boot/isolinux`.

```
$ mv /build/tags/6.2-1/packages/initramfs/initramfs_data_cpio.gz \
/root/iso/boot/isolinux
```

O Shell script `iso.sh`, desenvolvido para automatizar a geração da imagem ISO, foi utilizado para finalizar o processo de remasterização.

O comando abaixo efetua o processo de compressão do arquivo `/root/root.ext2` e, com uso do aplicativo `mkisofs`, gera a imagem ISO da nova distribuição Live-CD.

```
$ cd /root/scripts
$ . iso.sh
```

No microcomputador utilizado, o término do script demora em média 10 minutos e ao final a imagem `/root/lfslivecd-x86-6.2-UNOLIVE.iso` com tamanho de 545 MB está pronta para ser gravada em um mídia de CD/DVD.

Este processo final ocorreu diversas vezes antes do término do trabalho, pelo menos uma vez a cada novo pacote instalado, tudo isso para realização de testes. É evidente que nem todas as vezes uma mídia de CD/DVD foi queimada, então o aplicativo innotek VirtualBox foi utilizado.

O VirtualBox é uma máquina virtual⁴⁵ de licença opensource, que permite a execução de vários sistemas operacionais ao mesmo tempo, uma prática que vem se tornando muito comum entre os usuários e que facilita a adaptação dos que desejam se aventurar em diversos sistemas ou até mesmo executar testes com utilitários. O VirtualBox possui versões para os sistemas operacionais Windows, Linux

⁴⁵ Máquina Virtual: é um software que cria um ambiente entre a plataforma e o utilizador final, onde este pode operar outros softwares.

e ainda uma versão para Mac. Uma série de recursos estão disponíveis no VirtualBox, como por exemplo a implementação de controladores USB virtuais, permitindo ligar os dispositivos USB nas máquinas virtuais sem necessidade de configurações adicionais no sistema host (VIRTUALBOX, 2007).

Para a realização dos testes VirtualBox foi preciso configurá-lo. As ilustrações a seguir auxiliam o entendimento deste processo.

No primeiro acesso ao aplicativo foi necessário clicar no botão Novo para abrir o assistente de criação de Máquina Virtual. A primeira tela exibida é de boas-vindas e na próxima tela foi preciso informar um nome e um tipo de sistema, conforme a figura 14.



Ilustração 14: Assistente para criação da Máquina Virtual - Passo 1.

O próximo passo foi selecionar a quantidade de memória que o sistema

anfitrião necessita reservar para a Máquina Virtual.



Ilustração 15: Assistente para criação da Máquina Virtual - Passo 2.

Finalmente ao concluir o assistente, um novo item chamado LFS é adicionado na lista das Máquinas Virtuais registradas.

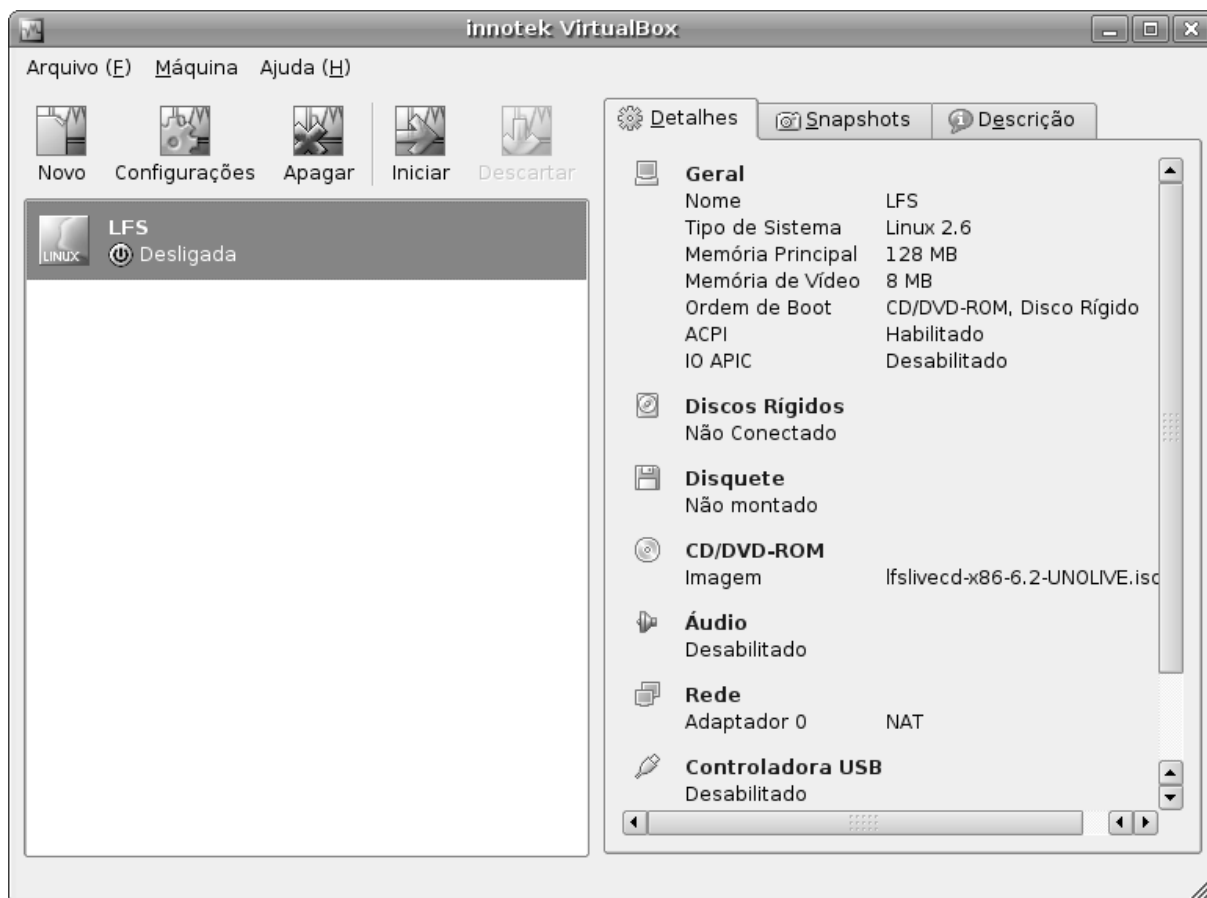


Ilustração 16: Tela principal do aplicativo VirtualBox mostrando a lista de Máquinas Virtuais registradas.

Agora basta um simples clique no botão Iniciar e o processo de boot no Live-CD inicia em uma nova janela aberta pelo aplicativo.

5. CONCLUSÕES E TRABALHOS FUTUROS

O presente trabalho teve como objetivo acompanhar o processo de customização de uma distribuição Linux, descrevendo paralelamente os métodos disponíveis, seus requisitos e características.

Dos métodos apresentados, dois deles foram utilizados onde na primeira etapa utilizou-se o Linux From Scratch e na segunda o da Remasterização. Com relação ao primeiro, muitas dificuldades foram encontradas devido a pouca documentação em idioma Português e a complexidade do processo, podendo esta ser notada com a análise dos pré-requisitos do livro LFS. As 266 páginas de instruções e códigos do livro servem para a construção de um sistema muito básico e assumem que o usuário tenha um conhecimento razoável sobre uso e instalação de software em sistemas Linux.

Problemas decorrentes da espécie do hardware surgiram durante a compilação de pacotes essenciais como o GCC nas duas primeiras tentativas. Na terceira tentativa o que impediu a conclusão dos passos de todos os capítulos do livro foram os problemas referentes aos *locales* e internacionalização. A partir daí procurou-se ferramentas para aprimorar o processo de construção, então a pesquisa resultou em dois conjuntos de Shell scripts de automatização e, após uma análise e comparação um deles acabou por ser utilizado e tornou-se fundamental do desenvolvimento deste trabalho.

Do Linux From Scratch, um conhecimento muito grande foi adquirido na área de sistemas operacionais Linux, sobre suas funcionalidades, sua integração e as

dependências de suas partes além da linguagem Shell script.

Com o sistema básico montado e funcionando, na segunda etapa tratou-se de adicionar recursos ao Live-CD, para isto o método de remasterização foi utilizado. Com o auxílio do livro BLFS foram acrescentados algumas ferramentas a fim de atender as necessidades dos professores e alunos em aulas de laboratório.

O desenvolvimento da aplicação usando Python, além de atribuir um valor a este trabalho, acima de tudo trouxe conhecimento e interesse pela linguagem, tudo isso por ser de livre licença e possuir uma semântica dinâmica.

O assunto deste não é novo, mas existe uma escassez de livros que o tratam e sendo assim a maioria do material utilizado nas pesquisas vieram de artigos e notícias publicados em websites da internet, tendo estes que passar por uma análise a fim de reconhecer o verdadeiro valor científico.

Depois deste estudo, pode-se afirmar que, assim como não existe a melhor distribuição, também não existe o melhor método de se personalizar uma distribuição pois depende muito da necessidade e conhecimento do usuário. Se o usuário possuir pouco conhecimento em Linux, o ideal é utilizar a remasterização que é um método simples e possui bastante artigos no idioma português para servirem de suporte. Se o usuário é um bom conhecedor de Linux e quer algo voltado totalmente ao contexto de suas necessidades, o método recomendado é o Linux From Scratch que, apesar de custar bastante tempo e ser complexo, gera bons resultados.

5.1 Trabalhos Futuros

Com relação aos trabalhos futuros, poderia ser indicado:

- ♦ Aperfeiçoamento no Painel do Live-CD, acrescentando novas funcionalidades e melhorando as existentes.

- ♦ Melhoramento no Shell script de detecção de dispositivos para torná-lo flexível quanto a detecção de pendrives na presença de HDs de tecnologia *Sata*. A nomenclatura usada para os HDs *Sata* é a mesma para os dispositivos removíveis, podendo com isso resultar em uma montagem indesejada ou falha.
- ♦ Criação de um Shell script capaz de transferir o conteúdo do Live-CD para um pendrive e torná-lo bootável.
- ♦ Customizar a distribuição usando como base o Live-CD 6.3 oficial, recentemente lançado.
- ♦ Inclusão de outras aplicações, úteis para as mais diversas disciplinas dos cursos de Ciência da Computação e Sistemas de Informação e posteriormente estender também à outros cursos.
- ♦ Realizar outros testes para identificar possíveis incompatibilidades com os diversos de hardwares encontrados no campo computacional da instituição.

6. REFERÊNCIAS

ALECRIM, Emerson. **O Kernel do Linux**. 2004. Disponível em: <http://www.infowester.com/linuxkernel.php>. Acesso em: 23 out. 2006.

ALECRIM, Emerson. **Apresentando o KDE**. 2006a. Disponível em: <http://www.infowester.com/kde.php>. Acesso em: 30 out. 2006.

ALECRIM, Emerson. **Apresentando o GNOME**. 2006b. Disponível em: <http://www.infowester.com/gnome.php>. Acesso em: 30 out. 2006.

APACHE. **About the Apache HTTP Server Project - The Apache HTTP Server Project**. 2007a. Disponível em: http://httpd.apache.org/ABOUT_APACHE.html. Acesso em 01 nov. 2007

APACHE. **Apache module mod_dir**. 2007b. Disponível em: http://httpd.apache.org/docs/1.3/mod/mod_dir.html. Acesso em 01 nov. 2007

ACHOUR, Mehdi et al. **PHP: Preface - Manual**. 2007. Disponível em: <http://www.php.net/manual/en/preface.php>. Acesso em 20 out. 2007.

BEEKMANS, Gerard. **Linux From Scratch**. 2005. Disponível em: http://codigolivre.org.br/frs/download.php/4401/LFS-BOOK-6.1-pt_BR-7.pdf. Acesso em: 14 out. 2006.

BEEKMANS, Gerard. **What is Automated Linux From Scratch?**. 2007a. Disponível em: <http://www.linuxfromscratch.org/alfs/>. Acesso em: 7 set. 2007.

BEEKMANS, Gerard. **What is Beyond Linux From Scratch?**. 2007b. Disponível em: <http://www.linuxfromscratch.org/blfs/>. Acesso em: 7 set. 2007.

BEEKMANS, Gerard. **What is Linux From Scratch?**. 2007c. Disponível em: <http://www.linuxfromscratch.org/lfs/>. Acesso em: 7 set. 2007.

BEEKMANS, Gerard. **Linux From Scratch**. 2007d. Disponível em: <http://www.linuxfromscratch.org/lfs/view/6.2/>. Acesso em: 2 jan. 2007.

BEEKMANS, Gerard. **Beyond Linux From Scratch**. 2007e. Disponível em: <http://www.linuxfromscratch.org/blfs/view/6.2.0/>. Acesso em: 2 jan. 2007.

CAMPOS, Augusto. **Qual a melhor distribuição de Linux?**, 2005. Disponível em <http://br-linux.org/linux/qual-a-melhor-distribuicao-de-linux>. Acesso em: 22 set. 2006.

CAMPOS, Augusto. **O que é software livre**. 2006. Disponível em <http://br-linux.org/linux/faq-softwarelivre>. Acesso em: 22 set. 2006.

DANESH, Arman. **Dominando o Linux: a bíblia**. São Paulo: Makron Books, 1999. 602 p.

DREAMLINIX. **O Dreamlinux**. 2006a. Disponível em: <http://www.dreamlinux.com.br/portugues/index.html>. Acesso em: 02 nov. 2006.

DREAMLINIX. **PROJETO DREAMLINIX**. 2006b. Disponível em: <http://www.dreamlinux.com.br/portugues/saiba-tutor.html>. Acesso em: 02 nov. 2006.

DREAMLINIX. **MKDistro um tutorial básico**. 2006c. Disponível em: http://dreamlinux.incubadora.fapesp.br/portal/tutoriais/mkdistro_basic_tutorial/mkdistro_pt1.html. Acesso em: 02 nov. 2006.

EDUBUNTU. **What is Edubuntu?**. 2006. Disponível em: <https://wiki.edubuntu.org/EdubuntuWiki>. Acesso em: 14 nov. 2006.

ENEC. **Software Livre, uma Perspectiva revolucionária**. 2004. Disponível em: <http://www.enec.org.br/cgi-bin/twiki/view/ENEC/SLperspectivaRevolucionaria>. Acesso em: 01 out. 2006.

GIBBS, Mark. **Open source opens opportunities**. 2005 Disponível em: <http://www.linuxworld.com.au/index.php/id;1928390333;fp;16;fpid;0>. Acesso em: 05 out. 2006.

GLADE. **Glade - a User Interface Designer for GTK+ and GNOME**. 2007. Disponível em: <http://glade.gnome.org/>. Acesso em 01 nov. 2007

GOBLINX. **Mini Tutor Homepage**. 2006a. Disponível em: <http://www.goblinx.com.br/>. Acesso em: 01 nov. 2006.

GOBLINX. **Bem vindo a Caverna**. 2006b. Disponível em: <http://www.goblinx.com.br/minitutors.htm>. Acesso em: 01 nov. 2006.

HENSTRIDGE, James. **libglade - GTK 2.0 branch**. 2002. Disponível em: <http://freshmeat.net/projects/libglade/>. Acesso em 01 nov. 2007

HEXSEL, Roberto. **O que é Software Livre?**. 2006. Disponível em: <http://www.softwarelivre.gov.br/SwLivre/>. Acesso em: 21 set. 2006.

JAVA. **Saiba mais sobre a tecnologia Java**. 2007. Disponível em: http://www.java.com/pt_BR/about/. Acesso em: 12 ago. 2007.

KUBUNTU. **Kubuntu**. 2006. Disponível em: <https://wiki.kubuntu.org/Kubuntu>. Acesso em: 14 nov. 2006.

MANDRIVA CONECTIVA. **Fundamentos do Sistema Linux**. Conectiva S.A., 2005.

MARQUES, Tiago. **LiveCD**. 2007. Disponível em: <http://glua.ua.pt/LiveCD>. Acesso em: 01 out. 2007.

MAXWELL, Scott. **Kernel do Linux**. São Paulo: Makron Books, 2000. 308 p.

MORIMOTO, Carlos E.. **Entendendo como o Linux funciona – A árvore genealógica das distribuições**. 2005a. Disponível em: <http://www.guiadohardware.net/tutoriais/097/index7.php>. Acesso em: 10 out. 2006.

MORIMOTO, Carlos E.. **Instalando o Kurumin (e outras distros) num pendrive ou cartão**. 2005b. Disponível em: <http://www.guiadohardware.net/tutoriais/108/>. Acesso em: 10 out. 2006.

MORIMOTO, Carlos E.. **O Kernel e as distribuições Linux**. 2006a. Disponível em: <http://www.guiadohardware.net/guias/06/index8.php>. Acesso em: 10 out. 2006.

MORIMOTO, Carlos E.. **Kurumin: Desvendando seus segredos**. 2006b. Disponível em: <http://www.guiadohardware.net/livros/kurumin/09/>. Acesso em: 20 out. 2006.

MORIMOTO, Carlos E.. **Kurumin Linux** . 2006c. Disponível em: <http://www.guiadohardware.net/gdhpress/kurumin/>. Acesso em: 10 out. 2006.

NILSEN, Mark. **How to use a ramdisk for linux**. 1999. Disponível em: <http://www.linuxfocus.org/English/November1999/article124.html>. Acesso em: 05 out. 2006

NORTON, Peter; GRIFFITH, Arthur. **Guia completo do Linux**. 2. ed. São Paulo: Berkeley, 2002. 597 p.

OSXBOOK. **Knoppix on the Intel-based Macintosh**. 2006. Disponível em: <http://www.osxbook.com/book/bonus/misc/knoppix/>. Acesso em: 01 nov. 2006.

PATRAKOV, Alexander E.. **lfscd-remastering-HOWTO**. 2007. Disponível em <http://www.linuxfromscratch.org/hints/downloads/files/lfscd-remastering-howto.txt>. Acesso em: 01 out. 2007.

PAULA, Fábio Berbert de. **Interfaces Gráficas no Linux**. 2002. Disponível em: <http://www.vivaolinux.com.br/artigos/verArtigo.php?codigo=12#>. Acesso em: 29 nov. 2006.

PyGTK. **PyGTK: GTK+ for Python**. 2007. Disponível em: <http://www.pygtk.org/>. Acesso em 01 nov. 2007

RYSANEK, Frank. **ISOLINUX/SYSLINUX splash image mini-HOWTO**. 2003. Disponível em: <http://www.sweb.cz/Frantisek.Rysanek/splash/isolinux-splash-HOWTO.html>. Acesso em: 01 out. 2007.

SILBERSCHATZ, Abraham; GALVIN, Peter; GAGNE, Greg. **Sistemas Operacionais Conceitos e Aplicações**. Tradução: RIECHE, Adriana Ceschin. Rio de Janeiro: Campus, 2000. 585 p.

SCRATCH. **What is Linux From Scratch?**. 2006. Disponível em: <http://wiki.linuxfromscratch.org/lfs/>. Acesso em: 25 out. 2006.

SILVA, José Messias Alves da. **Construindo On-line sua distribuição GNU/Linux customizada**. 2006. Disponível em <http://www.dicas-l.com.br/dicas-l/20060828.php>. Acesso em: 01 out. 2006.

SLACKWARE. **pygobject**. 2007a. Disponível em: <http://www.slackware.com/~alien/slackbuilds/pygobject/>. Acesso em 01 nov. 2007

SLACKWARE. **pycairo**. 2007b. Disponível em:
<http://www.slackware.com/~alien/slackbuilds/pycairo/>. Acesso em 01 nov. 2007

TANENBAUM, Andrew S.. **Sistemas operacionais modernos**. 2. ed. Rio de Janeiro: Pearson, 2004. 695 p.

TUXS. **10 things to do with a live Linux CD**. 2006. Disponível em:
<http://www.tuxs.org/dolive.htm>. Acesso em: 01 nov. 2006.

UBUNTU. **Ubuntu: Linux for human beings**. 2006. Disponível em:
<http://www.ubuntu.com/>. Acesso em: 14 nov. 2006.

VIEIRA, Gustavo. **Administração Linux – Sabendo um pouco mais**. 2006.
Disponível em:
http://www.gustavovieira.adm.br/index.php?option=com_content&task=view&id=23&Itemid=2. Acesso em: 10 out. 2006.

VIRTUALBOX. **VirtualBox -- professional, flexible, open**. 2007. Disponível em:
<http://www.virtualbox.org/wiki/VirtualBox>. Acesso em: 10 nov 2007.

XFCE. **Características do XFCE**. 2007. Disponível em:
<http://www.xfce.org/about/features>. Acesso em 10 nov 2007.

XUBUNTU. **Welcome to Xubuntu!**. 2006. Disponível em: <http://www.xubuntu.org/>.
Acesso em: 14 nov. 2006.

7. ANEXOS

Anexo I

Listagem dos pacotes e patches utilizados no processo de remasterização, bem como os URLs dos quais foram obtidos:

- ♦ `lfslivecd-x86-6.2-5.iso`
Descrição: Imagem do LFS Live-CD 6.2-5 oficial.
URL:<ftp://ftp.osuosl.org/pub/lfs-livecd/lfslivecd-x86-6.2-5.iso>
- ♦ `lfslivecd-x86-6.2-pre2-BuildScripts.tar.bz2`
Descrição: Conjunto de Shell scripts para automatizar a construção de uma distribuição usando o método Linux From Scratch.
URL:<http://www.torredehanoi.org/pub/lfs-livecd/scripts/lfslivecd-x86-6.2-pre2-BuildScripts.tar.bz2>
- ♦ `tags_6.2-1_packages_initramfs-r2089.zip`
Descrição: Pacote necessário para recriação da imagem *initramfs*.
URL:http://wiki.linuxfromscratch.org/livecd/browser/tags/6.2-1/packages/initramfs/tags_6.2-1_packages_initramfs-r2089.zip
- ♦ `blfs-bootscripts-20060910.tar.bz2`
Descrição: É um conjunto de Shell scripts utilizado para automatizar a criação de entradas nos diferentes *runlevels* do sistema.
URL:<http://www.linuxfromscratch.org/blfs/downloads/6.2.0/blfs-bootscripts-20060910.tar.bz2>
- ♦ `httpd-2.2.2.tar.bz2`
Descrição: Apache 2.2.2, é um servidor HTTP.
URL: <http://archive.apache.org/dist/httpd/httpd-2.2.2.tar.bz2>
- ♦ `phpMyAdmin-2.11.2.1-all-languages.tar.bz2`
Descrição: phpMyAdmin 2.11.2, um software de gerenciamento do MySQL.
URL:<http://ufpr.dl.sourceforge.net/sourceforge/phpmyadmin/phpMyAdmin-2.11.2.1-all-languages.tar.bz2>
- ♦ `cake_1.1.18.5850.tar.bz2`
Descrição: CakePHP 1.1.18.5850, um framework de desenvolvimento PHP.
URL:http://cakeforge.org/frs/download.php/549/cake_1.1.18.5850.tar.bz2

- ♦ symfony-stable.tgz
 Descrição: Symfony PHP 1.0.8, um framework de desenvolvimento.
 URL:<http://www.symfony-project.com/get/symfony-stable.tgz>
- ♦ php-5.2.4.tar.bz2
 Descrição: PHP 5.2.4, é uma linguagem de programação.
 URL:<http://br.php.net/get/php-5.2.4.tar.bz2/from/this/mirror>
- ♦ Python-2.5.1.tar.bz2
 Descrição: Python 2.5.1, é uma linguagem de programação.
 URL:<http://www.python.org/ftp/python/2.5.1/Python-2.5.1.tar.bz2>
- ♦ glade3-3.0.0.tar.gz
 Descrição: Glade3 3.0.0, é uma ferramenta para construção de interfaces de usuário.
 URL:<http://ftp.gnome.org/pub/GNOME/sources/glade3/3.0/glade3-3.0.0.tar.gz>
- ♦ firefox-1.5.0.9-source.tar.bz2
 Descrição: Mozilla Firefox, é um navegador de Internet.
 URL:<http://releases.mozilla.org/pub/mozilla.org/firefox/releases/1.5.0.9/source/firefox-1.5.0.9-source.tar.bz2>
- ♦ install_flash_player_9_linux.tar.gz
 Descrição: Pacote contendo o plugin do *Flash Player*, usado no Mozilla Firefox.
 URL:http://fpdownload.macromedia.com/get/flashplayer/current/install_flash_player_9_linux.tar.gz
- ♦ firebug-1.05-fx+fl.xpi
 Descrição: Firebug 1.05, é uma extensão do Mozilla Firefox.
 URL:<https://addons.mozilla.org/en-US/firefox/downloads/file/15109/firebug-1.05-fx+fl.xpi>
- ♦ switchproxy_tool-1.4.1-fx+mz+tb.xpi
 Descrição: SwitchProxy Tool 1.4.1, é uma extensão do Mozilla Firefox.
 URL:https://addons.mozilla.org/en-US/firefox/downloads/file/530/switchproxy_tool-1.4.1-fx+mz+tb.xpi
- ♦ DBDesigner4.0.5.4.tar.gz
 Descrição: DBDesigner-4.0.5.4, é um software para modelagem de banco de dados.
 URL:<http://downloads.mysql.com/DBDesigner4/DBDesigner4.0.5.4.tar.gz>
- ♦ kyllixlibs3-borqt-3.0-2.tar.gz
 Descrição: Kylixlibs3-borqt-3.0-2, é um conjunto de bibliotecas do Kylix necessária para o DBDesigner.
 URL:<http://prdownloads.sourceforge.net/kylixlibs/kylixlibs3-borqt-3.0-2.tar.gz>
- ♦ geany-0.11.tar.bz2
 Descrição: Geany 0.11, é uma IDE de desenvolvimento para várias linguagens.
 URL:<http://prdownloads.sourceforge.net/geany/geany-0.11.tar.bz2>

- ♦ dia-0.96.1.tar.bz2
Descrição: Dia diagrama 0.96, é um software para construção de diagramas.
URL:<http://ftp.gnome.org/pub/gnome/sources/dia/0.96/dia-0.96.1.tar.bz2>
- ♦ net-tools-1.60.tar.bz2
Descrição: Net-tools-1.60 , é um pacote que contém uma coleção de programas para controlar o subsistema de rede do Kernel.
URL:<http://www.tazenda.demon.co.uk/phil/net-tools/net-tools-1.60.tar.bz2>
- ♦ firefox-1.5.0.9-pangoxft-1.patch
Descrição: Patch de correção requerido pelo Mozilla Firefox pelo fato do Pango estar reindexando as fontes.
URL:<http://www.linuxfromscratch.org/patches/blfs/6.2.0/firefox-1.5.0.9-pangoxft-1.patch>
- ♦ net-tools-1.60-gcc34-3.patch
Descrição: Patch de correção requerido pelo net-tools.
URL:<http://www.linuxfromscratch.org/patches/blfs/6.2.0/net-tools-1.60-gcc34-3.patch>
- ♦ net-tools-1.60-kernel_headers-2.patch
Descrição: Patch de correção requerido pelo net-tools.
URL:http://www.linuxfromscratch.org/patches/blfs/6.2.0/net-tools-1.60-kernel_headers-2.patch
- ♦ net-tools-1.60-mii_ioctl-1.patch
Descrição: Patch de correção requerido pelo net-tools.
URL:http://www.linuxfromscratch.org/patches/blfs/6.2.0/net-tools-1.60-mii_ioctl-1.patch
- ♦ httpd-2.2.2-config-1.patch
Descrição: Patch de correção requerido pelo Apache.
URL:<http://www.linuxfromscratch.org/patches/blfs/6.2.0/httpd-2.2.2-config-1.patch>

Anexo II

Listagem dos pacotes e patches com link quebrados, bem como os alternativos utilizados, que foram encontrados durante o processamento do conjunto de Shell scripts `lfslivecd-x86-6.2-pre2-BuildScripts.tar.bz2`. Em alguns casos o mesmo arquivo não foi encontrado em outros repositórios, mas a mesma versão compactada em outro formato foi, nesses casos o *sha1sum* também teve que ser atualizado.

- ♦ gcc

- <http://gcc.fyxm.net/releases/gcc-4.0.2/gcc-4.0.2.tar.bz2>;
<http://lfs.osuosl.org/patches/downloads/gcc/gcc-4.0.2-specs-1.patch>;
- ♦ glibc
<http://ftp.osuosl.org/pub/lfs/conglomeration/glibc/glibc-2.3.6.tar.bz2>;
 - ♦ tcl
<http://distro.ibiblio.org/pub/linux/distributions/zenwalk/i486/source/extra/tcl/tcl8.4.12-src.tar.bz2>;
 - ♦ coreutils
<http://ftp.gnu.org/pub/gnu/coreutils/coreutils-5.93.tar.bz2>;
<http://lfs.osuosl.org/patches/downloads/coreutils/coreutils-5.93-uname-1.patch>;
http://lfs.osuosl.org/patches/downloads/coreutils/coreutils-5.93-suppress_uptime_kill_su-1.patch;
<http://www.linuxfromscratch.org/patches/downloads/coreutils/coreutils-5.93-i18n-1.patch>;
 - ♦ findutils
<http://riksun.riken.go.jp/pub/pub/Linux/slackware/slackware-current/source/a/findutils/findutils-4.2.27.tar.bz2>;
 - ♦ gettext
<http://riksun.riken.go.jp/pub/pub/Linux/slackware/slackware-current/source/a/gettext/gettext-0.14.5.tar.bz2>;
 - ♦ ncurses
<http://ftp.lfs-matrix.net/pub/lfs/conglomeration/ncurses/ncurses-5.5.tar.gz>
 (sha1sum atualizado);
<http://www.linuxfromscratch.org/patches/lfs/6.2/ncurses-5.5-fixes-1.patch>;
 - ♦ bash
<http://osiris.tfh-berlin.de/mirror/t2-source/6.0/b/bash-3.1.tar.bz2>;
<http://osiris.tfh-berlin.de/mirror/t2-source/6.0/b/bash-doc-3.1.tar.bz2>;
 - ♦ m4
<http://ftp.gnu.org/gnu/m4/m4-1.4.4.tar.bz2>;
 - ♦ man-pages
<http://rock.inode.at/ROCK-2.1/m/man-pages-2.17.tar.bz2>;
 - ♦ bison
<http://ftp.gnu.org/gnu/bison/bison-2.1.tar.bz2>;
 - ♦ readline
<http://gsmp.tfh-berlin.de/mirror/t2-source/6.0/r/readline-5.1.tar.bz2>;
 - ♦ vim
<http://riksun.riken.go.jp/pub/pub/Linux/slackware/slackware-current/source/xap/vim-gvim/vim-6.4-lang.tar.bz2>;

- ♦ less
<http://ftp.osuosl.org/pub/lfs/lfs-packages/development/less-394.tar.gz>
(sha1sum atualizado);
- ♦ groff
http://ubuntu.interlegis.gov.br/archive/pool/main/g/groff/groff_1.18.1.1-10.diff.gz;
- ♦ groff 1.19
http://slackware.mirrors.pair.com/slackware_source/ap/groff/groff-1.19.2.tar.bz2;
- ♦ iproute2
<http://osiris.tfh-berlin.de/mirror/t2-source/6.0/i/iproute2-051007.tar.bz2>;
- ♦ texinfo
<http://lfs.vlsm.org/lfs/conglomeration/texinfo/texinfo-4.8-multibyte-1.patch>;
- ♦ file
<http://lfs.vlsm.org/lfs/conglomeration/file/file-4.16.tar.gz> (sha1sum atualizado);
- ♦ libtool
<http://ftp.isr.ist.utl.pt/pub/slackware/slackware-10.2/source/d/libtool/libtool-1.5.20.tar.bz2>;
- ♦ diffutils
<http://lfs.vlsm.org/lfs/lfs-packages/development/diffutils-2.8.1-i18n-1.patch>;
- ♦ kbd
<http://www.linuxfromscratch.org/patches/lfs/6.2/kbd-1.12-backspace-1.patch>;
- ♦ e2fsprogs
http://slackware.mirrors.pair.com/slackware_source/a/e2fsprogs/e2fsprogs-1.38.tar.bz2;
- ♦ grep
http://lfs.oss-mirror.org/patches/downloads/grep/grep-2.5.1a-redhat_fixes-1.patch;
- ♦ grub
<http://osiris.tfh-berlin.de/mirror/t2-source/2.1/g/grub-0.97.tar.bz2>;
- ♦ man
<http://osiris.tfh-berlin.de/mirror/t2-source/6.0/m/man-1.6b.tar.bz2>;
- ♦ procps
<http://osiris.tfh-berlin.de/mirror/t2-source/6.0/p/procps-3.2.6.tar.bz2>;
- ♦ psmisc

http://k240.humlak.cz/down_dir/distfiles/psmisc-21.9.tar.gz (sha1sum atualizado);

- ♦ shadow
<http://www.mirrors.wiretapped.net/security/host-security/shadow/old/shadow-4.0.13.tar.bz2>;
http://www.linuxfromscratch.org/patches/downloads/shadow/shadow-4.0.13-configure_fix-1.patch;
- ♦ udev
<http://mirrors.uol.com.br/pub/linux/utils/kernel/hotplug/udev-071.tar.bz2>;
<http://downloads.linuxfromscratch.org/udev-config-5.rules>;
- ♦ openssl
<http://gsmptfh-berlin.de/mirror/t2-source/2.1/o/openssl-0.9.7i.tar.bz2>;
http://www.linuxfromscratch.org/patches/downloads/openssl/openssl-0.9.7i-fix_manpages-1.patch;
- ♦ xfsprogs
http://swoolley.homeip.net/source_cache/xfsprogs-2.6.36.src.tar.gz (sha1sum atualizado);
- ♦ pkgconfig
<http://riksun.riken.go.jp/pub/pub/Linux/slackware/slackware-current/source/d/pkgconfig/pkg-config-0.20.tar.bz2>;
- ♦ curl
<http://dl.ambiweb.de/mirrors/curl.haxx.se/curl-7.15.1.tar.bz2>;
- ♦ subversion
<http://subversion.tigris.org/downloads/subversion-1.2.3.tar.bz2>
- ♦ libxml2
<http://ftp.acc.umu.se/pub/gnome/sources/libxml2/2.6/libxml2-2.6.22.tar.bz2>
 (sha1sum atualizado);
- ♦ libxslt
<http://ftp.gnome.org/pub/gnome/sources/libxslt/1.1/libxslt-1.1.15.tar.bz2>;
- ♦ html_tidy
http://tidy.sourceforge.net/src/old/tidy_src_051026.tgz (sha1sum atualizado);
http://tidy.sourceforge.net/docs/tidy_docs_051020.tgz (sha1sum atualizado);
- ♦ fonts-dejavu
<http://optusnet.dl.sourceforge.net/sourceforge/dejavu/dejavu-ttf-2.0.tar.gz>;
- ♦ libtiff
<http://riksun.riken.go.jp/pub/pub/Linux/slackware/slackware-current/source/l/libtiff/tiff-3.7.4.tar.bz2>;

- ◆ glib2
<http://ftp.gnome.org/pub/GNOME/sources/glib/2.8/glib-2.8.4.tar.bz2>;
- ◆ cairo
<http://riksun.riken.go.jp/pub/pub/Linux/slackware/slackware-current/source/l/cairo/cairo-1.0.2.tar.bz2>;
- ◆ pango
<http://ftp.gnome.org/pub/gnome/sources/pango/1.10/pango-1.10.2.tar.bz2>;
- ◆ cvs
<http://ftp.gnu.org/non-gnu/cvs/source/stable/1.11.21/cvs-1.11.21.tar.bz2>;
<http://www.linuxfromscratch.org/patches/downloads/cvs/cvs-1.11.21-zlib-1.patch>;
- ◆ samba
<http://riksun.riken.go.jp/pub/pub/Linux/slackware/slackware-current/source/n/samba/samba-3.0.20b.tar.bz2>;
- ◆ wireless-tools
http://ftp.sh.cvut.cz/MIRRORS/rock/ROCK-2.1/w/wireless_tools.27.tar.bz2;
- ◆ nfs-utils
<http://www.linuxfromscratch.org/patches/downloads/nfs-utils/nfs-utils-1.0.7-gcc4-1.patch>;
- ◆ rsync
<http://www.aljex.com/bkw/sco/rsync-2.6.6.tar.bz2>;
- ◆ sudo
<http://osiris.tfh-berlin.de/mirror/t2-source/6.0/s/sudo-1.6.8p12.tar.bz2>;
- ◆ pciutils
<http://www.kernel.org/pub/software/utils/pciutils/pciutils-2.2.1.tar.bz2>;
- ◆ dialog
<ftp://invisible-island.net/dialog/dialog-1.0-20060221-.tgz>;
- ◆ nALFS
<http://www.linuxfromscratch.org/alfs/downloads/stable/nALFS-1.2.4.tar.bz2>;
- ◆ rp-pppoe
<http://slackware.osuosl.org/slackware-10.2/source/n/rp-pppoe/rp-pppoe-3.6.tar.gz>;
- ◆ squashfs
<http://ufpr.dl.sourceforge.net/sourceforge/squashfs/squashfs2.2.tar.gz>;
- ◆ linux

http://www.linuxfromscratch.org/patches/downloads/linux/linux-2.6.12.5-utf8_input-2.patch;

- ♦ bin86
<http://osiris.tfh-berlin.de/mirror/t2-source/6.0/b/bin86-0.16.17.tar.bz2>;
- ♦ lilo
<http://cross-lfs.org/files/packages/1.0.0/lilo-22.7.1.src.tar.gz>;
- ♦ syslinux
<http://www.bg.kernel.org/pub/linux/boot/syslinux/syslinux-3.11.tar.bz2>;

Anexo III

Arquivo livecd.py (código-fonte da aplicação Painel do Live-CD):

```
#!/usr/bin/python
#Arquivo: livecd.py
#Autor: Clemente Scaratti clementesc@gmail.com

import pygtk
pygtk.require("2.0")
import gtk, gtk.glade, sys, os

class Aplicacao:

    def __init__(self):
        """
        Metodo Construtor da classe
        """
        #Carrega a interface a partir do arquivo glade
        self.Painel = gtk.glade.XML('livecd.glade')

        #Associa os widgets a variaveis
        self.janela = self.Painel.get_widget('janelaPrincipal')
        self.dproxy = self.Painel.get_widget('diag_proxy')
        self.janPen = self.Painel.get_widget('janelaPen')

        #Conecta Sinais aos Callbacks
        self.Painel.signal_autoconnect(self)

        #Exibe toda interface
        self.janela.show_all()

        #Inicia o loop principal de eventos (GTK MainLoop)
        gtk.main()

    ### Definicao dos callbacks (metodos)
```

```

def sair(self, widget, data):
    """
    Sai do loop principal de eventos.
    Finalizando o programa.
    """
    gtk.main_quit()

def montarPen(self, widget):
    """
    Exibe a janela para selecao do dispositivo
    """
    self.janPen.show()

def montarDesmontar(self, widget):
    """
    Monta ou Desmonta um dispositivo removivel
    """
    if self.rbmdd.name == 'mount_pen':
        os.system('mkdir /media/Pen ; mount /dev/sda1
/media/Pen ; sleep 2 &')
    elif self.rbmdd.name == 'umount_pen':
        os.system('umount /dev/sda1 && rm -rf /media/Pen ;
sleep 2 &')
    elif self.rbmdd.name == 'mount_floppy':
        os.system('mkdir /media/Disquete ; mount /dev/fd0
/media/Disquete ; sleep 2 &')
    elif self.rbmdd.name == 'umount_floppy':
        os.system('umount /dev/fd0 && rm -rf /media/Disquete
; sleep 2 &')

def setaDispositivo(self, widget):
    """
    Utilizado para montar ou desmontar dispositivos
    Armazena na variavel rbmdd a acao que o metodo
    montarDesmontar deve executar
    """
    self.rbmdd = self.Painel.get_widget(widget.name)

def exportarProxy(self, widget):
    """
    Acrescenta uma linha de codigo no arquivo /etc/bashrc.
    Quando executar uma nova instancia do shell o proxy
    estara configurado
    """
    self.servidor = self.Painel.get_widget('ed_servidor')
    self.porta = self.Painel.get_widget('ed_porta')
    self.usuario = self.Painel.get_widget('ed_usuario')
    self.senha = self.Painel.get_widget('ed_senha')

    comando = "echo \"http_proxy=http://"
    comando+= self.usuario.get_text()

```

```

        comando+= ":"
        comando+= self.senha.get_text()
        comando+= "@"
        comando+= self.servidor.get_text()
        comando+= ":"
        comando+= self.porta.get_text()
        comando+= "\" >> /etc/bashrc"

        #Mostra a janela de informacao do proxy
        self.dproxy.show()
        os.system(comando)

def fecharDialogoProxy(self,widget):
    """
    Esconde a janela de informacao do proxy
    Sera executado quando o botao fechar for clicado
    """
    self.dproxy.hide()

def fecharJanPen(self,widget):
    """
    Esconde a janela de selecao de dispositivos
    Sera executado ao clique no botao fechar
    """
    self.janPen.hide()

def abreDbdesigner(self, widget):
    """
    Executa uma chamada ao sistema operacional para
    executar o programa DBDesigner
    """
    os.system("/bin/sh -c /opt/DBDesigner4/startdbd &")

def abrePhpMyAdmin(self, widget):
    """
    Executa uma chamada ao sistema operacional para
    executar o programa firefox
    e abrir a pagina http://localhost/phpmyadmin
    """
    os.system("/usr/bin/firefox http://localhost/phpmyadmin
&")

def abreLink(self, widget):
    """
    Executa uma chamada ao sistema operacional para
    executar o programa firefox
    e abrir o URL recebido por parametro
    """
    os.system("/usr/bin/firefox "+widget.get_active_text()+"
&")

```



```

def abreAplicacao(self, widget):
    """
    Executa uma chamada ao sistema operacional para executar
    o software detectado pelo nome do botao clicado
    """
    nome = widget.get_name()
    nome = nome.split("bt_")
    if nome[1] == 'net-setup' :
        os.system("/usr/bin/terminal -x net-setup &")
    elif nome[1] == 'mysql' :
        os.system("/usr/bin/terminal -x mysql -u root &")
    else:
        os.system("/bin/sh -c "+nome[1]+" &")

def alterarServico(self, widget):
    """
    Inicia ou para um processo. O nome do processo e a acao
    eh detectada pelo nome do botao clicado
    """
    nome = widget.get_name()
    nome = nome.split("_")
    os.system("sudo /etc/rc.d/init.d/"+nome[2]+" "+nome[1])

#Inicia a aplicacao
if __name__ == "__main__":
    Aplicacao()

```