

# Desenvolvimento de uma Distribuição Live-CD para Uso Acadêmico na Fatec Senac Pelotas

Álvaro Vitaca Amaro

Faculdade de Tecnologia Senac Pelotas (FATEC)  
Faculdade de Análise e Desenvolvimento de Sistemas  
Rua Gonçalves Chaves, 602 – Centro  
alvitaca@gmail.com

**Resumo.** *Este artigo tem como objetivo demonstrar o desenvolvimento da distribuição Linux Senix baseada em um sistema live-cd customizado para o uso acadêmico no curso de análise e desenvolvimento de sistemas da Faculdade de Tecnologia Senac Pelotas com base no projeto Linux from Scratch, possibilitando aos alunos e professores uma série de ferramentas para prestar suporte ao estudo das unidades curriculares.*

Palavras-chave: Linux; Customização; Software livre;

**Abstract.** *This article has as objective to demonstrate the development of the distribution Linux Senix based on a customized live-cd system for the academic use in the course of analysis and development of systems of the College of Technology Senac Pelotas with basis on the project Linux from Scratch, making possible to students and professors a series of tools to give support to the study of the curricular units.*

Keywords: Linux; Customization; Free Software;

## 1. Introdução

Os sistemas operacionais baseados em Linux estão cada vez mais presentes no cotidiano dos profissionais da área de TI (Tecnologia da Informação), pois apresentam um ótimo custo/benefício com uma capacidade de operação capaz de disponibilizar soluções que podem ser utilizadas tanto em organizações privadas quanto públicas.

Devido a inúmera quantidade de softwares existentes o Linux atinge praticamente todos os setores do mercado com as mais variadas demandas por soluções em tecnologia da informação.

No ambiente corporativo, as empresas que adotam o Linux contam com o fato de poderem alterar o sistema para que ele se adapte às necessidades da companhia. Nas instituições de ensino, tanto de nível médio quanto como de nível superior, a utilização do Linux foi pioneira devido a menor resistência as novidades e a utilização didática do sistema operacional e das aplicações de código-fonte aberto.

Porém, mesmo com essas vantagens, o crescimento do uso do Linux ainda é pequeno entre os usuários domésticos, pois o vê como uma alternativa, não como uma solução definitiva. Segundo [INFO09], o número de usuários domésticos de Linux tem

crescido 10% ao ano. Já os usuários corporativos, corresponde a apenas 1% de usuários Linux.

Para mudar este quadro, a familiarização do Linux no ambiente de uma faculdade de tecnologia, voltada ao mercado de trabalho, poderá fazer com que a adoção do Linux aumente no ambiente corporativo. O desenvolvimento de uma distribuição customizada para o ambiente acadêmico poderá ser de grande auxílio para este crescimento. Além disto, pode proporcionar ao aluno egresso e futuro profissional uma alternativa para a realização de suas tarefas no ambiente de trabalho devido a aquisição do conhecimento sobre o Linux.

É disponibilizado por meio de distribuições. Uma distribuição Linux corresponde ao agrupamento de aplicações e ferramentas e tem o objetivo de facilitar a utilização do sistema operacional Linux. O conjunto de aplicações e ferramentas é extremamente diverso e pode tornar confusa ao usuário leigo ou iniciante. De acordo com esta constatação, a customização de uma distribuição para as funcionalidades necessárias do usuário faz com que a sua adoção torne-se mais simples e rápida. O Senix que é uma distribuição Linux baseada no projeto Linux from Scratch, roda diretamente de uma mídia removível seja ela CD-Rom, DVD ou pendrive, o chamado Live-CD e foi criada para auxiliar a comunidade acadêmica da Faculdade Senac Pelotas nas aulas práticas e nos trabalhos acadêmicos.

## 2. O Linux

É um sistema operacional do tipo Unix, multiusuário, multitarefa e multiprocessado, de código-fonte aberto e disponível para diversas arquiteturas de computadores. O Linux segue um conjunto de normas chamadas POSIX (*Portable Operating System Interface*), cujo o objetivo é assegurar a portabilidade do código-fonte de um programa desenvolvido para um sistema que adere a estas normas para outro também aderente.

O núcleo do sistema operacional chamado de *Kernel*, que possui as funcionalidades de gerenciamento do hardware e das aplicações, corresponde ao que é chamado de Linux. Este *Kernel* foi inicialmente criado por Linus Torvalds (Figura 1), com o intuito de aprender sobre o funcionamento de sistemas operacionais [Linux Foundation 2008]. Por um meio colaborativo, este projeto acabou sendo adotado por milhares de desenvolvedores ao redor do mundo. A partir disto, foram desenvolvidas diversas ferramentas e aplicações que tornaram o Linux possível de ser utilizado por usuários não desenvolvedores. Desta forma, com o auxílio de toda a comunidade do *software* livre, o Linux tem sido desenvolvido e aprimorado constantemente. Hoje, o sistema operacional Linux possui, praticamente, as mesmas capacidades e ferramentas disponibilizadas pelos sistemas operacionais comerciais.



Figura 1 - Linus Torvalds, criador do Kernel Linux.

### 3. Projeto GNU

Este projeto foi iniciado em 1984 por Richard Stallman com o objetivo de criar um sistema operacional totalmente livre compatível com o UNIX, porém utilizando ferramentas próprias que seriam desenvolvidas especialmente para ele [Gnu 2009]. Outro detalhe que chamou a atenção na época, foi o fato de que qualquer pessoa poderia ter o direito de utilizar, modificar ou redistribuir o sistema, desde que disponibilizasse estas modificações para toda a comunidade.

Ao longo do tempo o projeto desenvolveu todas as peças fundamentais para um sistema operacional, tais como compiladores em C, editores de texto, interpretadores de comandos, ferramentas para compressão de dados entre outras. Porém apesar de todas as ferramentas estarem disponíveis dentro do prazo previsto, o *Kernel* do sistema operacional ainda não havia sido completamente desenvolvido.

Eis que durante essa época já existia um *Kernel* com o desenvolvimento consideravelmente avançado, capaz de utilizar todas as ferramentas desenvolvidas pelo projeto GNU com apenas algumas pequenas modificações. Este *Kernel* era o Linux.

Por comodidade pessoas utilizam apenas o termo Linux para designar o sistema operacional completo, porém o núcleo apenas não seria nada sem as ferramentas desenvolvidas pelo projeto GNU, sendo o termo correto para denominar o sistema operacional GNU/Linux.

### 4. Aplicações no GNU/Linux

Atualmente o GNU/Linux possui uma riqueza incomparável de aplicativos para todas funções, chegando ao extremo de existirem inúmeros aplicativos para as mesmas atividades.

Grande parte dos aplicativos disponíveis para GNU/Linux são de código aberto permitindo alterações conforme a necessidades usuários, e em alguns casos podem ser oferecidos pelas distribuições através de repositórios dedicados, pacotes pré-compilados com o sistema de empacotamento padrão ou pela disponibilidade dos fontes na página do *software*.

Como exemplo de aplicativos disponíveis para o sistema tem-se o Apache [Apache 2009] que é um servidor *web* livre, o Gimp [Gimp 2009] que é um aplicativo para manipulação de imagens e o OpenOffice.org [OpenOffice.org 2009] que é uma suíte completa de aplicativos para escritório e muitos outros.

### 5. Distribuições GNU/Linux

Uma distribuição GNU/Linux pode ser considerada como sendo o conjunto constituído pelo *Kernel*, ferramentas de sistema e aplicativos, sendo estes últimos escolhidos pelos desenvolvedores conforme a finalidade da distribuição e reunidos em uma imagem de CD ou DVD. Todas as distribuições possuem dois pontos em comum: o *Kernel* e as ferramentas de sistema.

O *Kernel* encontrado em uma distribuição é frequentemente diferente da versão encontrada em outra distribuição [Danesh 1999]. Essa regra é válida também para as ferramentas, aplicativos, utilitários, *drivers*, programas de instalação e de atualização.

Existem aproximadamente 500 distribuições no mundo todo, segundo [DistroWatch 2009], do qual menos da metade tem um desenvolvimento ativo (Figura 2). Este número demonstra que as principais distribuições podem falhar ao tentar

abranger alguma finalidade específica, tal como Edição Gráfica, Segurança ou Educação, contribuindo para o surgimento de outros projetos derivados voltados para estas finalidades. Um exemplo disto é a distribuição *BackTrack* voltada para a área de segurança [Backtrack 2009].

A grande vantagem em se utilizar uma distribuição existente e customizá-la está no ganho de tempo ao se utilizar projetos desenvolvidos por outras pessoas agregando novas funcionalidades com o objetivo de expandir sua atuação.

As distribuições conhecidas como Live-CD não necessitam ser instaladas, podendo ser executadas diretamente de um CD ou DVD. Um exemplo deste tipo distribuição é o Kurumin[Morimoto 2008].



Figura 2 – Logotipos de Algumas Distribuições GNU/Linux com Desenvolvimento Ativo.

## 6. Principais Distribuições

Muitas distribuições GNU/Linux estão disponíveis para download, porém poucas fazem parte do seleto grupo das distribuições principais, que recebem esta designação pela sólida comunidade de desenvolvedores e desenvolvimento ininterrupto do projeto. Entre estas estão o Slackware, Redhat, Debita.

Juntas estas quatro distribuições englobam as características que trataram de moldar os sistemas GNU/Linux tais como são hoje, pois cada uma deu origem a um conjunto de derivações ou novas soluções que acabaram por favorecer toda comunidade de *software* livre.

### 6.1 Redhat

Foi uma das primeiras distribuições do mundo, é muito popular nos EUA, tem o objetivo tornar o GNU/Linux mais acessível através da automatização do sistema e configurações. Uma de suas maiores contribuições foi o sistema de empacotamento rpm [Rpm 2009], sendo a primeira a utilizar o conceito de pacotes, onde cada um fornece um pedaço diferente do *software*, configurado, completamente testado e pronto para rodar.

Atualmente está voltado ao mercado de servidores corporativos vendendo suporte e soluções pagas. Restando apenas sua ramificação, o Projeto Fedora [Fedora Project 2009], que mantém-se livre e voltada ao grande público. Entre suas derivadas pode-se citar o Mandriva, Suse, Fedora e CentOS.

Na Figura 3, uma tela da distribuição Red Hat versão 9.0 rodando o ambiente *desktop* KDE.

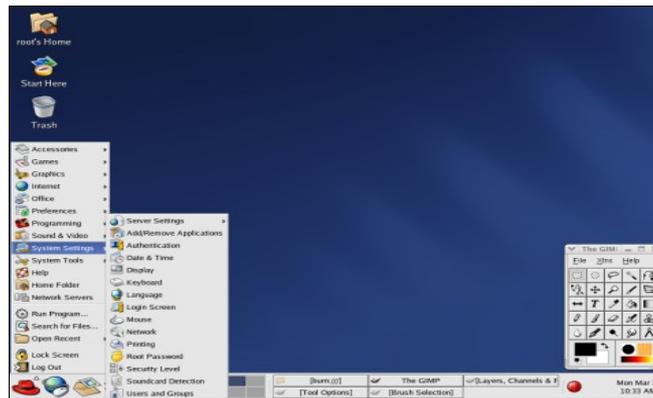


Figura 3 – Sistema Red Hat Linux 9

## 6.2 Slackware

É uma das mais famosas e antigas distribuições do mundo GNU/Linux, sua meta é permanecer o mais fiel possível ao Unix com a finalidade de maximizar a estabilidade do sistema.

Embora seja considerada por muitos uma distribuição difícil de usar, voltada principalmente para usuários experientes, possui um sistema de gerenciamento de pacotes simples, assim como sua interface de instalação, que é uma das poucas que continua em modo-texto.

Entre suas derivações encontra-se o Slax [Slax 2009] que é um Live-CD e o Vector Linux [Vector Linux 2009] cujo o foco são os computadores antigos.

Na Figura 4, uma tela da distribuição Slackware versão 12.2 rodando o gerenciador de janelas brasileiro *Window Maker* [Window Maker 2009].



Figura 4 – Slackware Linux

## 6.3 Debian

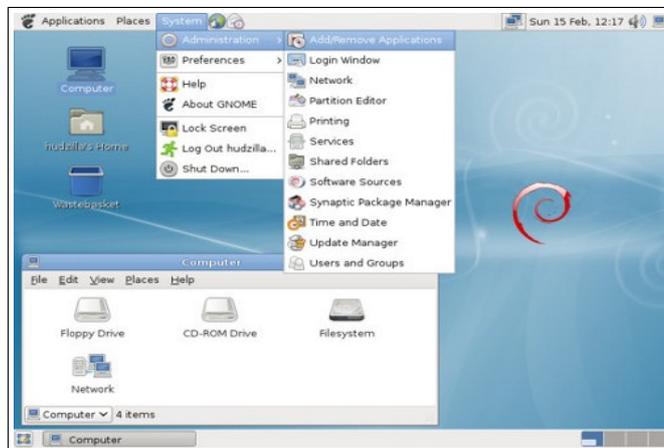
Popular no mundo inteiro, disponibiliza em seus repositórios programas de código-fonte aberto, desenvolvido por voluntários espalhados pelo mundo sem qualquer fim lucrativo.

Esta distribuição deu origem a quase metade das distribuições em uso atualmente [Morimoto 2005], sendo o Kurumin, Knoppix, Mint e Ubuntu, exemplos disto.

Recentemente a utilização do Debian era restrita a usuários experientes e sua

utilização focada em servidores, mas com o passar do tempo o projeto sofreu uma forte influência de suas distribuições derivadas com foco na facilidade de uso, tornando o sistema acessível a todo tipo de usuário e possibilitando seu uso em estações de trabalho ou em casa.

Na Figura 5, uma tela da distribuição Debian versão 5.0.2 rodando o ambiente *desktop* Gnome.



**Figura 5 - Debian Linux**

## 7. Distribuições em formato *Live-CD*

São distribuições que rodam diretamente de uma mídia de CD/DVD sem a necessidade de serem obrigatoriamente instaladas em um disco rígido. Deve-se apenas configurar para inicializar a partir do *drive* de CD ou outra mídia removível. Depois de carregado o sistema pode obter acesso a todos os dispositivos e executar aplicações.

Por meio do Live-CD o usuário pode testar um determinado sistema GNU/Linux sem desinstalar o sistema padrão ou utilizá-lo como ferramenta de auxílio a problemas pela possibilidade de acessar as partições de disco rígido.

Na Tabela 1, estão listadas as vantagens e desvantagens que compreendem todas as distribuições Live-CDs existentes.

**Tabela 1. Vantagens e desvantagens no uso de uma distribuição GNU/Linux Live-CD.**

Vantagens	Desvantagens
Como se trata de um sistema de base Linux, não necessita antivírus e <i>anti-spywares</i> para evitar a contaminação do sistema, pois é imune a vírus devido a diferente arquitetura do sistema.	O sistema demora um pouco para iniciar porque a leitura do CD é mais lenta que a leitura do disco.
A portabilidade, permite a qualquer usuário rodá-lo em qualquer computador apenas colocando-o no <i>drive</i> após configurar o <i>boot</i> através da mídia.	Todas alterações efetuadas no ambiente do Live CD serão perdidas quando reiniciar o computador.
Não é necessário ter um disco	Exige considerável poder

rígido instalado na máquina para rodá-lo, bastar ter um <i>drive</i> de CD/DVD. Caso o sistema ou disco esteja danificado, pode-se usar o Live-CD para recuperá-lo.	de processamento da cpu e adequada quantidade de memória, já que as informações terão que ser descompactadas antes de serem executadas.
---	---

### 7.1 Live-USB

Atualmente mesmas equipes envolvidas nos projetos das distribuições *Live-CDs* estão tratando de fornecer a documentação necessária para que o sistema operacional rode diretamente de um *pendrive*. Essas distribuições são conhecidas como *Live-USB*, que como os os *Live-CDs* podem ser utilizadas para testar um sistema operacional ou como ferramenta de auxiliar na manutenção de um computador.

Uma vantagem deste tipo de distribuição é possibilidade de customizar o sistema, podendo instalar aplicativos no *pendrive*, opção esta não permitida nos *Live-CDs*[Tio Geek 2008].

### 7.2 Processo de Remasterização

A remasterização, de Live-CDs permite criar um sistema operacional voltado a finalidades específicas. Existem inúmeras ferramentas que ajudam na automatização desse processo, tornando a remasterização acessível a qualquer um, independente de sua experiência. Como exemplos de distribuições Live-CD pode-se citar o Knoppix, Kurumin e o Ubuntu.

## 8. Métodos para criação de distribuições

Existem muitos modos de se agregar novas funcionalidades a uma distribuição com o objetivo de maximizar a integração entre o usuário e o sistema. Uma maneira muito comum é a utilização de ferramentas automatizadas com interface gráfica que tratam de guiar o usuário passo-a-passo através de opções disponíveis. Porém não permitem um conhecimento integral do sistema que está sendo desenvolvido, agindo mais como agentes facilitadores, ao contrário da metodologia *Linux From Scratch* que é extremamente didática e exige comprometimento com o que está sendo desenvolvido. Entre estes tipos de ferramenta pode-se citar o MKDistro e o RemasterSys.

O MKDistro é conjunto de *scripts* que podem ser empregados na construção de distribuições, agindo como uma interface (Figura 6), que auxilia o usuário a moldar o sistema conforme suas necessidades e gostos por um processo totalmente automatizado [DreamLinux 2006]. Conforme se vai utilizando a ferramenta vão sendo apresentadas telas com as opções e uma breve explicação para cada uma, o que torna seu uso acessível tanto a usuários experientes como iniciantes.

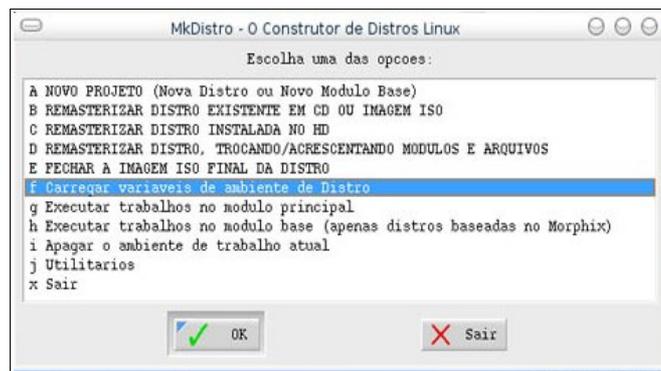


Figura 6 – Ferramenta MKDistro.

Por outro lado, o RemasterSys (Figura 7) é um *software* que permite criar um Live-CD personalizado ou *backup* do Debian e derivados. Permite a inclusão de programas do usuário, papéis de parede e configurações especiais criados após a instalação do sistema no computador no CD [RemasterSys 2008].

Seu objetivo inicial foi criar facilmente um cópia de segurança ou uma cópia de um sistema Debian ou derivado que fosse possível distribuir [Brijeski 2007]. Sendo sua principal inspiração o *script* Mklivecd utilizado no Mandriva e o *script* RemasterMe [RemasterSys 2008] para PCLinuxOS. Como a portabilidade desses *scripts* era muito complicada ou limitada, o RemasterSys foi desenvolvido inteiramente do zero agregando uma gama muito maior de funcionalidades.

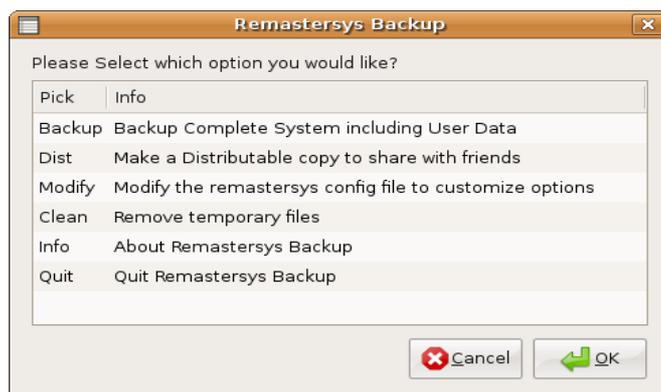


Figura 7 – A Ferramenta RemasterSys e seu conjunto de opções.

## 9. Desenvolvimento da Distribuição com o *Linux From Scratch* (LFS)

É uma metodologia desenvolvida por Gerard Beekmans que tem como objetivo guiar o usuário na construção e customização de uma distribuição, explicando aspectos funcionais e técnicos passo a passo.

Como resultado obtêm-se um sistema rápido e compacto, sem adição de pacotes dispensáveis, permitindo uma grande economia de espaço e tempo. Para tal é utilizado o próprio Live-CD que contém uma documentação básica em HTML do projeto e ferramentas utilizadas na compilação.

O projeto possui outras documentações, disponibilizadas no site em formato PDF (Figura 8). Estas outras documentações visam auxiliar, de modo geral, o desenvolvedor na compilação e utilização de ferramentas pertencentes a outros projetos livres com a ideia de expandir o foco operacional de um sistema construído e customizado a partir do LFS [Beekmans 2009].

De posse de todo este conteúdo fica fácil para o usuário criar suas próprias

ferramentas de trabalho, como *scripts*, que permitem a automatização das etapas básicas e minimizem a ocorrência de erros provenientes da digitação.

A maior vantagem deste projeto perante outros, reside na extrema flexibilidade das opções disponíveis ao desenvolvedor, permitindo-o modelar o sistema de acordo com as necessidades de seu objetivo com acesso aos fontes [Beekmans 1998].

Durante a etapa de desenvolvimento foram utilizadas referências de dois outros projetos integrados conhecidos como *Beyond Linux From Scratch* (BLFS) e *Cross Beyond Linux From Scratch* (CBLFS). Ambos tratam de complementar o LFS, provendo grande e vasta fonte de instrução para instalação e configuração de pacotes homologados. Estes pacotes que se encontram no topo da base do sistema LFS, de forma que ao final do processo pode-se obter um sistema para qualquer finalidade.

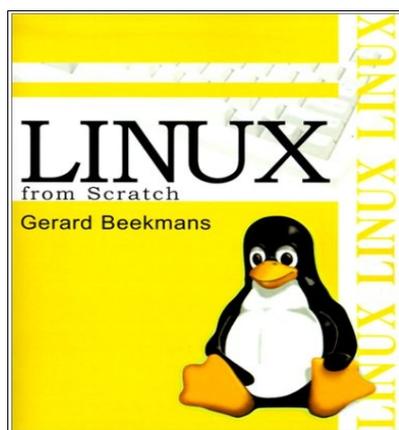


Figura 8 – Capa do Ebook Linux From Scratch.

## 9.1 Informações Gerais Sobre o Senix

Existem inúmeros ambientes *desktop* para o GNU/Linux, entre os mais utilizados estão o KDE, o Gnome e o XFCE. Os dois primeiros apesar de muito consagrados no universo Linux e exigem uma máquina com poder de processamento e quantidade de memória considerável para que todos os recursos oferecidos por eles possam ser utilizados sem nenhuma restrição. Por outro lado o XFCE, é um ambiente cujo o maior objetivo é manter-se leve buscando preservar a capacidade de processamento do computador consumindo a menor quantidade possível de memória RAM [Xfce 2009].

O XFCE exige poucos recursos computacionais sendo essa é a razão pela qual ele está integrado ao Senix Linux, pois desta forma alunos e professores poderão rodar o ambiente *desktop* completo em qualquer máquina com uma quantidade de memória maior ou equivalente a 192 MB [Xfce 209]. Esta quantidade de memória RAM ou maior é encontrada normalmente no sistemas computacionais atuais.

O Senix possui suporte de leitura e escrita a vários sistemas de arquivos, de diversos sistemas operacionais. Ele pode ler e escrever nas partições formatadas em FAT, NTFS, JFS, XFS, HPFS, EXT2, EXT3 e ReiserFS. Por isto, os Live-CDs GNU/Linux são muito utilizados na manutenção e recuperação de outros sistemas operacionais.

A base para inicialização do Live-CD reside no *Initrdramfs* [Linux Devices 2006], uma espécie de *Ramdisk*, que fica no encargo de fornecer um sistema de arquivos inicial bastante básico ao *Kernel*. Este sistema de arquivos será utilizado para auto-inicialização do dispositivo, localizando e montando o sistema de arquivos do dispositivo de armazenamento. O *Kernel* é setado para utilizar o *Initrdramfs* como

diretório raiz, simulando o acesso ao disco rígido aonde ficará armazenado alguns arquivos de configuração.

Já na próxima etapa serão executados os aplicativos de detecção de hardware que tem como finalidade ativar e configurar os periféricos existentes no computador, quando possível, sem qualquer necessidade de intervenção do usuário.

O Senix utiliza um módulo do *Kernel* que atua como sistema unificador de arquivos chamado ZisoFS [Freshmeat 2001]. Este módulo é o responsável pela união do conteúdo atualizado de múltiplos diretórios mantendo seu conteúdo original separado. Outra característica é que esse módulo permite a compressão de uma grande quantidade de dados e mistura de arquivos com e sem permissão de escrita, como a possibilidade de edição de qualquer parte, tratando de eliminar arquivos duplicados [All Experts 2009].

Um módulo do *Kernel* que possui as mesmas características do ZisoFS e é utilizado também em distribuições Live-CDs é o SquashFs, CramFs e o Cloop [CE Linux 2008].

## 9.2 Processo de Desenvolvimento do Senix

O principal objetivo do Senix é criar um sistema Linux Live-CD customizado para o uso acadêmico no curso de análise e desenvolvimento de sistemas, onde alunos e professores, tenham ao seu dispor uma série de ferramentas com a finalidade de prestar suporte ao estudo das disciplinas.

O processo de construção do Senix baseia-se na remasterização. Assim o desenvolvedor de posse do CD ou imagem ISO da distribuição de origem, pode alterar e personalizar os componentes de forma a obter uma distribuição totalmente nova com características específicas.

Para o desenvolvimento do protótipo foram utilizados os seguintes softwares de código-fonte aberto:

- Distribuição Debian GNU/Linux 5.0.2 *Lenny* – *Kernel* 2.6.26-2-686;
- Distribuição LFS GNU/Linux Live-CD 6.3 r2168 Oficial;
- VirtualBox 3.0.10 [VirtualBox 2009];
- Suíte para gravação *GnomeBaker* 0.6.4 [GnomeBaker 2009];
- Possuir cerca de 4 GB de espaço livre em disco.
- Os pacotes Wodim egensoimage, [CdrKit 2008] que contém todos os utilitários que permitem criar, manipular e gravar um CD ou imagem ISO.

O primeiro passo a ser realizado é montar a imagem *ISO* do *Live-CD* do Projeto *Linux From Scratch* em um ponto de montagem, para isso é criado o diretório *lfs* dentro do diretório de montagem */mnt* e para montar-se a imagem *ISO* com o auxílio do *loop device* dentro deste diretório (Figura 9)

```
mkdir /mnt/lfs
mount -o loop /caminho/imagem/lfslivecd-x86-6.3-r2160.iso /mnt/lfs
```

Figura 9 – Montando a imagem isso.

Deste modo pode-se ter acesso a todo o conteúdo da imagem *ISO* sem que seja necessário gravá-la em um *CD*. Logo após, deve-se criar um diretório de trabalho

chamado *livecd* dentro do diretório de montagem, para onde é copiado o arquivo *root.ext2* (Figura 10), que é uma imagem do sistema de arquivos encontrada dentro do diretório */boot* do Live-CD juntamente o arquivo *Initramfs*.

```
mkdir /mnt/livecd
cp -v /mnt/lfs/boot/root.ext2 /mnt/livecd/
```

**Figura 10 – Copiando o sistema de arquivos do Live-CD para um diretório.**

Cria-se um diretório chamado *root* dentro do diretório de trabalho para montar o sistema de arquivos da imagem (Figura 11), que permite a manipulação do conteúdo, utilizando o *loop* novamente.

```
mkdir /mnt/livecd/root
mount -o loop /mnt/livecd/root.ext2 /mnt/livecd/root
```

**Figura 11 – Montando o sistema de arquivos do Live-CD.**

Cria-se também um diretório temporário chamado *build*, que ficará dentro do diretório de trabalho e outro de mesmo nome no sistema de arquivos da imagem montada, vinculando os dois através da montagem com a opção *-bind* (Figura 12). Isso permite que os pacotes dos programas a serem instalados sejam acessados em dois lugares distintos, ou seja, no sistema anfitrião que os enviará e no sistema de arquivos da imagem onde serão devidamente configurados e instalados.

```
mkdir /mnt/livecd/build /mnt/livecd/root/build
mount --bind /mnt/livecd/build /mnt/livecd/root/build
```

**Figura 12 – Vinculando os diretórios com uma montagem Bind.**

É necessário montar os sistemas de arquivos e os diretórios virtuais do sistema anfitrião dentro do ponto de montagem da imagem. São necessários para o uso do *chroot* que permite que alternar entre o diretório raiz do sistema hospedeiro e o raiz do sistema de arquivos da imagem (Figura 13).

```
mount -t proc proc /mnt/livecd/root/proc
mount -t sysfs sysfs /mnt/livecd/root/sys
mount -t devpts devpts /mnt/livecd/root/dev/pts
mount -t tmpfs tmpfs /mnt/livecd/root/dev/shm
```

**Figura 13 – Montando os sistemas de arquivos e diretórios virtuais.**

Feito isso o próximo passo é utilizar o *chroot* para acessar e manipular livremente o conteúdo do sistema de arquivos da imagem (Figura 14).

```
chroot /mnt/livecd/root
```

**Figura 14 – Utilizando o Chroot para ter acesso ao diretório raiz do sistema de arquivos do Live-CD.**

### 9.2.1 Finalizando o Processo e Criando a Imagem ISO.

Para criar uma imagem *ISO*, é necessário baixar os fontes do *Initramfs*. com o *Subversion* (Figura15), disponibilizado pelo projeto *Linux From Scratch* do seguinte modo:

```
svn co svn://svn.linuxfromscratch.org/livecd/trunk/packages/initramfs
```

**Figura 15 – Baixando os fontes do Initramfs. com Subversion.**

Após remove-se as linhas *include \$(ROOT)/scripts/functions* e *cp initramfs\_data.cpio.gz /boot/isolinux/initramfs\_data.cpio.gz* do arquivo *Makefile* e gera-se o arquivo *Initramfs*. para o Live-CD, sendo posteriormente movida para o diretório temporário *build* (Figura16).

```
cd initramfs/  
make compile-stage2 VERSION="nome-da-distribuicao" LINKER=ld-linux.so.2  
create-initramfs nome-da-distribuicao 2.6.22.5 initramfs_data.cpio.gz  
cp -v initramfs_data.cpio.gz /build
```

**Figura 16 – Compilando o Initramfs. e customizando-o.**

Após é necessário sair do ambiente montado com o auxílio do *chroot* e desmontar os sistemas de arquivos e os diretórios virtuais do sistema anfitrião (Figura 17) no ponto de montagem do sistema de arquivos do Live-CD, antes necessários:

```
umount /mnt/livecd/root/dev/shm  
umount /mnt/livecd/root/dev/pts  
umount /mnt/livecd/root/sys  
umount /mnt/livecd/root/proc  
umount /mnt/livecd/root/build  
rmdir /mnt/livecd/root/build/
```

**Figura 17 – Desmontando os sistemas de arquivos e diretórios virtuais.**

Limpa-se *bash\_history* que se encontra no diretório do *root* para que os comandos digitados durante a etapa de desenvolvimento não sejam visualizados pelos usuários que utilizarão o sistema (Figura 18), evitando eventuais problemas.

```
rm /mnt/livecd/root/root/.bash_history
```

**Figura 18 – Removendo o histórico de comandos digitados.**

Como durante todo o processo são criados e removidos arquivos várias vezes, os setores anteriormente ocupados por estes começam a conter resquícios de dados não apagados (Figura 19). Esses setores são inúteis e não permitem uma boa compactação do arquivo *root.ext2* para criação da imagem *ISO*. Basta então apagá-los da seguinte maneira:

```
dd if=/dev/zero of=$WORK/root/zeros
```

**Figura 19 – Apagando resquícios de arquivos dos setores.**

Este comando apresenta uma mensagem que o disco estava cheio, isto não é um erro. Mas é necessário remover o arquivo, pois ele preenche todo espaço livre do disco (Figura 20).

```
rm /mnt/lfs/livecd/root/zeros
```

**Figura 20 – Removendo o arquivo Zeroes.**

Desmonta-se o ponto de montagem do sistema de arquivos do Live-CD e utiliza-se o comando *sync* (Figura 21) para forçar a atualização dos dados e prevenção de erros no sistema de arquivos dessa imagem.

```
umount /mnt/livecd/root  
sync
```

**Figura 21 – Desmontando o sistema de arquivos e prevenindo erros.**

Cria-se o diretório que servirá de estrutura ao *Live-CD*, para onde posteriormente será copiado a estrutura de diretórios do *Live-CD* original e os arquivos contidos neste (Figura 22).

```
mkdir /mnt/livecd/iso  
cp -r /mnt/lfs/* /mnt/livecd/iso
```

**Figura 22 – Criando diretória de estrutura do Live-CD customizado.**

Substitui-se o arquivo *Initramfs*. original copiado com a estrutura de diretórios pelo que foi criado manualmente no processo anterior (Figura 23).

```
cp -fv /mnt/livecd/build/initramfs_data.cpio.gz /mnt/livecd/iso/boot/isolinux/
```

**Figura 23 – Copiando o Initramfs. customizado para o diretório de estrutura.**

Compacta-se o arquivo *root.ext2* (Figura 24), visando diminuir seu tamanho para que possa ser abrigado por um *CD* ou *DVD*.

```
mkzftree -F /mnt/livecd/root.ext2 /mnt/livecd/iso/root.ext2
```

**Figura 24 – Compactando o sistema de arquivos do Live-CD com a ferramenta Mkzftree.**

Recria-se novamente a imagem *ISO* com o auxílio da ferramenta *Mkisofs* (Figura 25) contida no pacote *Genisoimage*.

```
cd /mnt/livecd/iso
mkisofs -z -R -l --allow-leading-dots -D -o /mnt/livecd/lfslivecd-nome-da-distribuicao.iso \
-b boot/isolinux/isolinux.bin -c boot/boot.cat -no-emul-boot -boot-load-size 4 \
-boot-info-table -V "lfslivecd-nome-da-distribuicao"
```

**Figura 25 – Criando uma nova imagem iso da distribuição com a ferramenta Mkisofs.**

### 9.3 Pacotes

A seleção dos pacotes que integram a distribuição tem como base aplicativos iguais ou equivalentes utilizados nas aulas do curso de Análise e Desenvolvimento de Sistemas.

Foi também realizada uma pesquisa via email onde uma lista das ferramentas que seriam integradas a distribuição (Tabela 2) foram apresentadas aos professores que compõem o corpo docente do curso de Análise e Desenvolvimento de Sistemas e os mesmos tiveram a oportunidade de apresentar seu parecer ou dar suas sugestões .

**Tabela 2 - Relação dos Principais Pacotes do Senix.**

Pacote	Categoria	Unidade Curricular
Adobe Reader 9 e BrOffice.org 3	Escritório	Todas
Audacious, Mplayer/Mencoder e Codecs	Multimídia	Genérico
Bluefish, Geany	Web Design	Web Design
DBDesigner 4, MySQL e PhpMyAdmin	Banco de Dados	Banco de Dados
Dia 0.96.1 e Jude Community 5	UML	Modelagem de Software
Netbeans	IDE	Programação Estruturada
Firefox 3.5	Navegador de Internet	Todas
Gimp	Gráficos	Uso genérico
Java JDK, Python e PHP 5	API de Programação	Programação
Net-Tools, Nmap,	Rede	Redes de

WireShark, BindUtilities e Packet Tracer 5		Computadores
Wine	Emulador	Sistemas Operacionais
Xarchiver	Compactador de Arquivos	Uso genérico

No Senix os programas podem ser instalados através da compilação do código-fonte de um programa ou a instalação de um pacote pré-compilado com o auxílio da ferramenta *Pkgtool*.

## 9.4 Compilação

É um dos principais diferenciais entre os softwares de código-fonte livre e os softwares proprietários, pois basta que o usuário obtenha o código-fonte do programa e compile-o, podendo adaptar o programa a uma máquina específica ou vários tipos de máquinas.

Este processo resume-se a três linhas, onde o *script configure* define os parâmetros da instalação do programa, tais como o diretório alvo de instalação (Figura 12), além de permitir a detecção de eventuais problemas de dependências.

O *make* é o responsável pela compilação, através das instruções contidas no *Makefile* gerado pelo *script* de configuração e o *make install* instalará o programa compilado (Figura 26).

```

root [ /build/Pacotes/Monitor/libxml2-2.7.6 ]# ./configure --prefix=/usr
checking build system type... i686-pc-linux-gnu
checking host system type... i686-pc-linux-gnu
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking for gcc... gcc
checking for C compiler default output file name... a.out
checking whether the C compiler works... yes
checking whether we are cross compiling... no
checking for suffix of executables...
root [ /build/Pacotes/Monitor/libxml2-2.7.6 ]# make
make all-recursive
make[1]: Entrando no diretório `/build/Pacotes/Monitor/libxml2-2.7.6'
Making all in include
make[2]: Entrando no diretório `/build/Pacotes/Monitor/libxml2-2.7.6/include'
Making all in libxml
make[3]: Entrando no diretório `/build/Pacotes/Monitor/libxml2-2.7.6/include/libxml'
make[3]: Nada a ser feito para `all'.
make[3]: Saindo do diretório `/build/Pacotes/Monitor/libxml2-2.7.6/include/libxml'
make[3]: Entrando no diretório `/build/Pacotes/Monitor/libxml2-2.7.6/include'
make[3]: Nada a ser feito para `all-am'.
make[3]: Saindo do diretório `/build/Pacotes/Monitor/libxml2-2.7.6/include'
root [ /build/Pacotes/Monitor/libxml2-2.7.6 ]# make install
Making install in include
make[1]: Entrando no diretório `/build/Pacotes/Monitor/libxml2-2.7.6/include'
Making install in libxml
make[2]: Entrando no diretório `/build/Pacotes/Monitor/libxml2-2.7.6/include/libxml'
make[3]: Entrando no diretório `/build/Pacotes/Monitor/libxml2-2.7.6/include/libxml'
make install-exec-hook
make[4]: Entrando no diretório `/build/Pacotes/Monitor/libxml2-2.7.6/include/libxml'
/bin/sh ../mkinstalldirs /usr/include/libxml2/libxml
make[4]: Saindo do diretório `/build/Pacotes/Monitor/libxml2-2.7.6/include/libxml'
test -z "/usr/include/libxml2/libxml" || /bin/mkdir -p "/usr/include/libxml2/libxml"
/usr/bin/install -c -m 644 SAX.h entities.h encoding.h parserInternals.h xmlerror.h
ointer.h xlink.h xmlmemory.h nanohttp.h nanoftp.h uri.h valid.h xlink.h xmlversion
dule.h xmlschemas.h schemasInternals.h xmlschematypes.h xmlstring.h xmlunicode.h xmlreader.h
/usr/bin/install -c -m 644 SAX2.h xmlexports.h xmlwriter.h chvalid.h pattern.h xmlsave.h sch
make[3]: Saindo do diretório `/build/Pacotes/Monitor/libxml2-2.7.6/include/libxml'
make[2]: Saindo do diretório `/build/Pacotes/Monitor/libxml2-2.7.6/include/libxml'

```

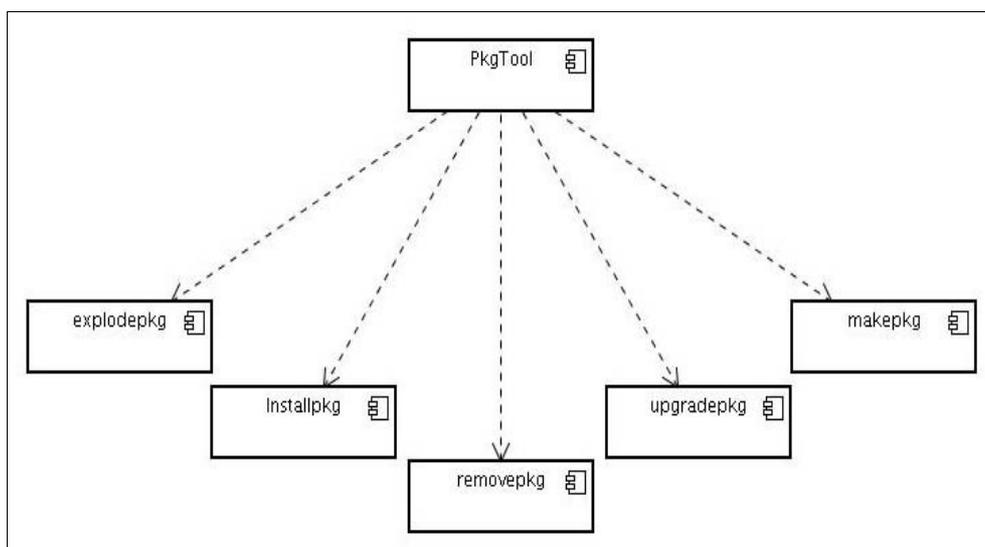
Figura 26 - Comandos utilizados na compilação dos fontes.

## 9.5 PkgTool

Distribuições atuais possuem um gerenciador de pacote responsável pela instalação, remoção e atualização dos pacotes. Estes gerenciadores manipulam apenas os pacotes pré-compilados desenvolvidos especialmente para eles.

Por serem flexíveis, sistemas baseados no LFS permitem a adição de grande parte dos recursos encontrados em outras distribuições, incluindo o gerenciamento de pacotes. Como uma alternativa ainda mais veloz a compilação, foi incluído no Senix o gerenciador *Pkgtool*, oriundo do Slackware [Slackware 2009], que permite a instalação e manipulação de pacotes através da utilização de vários outros *scripts* que o compõem (Figura 27). Esta ferramenta expande as possibilidades de atualização da distribuição com a simples remoção de um pacote obsoleto por outro atualizado, porém não trata de gerenciar as dependências dos pacotes, como acontece com a utilização do gerenciador *slapt-get* [Slapt-Get 2009].

A escolha desta ferramenta se baseia no fato de que o Slackware possui muitas soluções e documentações que podem ser utilizadas perfeitamente em um ambiente construído com o *Linux From Scratch*.



**Figura 27 – Diagrama de componentes da ferramenta Pkgtool.**

Na Tabela 3, estão listados os comandos para gerenciar o pacote do navegador Firefox (*firefox.tgz*) no sistema Senix.

**Tabela 3 - Comandos para Manipulação de Pacotes no Pkgtools.**

Opção	Descrição	Exemplo
<code>installpkg</code>	Instala o pacote	<code>installpkg firefox.tgz</code>
<code>removepkg</code>	Remove o pacote	<code>removepkg firefox.tgz</code>
<code>upgradepkg</code>	Atualiza o pacote	<code>upgradepkg firefox.tgz</code>
<code>explodepkg</code>	Extrai o conteúdo do pacote no diretório atual	<code>explodepkg firefox.tgz</code>
<code>makepkg</code>	Cria pacote	<code>makepkg seu-firefox.tgz</code>

## 9.6 Segurança no Senix

A segurança do sistema operacional tem se tornado um item cada vez mais relevante com o advento da internet e o surgimento de ferramentas que possibilitem a pessoas mal intencionadas capturar tráfego das informações, quebrar sistemas de encriptação, capturar senhas e explorar vulnerabilidades diversas.

No Linux este tipo de problema tende a ser tranquilo, pois vulnerabilidades em servidores muito utilizados, como o Apache e SSH, ocorrem, mas não são críticas [Secunia 2009]. Porém, ainda é necessário que o usuário tome algumas precauções como evitar deixar vários serviços desconhecidos ativados na máquina.

É evidente que tanto no Linux como no Windows o principal aspecto que determina a segurança do sistema é a atitude do usuário, pois fica ao encargo dele tomar medidas de segurança como o uso de um *firewall*, manter o sistema atualizado e ter discernimento o suficiente do que pode ou não ser executado [Revista Espaço Acadêmico 2005].

No caso do Senix, por se tratar de um Live-CD, ele não necessita de qualquer tipo de *software* para garantir a integridade das informações manipuladas, pois a diferente arquitetura do sistema elimina um imenso número de ameaças existentes, com exceção dos *rootkits* [Rootkit Analytys 2009], porém todas alterações efetuadas no ambiente do Live-CD serão perdidas quando o computador for reiniciado ou desligado, acabando também por neutralizar até este tipo de ataque.

## 9.7 Segurança de Redes

Foi implementado no projeto do Senix dois softwares que possuem relação direta com esta disciplina ministrada pelo professor Edécio Iepsen, o Iptables e o Shorewall. Esses aplicativos podem muito bem ser utilizados no aprendizado de qualquer disciplina envolvendo redes, para isso inclui documentações com bons exemplos para ambos projetos.

O Iptables é uma ferramenta de edição da tabela de filtragem de pacotes, ou *firewall* de estado, capaz de analisar o cabeçalho e tomar decisões sobre os destinos destes pacotes. Criando assim um poderoso sistema de *firewall* que armazena as conexões entrantes, evitando ataques do tipo *Stealth Scans*, que trazem *flags* especiais para técnicas de *port scanning* [Netfilter 2001]. Dentre as vantagens do Iptables encontram-se a estabilidade, rapidez e eficiência, além de sua configuração pode ser realizada através de *scripts*.

Se por um lado o Iptables é o firewall, o Shorewall é um "*front-end*" de configuração do Iptables, sendo possível implementar um firewall ou gateway através de entradas em um conjunto de arquivos de configuração [Shorewall 2009].

O Shorewall lê estes arquivos e informa ao Iptables as regras a serem implementadas. A vantagem da utilização do Shorewall é uma estrutura mais legível dos arquivos e regras do *firewall* e um número reduzido de linhas para implementação do código desejado.

## 10. Teste Realizados

Durante a fase de desenvolvimento do projeto o total de oito mídias foram distribuídas para utilização entre professores e alunos. Dos quais, em sua maioria, houve uma

aceitação positiva do trabalho, apesar de alguns pequenos problemas de configuração das ferramentas que acabaram por trazer sugestões realmente construtivas que agregaram mais conteúdo ao projeto.

O teste prático da distribuição foi necessário para se ter uma noção exata de como o sistema se comportaria em uso, por ser uma compilação de aplicativos voltados ao desenvolvimento, a maioria dos softwares encontrados no sistema são destinados a esse objetivo. Entretanto encontram-se também inúmeros softwares para estudo na área de redes ou que podem auxiliar na manutenção de computadores.

Com a utilização Senix por parte de alunos e professores foram relatados alguns problemas que foram corrigidos durante a fase de desenvolvimento da distribuição. Abaixo alguns problemas relatados e que foram sanados:

- Problema com a configuração do teclado, não sendo possível imprimir na tela alguns caracteres como o "/" e "ç", necessitando configuração do servidor X.
- Menus desordenados, não editáveis e sem categorias específicas impossibilitando a fácil localização do conteúdo.
- Falta de um papel de parede personalizado.
- Caracteres ilegíveis nos menus personalizados do gerenciador de pacotes *Pkgtools*.

Durante o desenvolvimento ocorreram alguns problemas detectados pelo desenvolvedor e que são listados a seguir:

- Alteração dos *scripts* de inicialização e configuração do idioma, fuso horário e teclado.
- Implementação do banco de dados MySQL devido ao desconhecido erro 2002 para o qual não há uma solução satisfatória, a não ser a reinstalação do aplicativo [MySQL 2009].
- Facilidade na corrupção dos dados na imagem do sistema de arquivos em ext2, sendo obrigatória a checagem do mesmo periodicamente.
- Dificil remoção de aplicativos já instalados antes do processo de remasterização.
- Documentação insatisfatória por parte do projeto XFCE, sendo mais fácil encontrar soluções em fóruns da comunidade GNU/Linux em Geral.

## 11. Resultados

De acordo com os testes realizados e o retorno dos usuários, o sistema se mostrou estável e atingiu o seu objetivo de possibilitar o uso de ferramentas nas unidades curriculares. Com base no parecer de seis alunos do 6º Semestre do curso de Análise e Desenvolvimento de Sistemas para quem as mídias foram distribuídas, o Senix se mostrou uma ferramenta extremamente útil para práticas nas unidades curriculares.

Segundo o professor Eduardo Monks que é o docente responsável pela unidades Redes de Computadores a utilização de ferramentas tais como o Wireshark (Figura 28), Tcpcap e Iperf possibilitam aos alunos a visualização do comportamento de uma rede de computadores.

Também conforme o relato do aluno Joel Telechi Melo que cursa o 6º Semestre do Curso de Análise e Desenvolvimento de sistemas foi possível exercer as atividades de desenvolvimento *web* em php e a modelagem das tabelas de banco de dados utilizando o

## DBDesigner em conjunto com o MySQL.

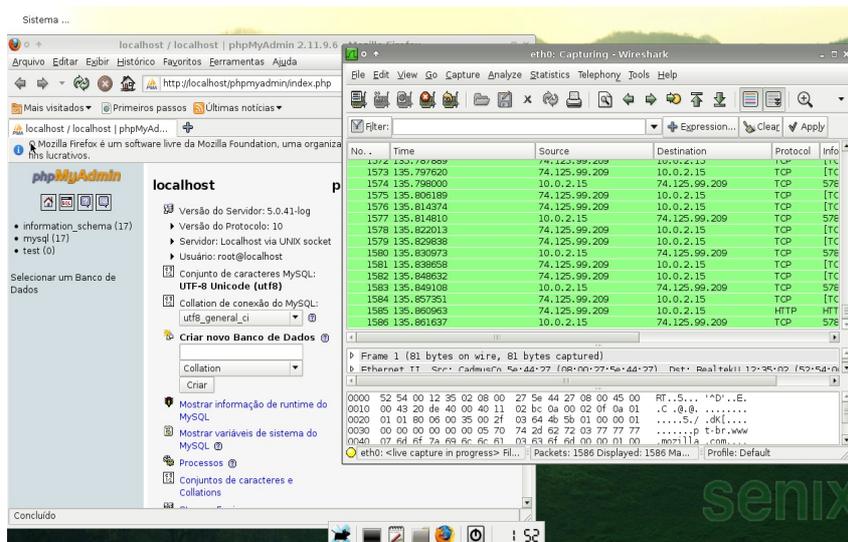


Figura 28 – Wireshark e PhpMyAdmin rodando no Senix.

Essas ferramentas foram incluídas no Senix e poderão ser utilizadas nas práticas dessa unidade em sala de aula e os alunos poderão ter também essa mesma facilidade em casa.

## 12. Conclusão

Este trabalho teve como objetivo desenvolver uma distribuição para utilização acadêmica na Fatec Senac Pelotas, descrevendo processos, requisitos e características da mesma.

O *Linux From Scratch* tornou possível o melhor entendimento das etapas do processo e funcionalidades que foram implementadas.

Com a utilização dos projetos integrados BLFS (*Beyond Linux From Scratch*) e CLFS (*Cross Beyond Linux From Scratch*), acrescentou-se as ferramentas que tem como objetivo atender as necessidades acadêmicas, além de permitir maior abrangência na atuação por parte da distribuição.

Existem inúmeras técnicas que permitem a construção de uma distribuição, porém tudo depende do objetivo e tempo disponível que o desenvolvedor possui.

Após este estudo, concluí-se que o melhor método para se construir e personalizar uma distribuição é através da técnica da remasterização. Pois é de fácil entendimento, possui diversas fontes em inúmeros idiomas, permitindo ao desenvolvedor construir uma nova distribuição a partir de outra desenvolvendo soluções, corrigindo problemas e agregando novos recursos.

Da proposta inicial do trabalho o objetivo foi alcançado que era oferecer ao usuário uma ferramenta de auxílio a aprendizagem e bucar a disseminação do Linux dentro do curso de Análise e Desenvolvimento de Sistemas da Fatec Senac Pelotas.

Como trabalhos futuros serão citados os seguintes itens:

- Criar um centro de controle que agregue todas funcionalidades e recursos existentes.
- Conceber uma nova versão da distribuição utilizando o futuro LFS 7.0 como base.

- Executar testes ainda mais extensivos para identificar todas as possíveis incompatibilidades com os diversos tipos de *hardware* encontrados no campo computacional da Fatec Senac Pelotas.

### 13. Referências Bibliográficas

- All Experts. (2009) “Zisofs at AllExperts”, <http://en.allexperts.com/e/z/zi/zisofs.htm>, Novembro.
- Ambrozio, F. (2008) “Criando pacotes no Slackware Linux”, <http://www.vivaolinux.com.br/artigo/Criando-pacotes-no-Slackware-Linux>, Outubro.
- Apache. (2009) “About the Apache HTTP Server Project”, [http://httpd.apache.org/ABOUT\\_APACHE.html](http://httpd.apache.org/ABOUT_APACHE.html), Novembro.
- Backtrack. (2009) “Backtrack”, <http://www.remote-exploit.org/backtrack.html>, Novembro.
- Beekmans, G.(1998) “Linux From Scratch”, <http://www.linuxfromscratch.org>, Agosto.
- Beekmans, G.(2009) “Download the Linux From Scratch Book”, <http://www.linuxfromscratch.org/lfs/download.html>, Setembro.
- Brijeski, T (2007) “RemasterSys Tool”. <http://www.geekconnection.org/remastersys/remastersystool.html>, Setembro.
- CE Linux. (2008) “SquashFsComparisons”, <http://tree.celinuxforum.org/CelfPubWiki/SquashFsComparisons>, Novembro.
- CdrKit. (2008) “Cdrkit - portable command-line CD/DVD recorder software”, <http://www.cdrkit.org>, Novembro.
- Danesh, A (1999) “Dominando o Linux: a bíblia”, página 9. Makron Books
- DistroWatch. (2009) “DistroWatch database summary”, <http://distrowatch.com/weekly.php?issue=20060828>, Outubro
- DreamLinux. (2006) “MKDistro um tutorial básico”, [http://dreamlinux.incubadora.fapesp.br/portal/tutoriais/mkdistro\\_basic\\_tutorial/mkdistro\\_pt1.html](http://dreamlinux.incubadora.fapesp.br/portal/tutoriais/mkdistro_basic_tutorial/mkdistro_pt1.html), Novembro.
- Fedora Project. (2009) “Overview – Fedora Project”, <http://fedoraproject.org/wiki/Overview>, Novembro.
- Freshmeat. (2001) “Zisofs in Freshmeat”, <http://freshmeat.net/projects/zisofs-tools>, Novembro.
- Gimp. (2009) “Gimp – Introduction”, <http://www.gimp.org/about/introduction.html>, Novembro.
- GnomeBaker. (2009) “GnomeBaker”, <http://sourceforge.net/projects/gnomebaker>, Novembro.
- Gnu. (2009) “The Gnu Operating System”, <http://www.gnu.org>, Novembro.
- Linux Foundation. (2009) “Linux Kernel Development”, <http://www.linuxfoundation.org/publications/linuxkerneldevelopment.php>, Outubro.
- Linux Devices. (2006) “Introducing initramfs, a new model for initial RAM disks”, <http://www.linuxfordevices.com/c/a/Linux-For-Devices-Articles/Introducing-initramfs-a-new-model-for-initial-RAM-disks>, Novembro.

Morimoto, C. (2005) “Entendendo como o Linux funciona - A árvore genealógica das distribuições”, <http://www.guiadohardware.net/tutoriais/entendendo-como-linux-funciona/arvore-genealogica-das-distribuicoes.html>, Outubro.

Morimoto, C. (2008) “Distribuições Linux: Os Live-CDs”, <http://www.gdhpress.com.br/blog/distribuicoes-live-cd>, Outubro.

Morimoto, C. (2009) “Mini-review do Vector Linux”, <http://www.gdhpress.com.br/blog/mini-review-vector>, Setembro.

MySQL. (2009) “Manual de Referência do MySQL - A.2.3 Erro: Can't connect to [local] MySQL server”, <http://dev.mysql.com/doc/refman/4.1/pt/can-not-connect-to-server.html#>, Novembro.

Nagios. (2009) “About Nagios”, <http://www.nagios.org/about>, Novembro.

Netfilter. (2001) “PACKET Filtering HOWTO”, <http://www.netfilter.org/documentation/HOWTO/pt/packet-filtering-HOWTO-7.html>, Novembro.

OpenOffice.org. (2009) “Why OpenOffice”, <http://why.openoffice.org>, Novembro.

Vector Linux. (2009) “Welcome to Vector Linux”, <http://vectorlinux.com>, Novembro.

VirtualBox. (2009) “VirtualBox”, <http://www.virtualbox.org>, Novembro.

Remastersys. (2008) “RemasterSys FAQ”, <http://klikit.pbworks.com/Remastersys+FAQ>, Novembro.

Revista Espaço Acadêmico. (2005) “Segurança da Informação e Disponibilidade de Serviços na Era da Internet”, <http://www.espacoacademico.com.br/044/44amsf.htm>, Outubro.

Rootkit Analytcs (2009) “RootkitAnalytics.com”, <http://rootkitanalytics.com>, Novembro.

Rpm. (2009) “Rpm”, [www.rpm.org](http://www.rpm.org), Novembro.

Secunia. (2009) “Secunia Advisories by Product”, [http://secunia.com/advisories/product/SOFT\\_A/#list](http://secunia.com/advisories/product/SOFT_A/#list), Novembro.

Shorewall. (2009) “Shoreline Firewall”, <http://www.shorewall.net>, Novembro.

Slackware. (2009) “Configuration Help”, <http://www.slackware.com/config/packages.php>, Setembro.

Slapt-Get. (2009) “Slapt-Get”. <http://software.jaos.org/#slapt-get>, Novembro.

Slax. (2009) “Slax: your pocket operating system”, [www.slax.org](http://www.slax.org), Novembro.

Window Maker. (2009) “Window Maker - Your Next Window Manager”, <http://www.windowmaker.info>, Novembro.

Tio Geek. (2008) “Instalando Ubuntu 8.10 no Pendrive USB”, <http://www.tiogeek.com/2008/10/kw=instalando-ubuntu-no-pendrive-usb>, Novembro.

Xfce. (2007) “Perguntas Frequentes”, <http://wiki.xfce.org/pt-br/faq>, Novembro.

Xfce. (2009) “Requerimentos do Sistema”, [http://wiki.xfce.org/pt-br/minimum\\_requirements](http://wiki.xfce.org/pt-br/minimum_requirements), Agosto.