

Linux System Administration 455

Sumário

Capítulo 1

Primeiro capítulo.....	8
1.1. Objetivos	8
1.2. Introdução	9
1.3. Distribuições GNU/Linux	11
1.4. Características de Algumas Distribuições	15
1.4.1. RedHat	16
1.4.2. SuSe	16
1.4.3. Mandriva	16
1.4.4. Slackware	17
1.4.5. Debian	17
1.4.6. Ubuntu.....	17
1.4.7. Fedora	18
1.4.8. OpenSuSe	18
1.4.9. Knoppix.....	18
1.4.10. Gentoo	19
1.5. Entendendo a estrutura do Linux	20
1.5. Diretório Recomendado	27
1.6. O diretório /sys	28
1.7. Diretórios Opcionais.....	28
1.8. Comandos de Movimentação	29

Capítulo 2 Comandos GNU/Linux.....	31
2.1. Objetivos.....	31

Capítulo 3 - Comandos Parte 2 e Editor de textos VI

.....	46
3.1. Objetivos.....	46
3.2. Editor de Texto VI (pronuncia-se “viai”).....	51
3.2.1. O VI possui 3 modos de operação:	53

Capítulo 4 - Inicialização do sistema Linux.....	56
4.1. Objetivos	56
4.2. Processo de inicialização do Linux.....	57
4.3. Arquivo /etc/inittab.....	58
4.4. Diretório /etc/init.d.....	60
4.5. Prática dirigida.....	63

4.6. Customizando os runlevels	65
4.6.1. Comando chkconfig:	65
4.6.2. Comando update-rc.d:	66
4.7. Arquivo /etc/fstab.....	69
Capítulo 5 - Dispositivos de bloco.....	71
5.1. Objetivos	71
5.2. Dispositivos em Linux	72
5.2.1. Explorando o /dev	72
5.1. Devices, UUID e Labels	78
5.1.1. Usando os dispositivos de armazenamento	80
5.1. Criando Partições no HD	82
5.1.1. Particionamento com FDISK	82
5.1.2. Particionamento com CFDISK	84
5.2. Aplicando um Filesystem	85
5.3. Arquivos de Informações de Filesystems	87
5.4. Configurações de Teclado no Console	88
5.5. Prática Dirigida 1	88
5.6. Para aprofundar o assunto	89
5.7. Prática Dirigida 2	89
Capítulo 6 - Gerenciamento de pacotes (instalação e remoção de programas).....	94
6.1. Objetivos:.....	94
6.2. Gerenciador de pacote dpkg	95
6.3. APT-GET (aptitude) - Advanced Packaging Tool	98
6.4. Gerenciamento de pacotes no Red Hat.....	103
6.5. Gerenciamento de pacotes com o yum.....	105
6.6. Compilando um programa disponibilizado como tar.gz ou tar.bz2.	108
Capítulo 7 Administração de usuários e permissões – Parte 1.....	111
7.1. Objetivos :.....	111
7.2. Introdução.....	112
7.3. Permissões no Linux.....	123
Capítulo 8 - Administração de usuários e permissões – parte 2.....	127
8.1. Objetivos	127
8.2. Prática dirigida: Modelo de como aplicar o conhecimento de permissões no dia-a-dia	134
8.2.1. Fase de criação.....	134
8.2.2. Permissão SGID BIT.....	143
8.2.3. Permissão STICK BIT.....	149

8.2.4. Permissão SUID.....	151
Capítulo 9 - Compactadores e agendamento de tarefas	153
9.1. Objetivos.....	153
9.2. Agendadores de tarefas do GNU/Linux	159
9.2.1. Agendador at.....	159
9.2.2. Crontab.....	160
Capítulo 10 - Shell Script Parte 1.....	165
10.1. Objetivos:.....	165
10.2. Introdução.....	166
10.3. Conhecimentos básicos.....	167
10.3.1. Crases.....	167
10.3.2. Variáveis.....	168
10.3.3. Cálculos em Shell Script - expr.....	168
10.3.4. Trabalhando com parâmetros	169
10.4. Estruturas Condicionais.....	172
Capítulo 11 Shell Script - Parte 2.....	175
11.1. Objetivos:.....	175
11.2. Introdução à Shell Script.....	176
11.3. Case.....	178
11.3.1. Script exemplo para case:	179
11.4. Estrutura de repetição o comando while	180
Capítulo 12 - Processos	183
12.1. Objetivo:.....	183
12.2. Introdução.....	184
12.3. Background e foreground	190
12.4. Prioridades de processos.....	193
12.5. Comandos complementares de processos.....	195
12.5.1. Comando fuser.....	195
12.5.2. Comando nohup.....	197
Capítulo 13 - Módulos.....	198
13.1. Objetivos :.....	198
13.2. Exemplo de configuração da placa de som.....	208
13.3. Exemplo de configuração da placa de rede.....	210
13.4. Configuração do teclado.....	213
13.4.1. Teclado em modo texto.	213
13.5. Configuração do mouse.....	218

Capítulo 14 Cotas de Disco.....	220
14.1. Objetivos:.....	220
14.2. Introdução:.....	221
14.3. Prática Dirigida.....	221
Capítulo 15 - LVM.....	234
15.1. Objetivos:.....	234
15.2. Introdução.....	235
15.3. Prática dirigida :.....	237
15.4. Usando o LVM	239
15.5. Administrando o LVM	240
15.6. Troubleshooting	241
Capítulo 16 - Rsyslog.....	243
16.1. Objetivos.....	243
16.1. Introdução teórica.....	244
16.1. Organização do Rsyslog.....	244
16.1. Facilidades do Syslog.....	245
16.2. Níveis.....	246
16.3. Destinos.....	246
16.4. Arquivos importantes	247
16.5. Prática dirigida.....	247
16.5.1. Verificando os logs.....	247
16.5.1. Logs Centralizados.....	248
16.5.1. Rotação de Logs.....	249
Capítulo 17 - Configurações de redes.....	253
17.1. Objetivos:.....	253
17.2. Introdução.....	254
17.3. O arquivo hostname.....	255
17.4. O arquivo /etc/hosts	256
17.5. O arquivo interfaces (Debian).....	257
17.6. O arquivo /etc/resolv.conf.....	262
Capítulo 18 - Wireless e NTP.....	264
18.1. Objetivos.....	264
Introdução.....	265
18.2. Prática Dirigida.....	266
18.2.1. Compilação do módulo da placa Wireless.....	266
18.2.2. Configurando uma rede wireless Ad-Hoc.....	268
18.2.3. Rede Wireless Managed - sem criptografia.....	269

18.2.4. Rede Wireless “Managed” - com criptografia WEP.....	271
18.3. NTP.....	272
Capítulo 19 - Acesso remoto.....	278
19.1. Objetivos :.....	278
19.2. Telnet – TELetype NETwork.....	280
19.2.1. Configurando o telnet num ambiente que usa inetd:	281
19.3. SSH - Secure SHell	284
Capítulo 20 Servidor X.....	296
20.1. Objetivos	296
20.2. Introdução Teórica	297
20.3. Configurando o suporte à Interface Gráfica	297
20.4. Variável de Ambiente DISPLAY	299
20.5. Window Managers	299
20.6. Display Managers	300
20.7. Protocolo XDMCP	300
20.8. Xnest	301
20.9. Prática Dirigida	301
20.9.1. Instalação e Configuração do Servidor X	301
20.9.2. Instalando um Window Manager	305
20.9.3. Display Managers	307
20.9.4. Usando o Xnest	308
20.9.5. Servidor X Remoto	310
Capítulo 21 - kernel.....	311
21.1. Objetivos :.....	311
21.2. Como recompilar o kernel no Linux.....	312
REFERÊNCIAS	
BIBLIOGRÁFICAS.....	320
ANEXOS.....	322
Primeiro anexo.....	323
Tópico do Anexo	323

Índice de tabelas

Índice de Figuras

Capítulo 1

Primeiro capítulo

1.1. Objetivos

- Descobrir o que é GNU/Linux;
- Entender a filosofia do Software Livre;
- Conhecer um pouco da história;
- Entender o que são Distribuições;
- Introdução ao Linux ;
- Estrutura de Diretórios;

1.2. Introdução

Utilizar um sistema GNU/Linux é muito mais do que optar por uma solução isenta de custos de licença. É usufruir uma filosofia que antecedeu o software proprietário, e que permitiu, por exemplo, que a internet crescesse de forma aberta como a conhecemos hoje. Como usuário de software livre, precisamos compreender um pouco mais sobre essa ideologia e como ela promoveu o surgimento das várias distribuições.

O sistema GNU/Linux é frequentemente chamado apenas pelo seu segundo nome, Linux. Entretanto, essa designação não faz justiça a todos os desenvolvedores que vêm desenvolvendo o sistema.

GNU, que é um acrônimo recursivo de GNU's not Unix, é um grupo que foi fundado em 1984 por seu idealizador, Richard Stallman, com o intuito de criar um sistema operacional "Unix-Like". Sendo assim, diversos softwares passaram a ser criados e mantidos pela comunidade que se formara, entretanto, havia um pedaço de código essencial que ainda não tinha sido criado: o kernel.

Em 1991, um jovem finlandês chamado Linus Torvalds disponibilizou para o mundo a primeira versão do Linux, um kernel "Unix-Like". A partir desse ponto, foi possível unir o kernel - Linux - com os softwares GNU, originando o que chamamos de GNU/Linux.

O mundo GNU/Linux não é apenas um conjunto de programas mas também uma filosofia de mundo livre e colaborativo, no qual as pessoas podem utilizar esses softwares livremente e, acima de tudo, aprender com eles, uma vez que seu código fonte deve ser disponível a todos que queiram melhorá-lo ou apenas aprender com ele. Para que esse mundo continue livre, Richard Stallman fundou a FSF - Free Software Foundation, que mantém a licença chamada GNU GPL - GNU General Public License.

- liberdade 0 - liberdade para rodar o programa para quaisquer propósitos;
- liberdade 1 - liberdade para estudar como o programa trabalha e adaptá-lo às suas necessidades. Ter acesso ao código fonte é essencial para isso.
- liberdade 2 - liberdade de redistribuir cópias de forma que você possa ajudar outras pessoas.
- liberdade 3 - liberdade para melhorar o programa e disponibilizar as melhorias para o público, de forma que toda a comunidade possa se beneficiar disso. Ter acesso ao código fonte é essencial também para isso.

Após a criação dessa licença, várias outras licenças, usando a filosofia de copyleft, foram criadas com o objetivo de defender a liberdade do conhecimento, informação e do código aberto. Abaixo podemos ver alguns exemplos:

- GFDL - GNU Free Documentation Licence (www.gnu.org/copyleft/fdl.html);
- OPL - Open Publication License (<http://www.opencontent.org/openpub/>);
- CC - Creative Commons (<http://creativecommons.org/about/licenses>);
- BSD - Berkeley Software Distribution (<http://www.freebsd.org/copyright/license.html>);
- SPL - Sun Public Licence (<http://java.sun.com/spl.html>);

Atualmente a GPL está disponível em três versões, GPLv1, GPLv2 e GPLv3. Fique por dentro de suas diferenças em: <http://www.gnu.org/licenses/gpl.html>

Para mais informações a respeito do kernel - Linux - podem ser obtidas no site oficial de seus mantenedores: <http://www.kernel.org>

A respeito do GNU e da FSF podem ser obtidas nos sites

- <http://www.gnu.org>
- <http://www.fsf.org>.

1.3. Distribuições GNU/Linux

Você já deve ter ouvido falar em Debian, RedHat, Slackware, SuSe, Conectiva, Mandrake, Ubuntu dentre outras. Mas, o que realmente é isso? O que são todos esses nomes? Todos esses nomes são o que chamamos de distribuições GNU/Linux. Uma distribuição nada mais é do que o kernel, Linux, softwares GNU e outros aplicativos que são desenvolvidos por outras comunidades ou grupos.

Mas, por que tantas distribuições? Justamente porque se você não se identifica com nenhuma delas, você é livre para fazer a sua própria. Ou seja, em 1993, um rapaz chamado Patrick Volkerding, juntou o kernel e vários outros aplicativos em uma distribuição chamada Slackware, que foi a primeira a ser distribuída em CD. A partir desse ponto, foram surgindo diversas outras distribuições que de alguma forma diferiam da filosofia do Slackware: como Debian ou RedHat, por exemplo.

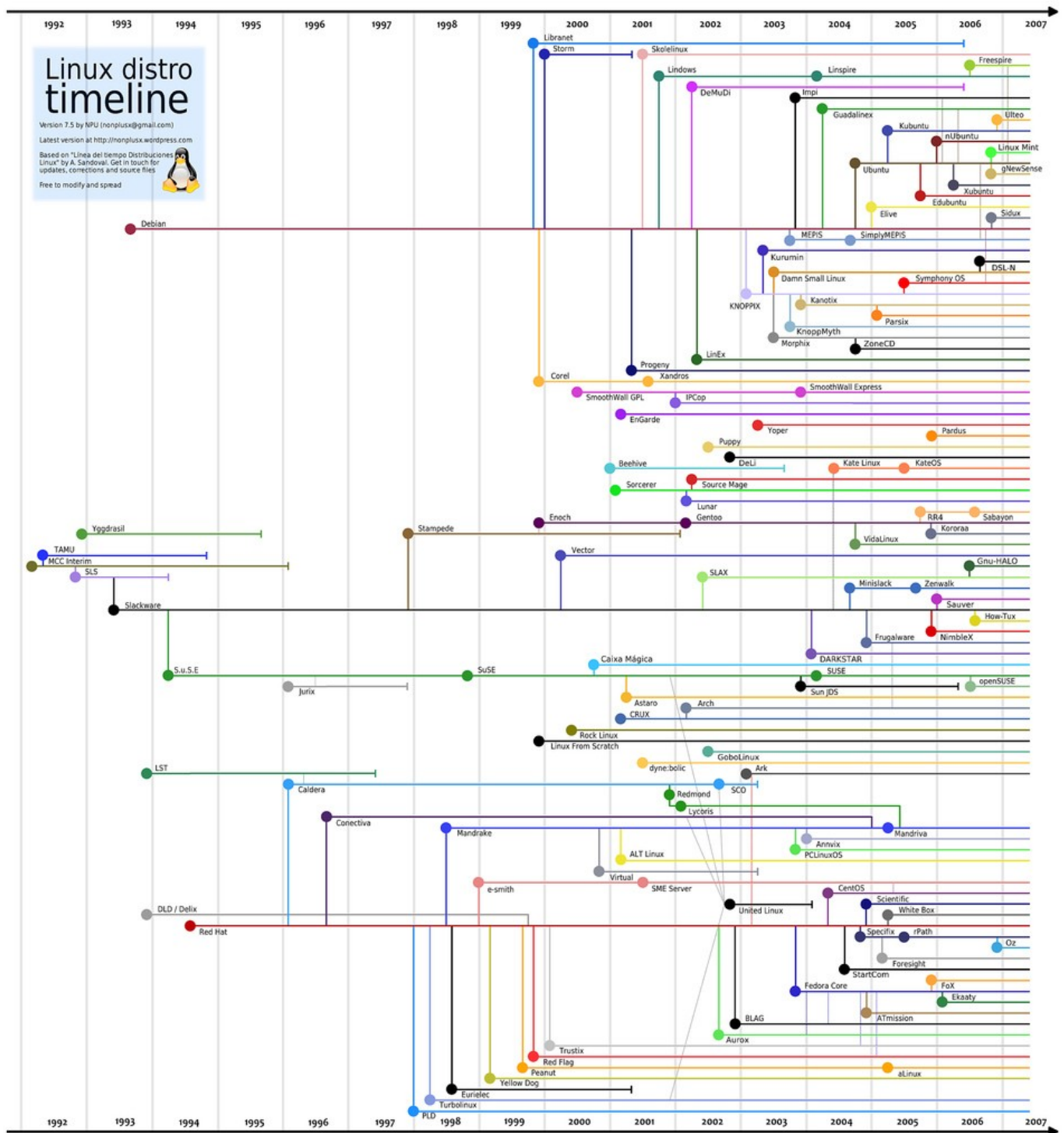


Ilustração 1: Linha de tempo GNU/Linux

Atualmente existem centenas de distribuições, algumas mais famosas que outras. Em sua maioria, as distribuições GNU/Linux são mantidas por grandes comunidades de colaboradores, entretanto, há outras que são mantidas por empresas. Dessa forma, podemos dividir as distros, abreviação bastante utilizada na comunidade e que se refere às distribuições, em duas categorias básicas:

- Livres
- Corporativas

Distribuições Livres - mantidas por comunidades de colaboradores sem fins lucrativos. Exemplos são: Debian, Ubuntu, Slackware, Gentoo, Knoppix e CentOS, entre outras.

Distribuições Corporativas - mantidas por empresas que vendem o suporte ao seu sistema. Exemplos são: RedHat, SuSe e Mandriva.

Neste ponto vale ressaltar o fato de que o produto vendido pelas empresas que comercializam sistemas GNU/Linux, são na verdade, os serviços relacionados ao sistema vendido, como suporte técnico, garantias e treinamentos, ou seja, o conhecimento do sistema. O fato de o produto não ser mais o software, mas sim o serviço, é devido à Licença GPL que garante as já citadas quatro liberdades básicas. Com isso, por mais que uma empresa queira fazer o seu próprio sistema GNU/Linux, enquanto ela estiver utilizando softwares registrados com GPL, serão obrigadas a distribuir o código fonte gratuitamente.

Dentro do conjunto de Distribuições Livres, podemos dividi-las novamente em duas outras categorias:

- Convencionais
- Live

Distribuições Convencionais- são distribuídas da forma tradicional, ou seja, uma ou mais mídias que são utilizadas para instalar o sistema no disco rígido;

Distribuições Live - são distribuídas em mídias com o intuito de rodarem a partir delas, sem a necessidade de instalar no HD. As distribuições Live ficaram famosas pois têm a intenção de fornecer um sistema GNU/Linux totalmente funcional, de forma fácil e sem a necessidade de o instalar na máquina. O fator que favoreceu essa abordagem é que em uma distribuição Live praticamente todos os componentes já vêm configurados, funcionando e com interfaces agradáveis aos usuários finais. Exemplos desse tipo de distribuição são o Knoppix, do qual se originaram diversas outras como Kurumin ou Kalango, que são versões brasileiras do Knoppix, e o Ubuntu, bastante difundido atualmente. Ainda para entender um pouco mais das distribuições, é necessário lembrar de mais duas características:

- From scratch
- Provenientes (Baseadas)
-

Distribuições From Scratch - São desenvolvidas do zero, ou seja, utiliza um kernel linux, alguns programas GNU e a grande maioria das suas particularidades é desenvolvida especificamente para ela. Exemplos:

- Debian ;
- RedHat;
- Gentoo;
- Slackware;
- entre outras;

Distribuições Provenientes (Baseadas) - Aproveitam ferramentas e bases já desenvolvidas por outras distribuições. Distribuições baseadas usam distribuições from scratch para alcançar seus objetivos mais rápido, dando maior atenção para ao propósito da distribuição. Exemplos: Ubuntu, DreamLinux, Kubuntu, Kurumin, Slax, BrDesktop entre muitas outras.

1.4. Características de Algumas Distribuições

Será mostrado a seguir a característica de algumas distribuições. Você pode encontrar uma lista das distribuições existentes, bem como das estatísticas de downloads, no site:

- <http://distrowatch.com>

1.4.1. RedHat

- **Tipo:** corporativa;

• **Descrição:** primeira distribuição corporativa a ser criada. Muito utilizada nas empresas por oferecer suporte técnico e ter seu sistema compatível com as diversas tecnologias disponíveis;

- **Interface padrão:** GNOME;
- **Sistema de pacote:** RPM - RedHat Package Manager;
- **Site oficial:** <http://www.redhat.com>

1.4.2. SuSe_

- **Tipo:** corporativa;

• **Descrição:** Comprada pela Novell em 2003, é a principal concorrente da RedHat, atuando no meio corporativo tanto em servidores quanto em desktops. Assim como a RedHat, possui parcerias com diversas empresas, a fim de manter seu sistema compatível com produtos de terceiros;

- **Interface padrão:** GNOME;
- **Sistema de pacote:** baseado em RPM, mas não segue o formato da RedHat à risca, tendo implementado algumas variações;
- **Site oficial:** <http://www.novell.com/linux>

1.4.3. Mandriva_

- **Distribuição:** corporativa;

• **Descrição:** originada da fusão da Mandrake e Conectiva, especializada em serviços e projetos embarcados;

- **Interface padrão:** KDE;
- **Sistema de pacote:** RPM;
- **Site oficial:** <http://www.mandriva.com>

1.4.4. Slackware_

• **Distribuição:** livre;

• **Descrição:** primeira distribuição GNU/Linux a ser distribuída em CD, é considerada como sendo a primeira distribuição. Organizada por seu criador Patrick Volkerding, caracteriza-se por sua filosofia de ser a distribuição mais ``Unix-Like" do mundo GNU/Linux.

• **Interface padrão:** KDE;

• **Sistema de pacote:** tgz;

• **Site oficial:** <http://www.slackware.com>

1.4.5. Debian_

• **Distribuição:** livre;

• **Descrição:** criada com o intuito de prover um sistema operacional totalmente livre e gratuito, foi uma das primeiras distribuições GNU/Linux a serem criadas. Atualmente é uma das maiores distribuições e a que mais gerou distribuições derivadas. Por ser uma referência em sistemas GNU/Linux, é a distribuição mais utilizada em órgãos públicos e governos;

• **Interface padrão:** GNOME;

• **Sistema de pacote:** DEB - Debian Package;

• **Site oficial:** <http://www.debian.org>

1.4.6. Ubuntu

• **Distribuição:** livre (convencional e Live);

• **Descrição:** com seu slogan Linux for Human Beings - é voltada para o usuário final, apesar de ter versão para servidores. Patrocinada pelo milionário Mark Shuttleworth é, atualmente, a maior distribuição em número de downloads.

• **Interface padrão:** GNOME ou KDE (para Kubuntu);

• **Sistema de pacote:** DEB - Debian Package;

• **Site oficial:** <http://www.ubuntu.com>

1.4.7. Fedora

- **Distribuição:** livre;

- **Descrição:** mantida pela RedHat, serve de teste para o carro chefe da empresa, o RedHat Enterprise.

- **Interface padrão:** GNOME;

- **Sistema de pacote:** RPM - RedHat Package Manager;

- **Site oficial:** <http://fedora.redhat.com>

1.4.8. OpenSuSe

- **Distribuição:** livre;

- **Descrição:** patrocinada pela Novell, baseia-se no SuSe Linux.

- **Interface padrão:** GNOME ou KDE;

- **Sistema de pacote:** RPM;

- **Site oficial:** <http://en.opensuse.org>

1.4.9. Knoppix

- **Distribuição:** livre (Live);

- **Descrição:** distribuição Live que popularizou distribuições do gênero, devido à sua ferramenta de remasterização que facilitou o processo de gerar novas distribuições;

- **Interface padrão:** KDE;

- **Sistema de pacote:** DEB;

- **Site oficial:** <http://www.knoppix.org>

1.4.10. Gentoo

• **Distribuição:** livre (Live);

• **Descrição:** Todos os programas são compilados na própria máquina. As principais vantagens são a performance e a personalização conforme as necessidades do usuário. A principal desvantagem é o trabalho e tempo necessários a sua instalação.

• **Interface padrão:** A escolha do usuário;

• **Sistema de pacote:** Emerge, código fonte;

• **Site oficial:** <http://www.gentoo.org>

1.5. Entendendo a estrutura do Linux

Como podemos ver na figura abaixo, o sistema operacional GNU/Linux pode ser dividido em algumas layers:

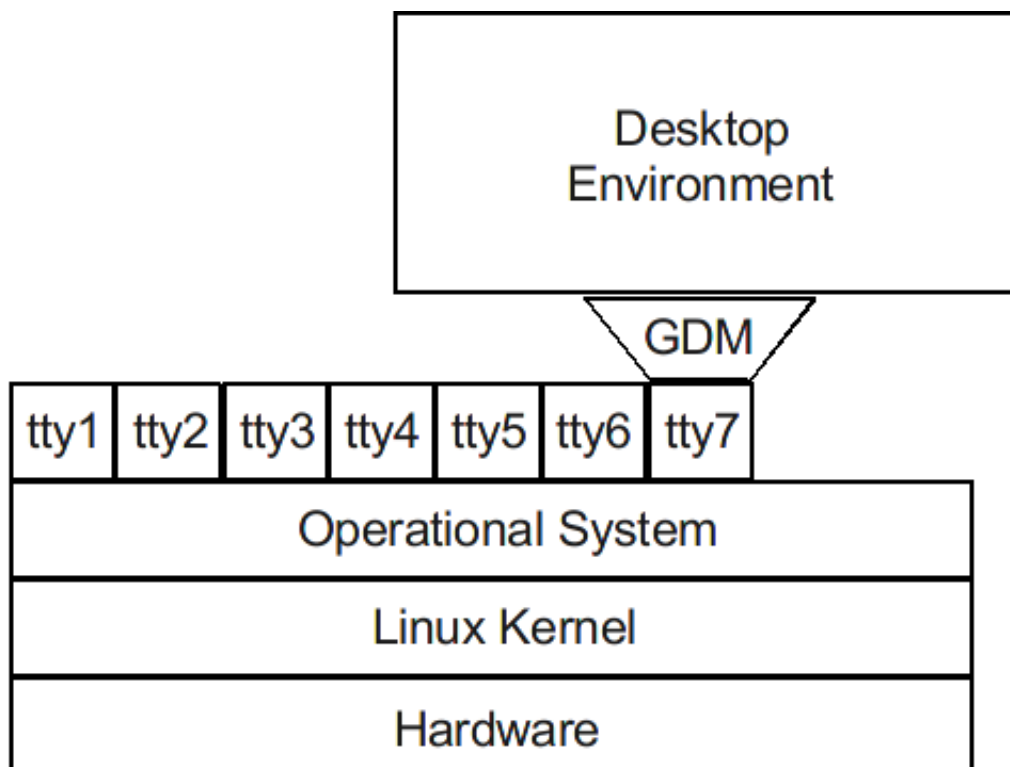


Ilustração 2: Estrutura do Sistema

Para entendermos melhor vamos descrever cada uma delas:

- **Hardware** - Dispositivos que estão disponíveis para o uso do sistema, tais como cd-rom, placa de rede, controladora scsi entre outros;

- **Kernel** - O núcleo do sistema operacional, essa layer é quem faz todas as interações com o hardware da máquina, interpretando todas as requisições das layers acima;

- **Sistema Operacional** - Essa layer tem como função auxiliar e abrigar todos os aplicativos das layers superiores. Segundo Linux torvalds esse layer não deve ser notada por um usuário leigo final;

•**ttyN** - Terminais Virtuais aonde são executados comandos e seta-das configurações. As tty's interpretam os comandos dados por um humano e converte os mesmos para uma linguagem que a máquina entenda;

•**DM** - A Layer de Display Manager é responsável por gerenciar os logins na interface gráfica e escolher o tipo de ambiente gráfico que deve ser executado;

•**Desktop Environment** - Mais conhecido como gerenciador de janela, é responsável por abrigar todos os programas que necessitam um gerenciador de janelas, e por deixar o ambiente mais agradável.

Quem já teve algum contato com o GNU/Linux, mesmo que superficial, deve ter percebido a presença de vários diretórios (pastas) no sistema. Entretanto, eles estão organizados de uma forma talvez não muito familiar. Neste capítulo, vamos conhecer a organização e explorar a estrutura de diretórios de um sistema GNU/Linux.

Desde que o GNU/Linux foi criado, muito se tem feito para seguir um padrão em relação à estrutura de diretórios. O primeiro esforço para padronização de sistemas de arquivos para o GNU/Linux foi o **FSSTND - Filesystem Standard**, lançado no ano de 1994.

Cada diretório do sistema tem seus respectivos arquivos que são armazenados conforme regras definidas pela **FHS - Filesystem Hierarchy Standard**, ou Hierarquia Padrão do Sistema de Arquivos, que define que tipo de arquivo deve ser guardado em cada diretório. Isso é muito importante, pois o padrão ajuda a manter compatibilidade entre as versões Linux existentes no mercado, permitindo que qualquer software escrito para o GNU/Linux seja executado em qualquer distribuição desenvolvida de acordo com os padrões FHS.

Atualmente, o FHS está na sua versão 2.3, e é mantido pelo *Free Standard Group*, uma organização sem fins lucrativos formada por grandes empresas como HP, IBM, Red Hat e Dell.



A FHS estar bem esclarecida, afinal é com ela que nós devemos fazer nossas atividades do dia-a-dia

A estrutura de diretórios também é conhecida como ``Árvore de Diretórios'' porque tem a forma de uma árvore. Mas, antes de estudarmos a estrutura de diretórios, temos que ter em mente o que são diretórios.

Um diretório nada mais é do que o local onde os arquivos são guardados no sistema. O arquivo pode ser um texto, uma imagem, planilha, etc. Os arquivos devem ser identificados por nomes para que sejam localizados por quem deseja utilizá-los.

Um detalhe importante a ser observado é que o GNU/Linux é *case sensitive*, isto é, ele diferencia letras maiúsculas e minúsculas nos arquivos e diretórios.

Sendo assim, um arquivo chamado Arquivo é diferente de **ARQUIVO** e diferente de **arquivo**.

A árvore de diretórios do GNU/Linux tem a seguinte estrutura:



/								
<i>bin</i>	<i>cdrom</i>	<i>etc</i>	<i>lib</i>	<i>mnt</i>	<i>proc</i>	<i>root</i>	<i>var</i>	
<i>boot</i>	<i>dev</i>	<i>home</i>	<i>media</i>	<i>opt</i>	<i>sbin</i>	<i>srv</i>	<i>tmp</i>	<i>usr</i>

Da estrutura mostrada acima, o FHS determina que um sistema GNU/Linux deve conter obrigatoriamente 14 diretórios, especificados a seguir:



/ (raiz)

Este é o principal diretório do GNU/Linux, e é representado por uma ``/' (barra). É no diretório raiz que ficam todos os demais diretórios do sistema.

Estes diretórios, que vamos conhecer agora, são chamados de subdiretórios pois estão dentro do diretório /.

`/bin`

O diretório `/bin` guarda os comandos essenciais para o funcionamento do sistema.

Esse é um diretório público, sendo assim, os comandos que estão nele podem ser utilizados por qualquer usuário do sistema. Entre os comandos, estão:

- `bash`;
- `ls`;
- `echo`;
- `cp`;

/boot

No diretório /boot estão os arquivos estáticos necessários à inicialização do sistema, e o gerenciador de boot.

O gerenciador de boot é um programa que carrega um sistema operacional e/ou permite escolher qual será iniciado.

/dev

No diretório /dev ficam todos os arquivos de dispositivos. O Linux faz a comunicação com os periféricos por meio de links especiais que ficam armazenados nesse diretório, facilitando assim o acesso aos mesmos.

/etc

No diretório /etc estão os arquivos de configuração do sistema. Nesse diretório vamos encontrar vários arquivos de configuração, tais como: scripts de inicialização do sistema, tabela do sistema de arquivos, configuração padrão para logins dos usuários, etc.

/lib

No diretório /lib estão as bibliotecas compartilhadas e módulos do kernel . As bibliotecas são funções que podem ser utilizadas por vários programas.

/media

Ponto de montagem para dispositivos removíveis, tais como:

- cd;
- dvd;
- disquete;
- pendrive;
- câmera digital;



Fique atento: Agora o diretório /media faz parte oficialmente das provas da LPI

/mnt

Esse diretório é utilizado para montagem temporária de sistemas de arquivos, tais como compartilhamentos de arquivos entre Windows e Linux, Linux e Linux, etc.

/opt

Normalmente, é utilizado por programas proprietários ou que não fazem parte oficialmente da distribuição.

/sbin

O diretório /sbin guarda os comandos utilizados para inicializar, reparar, restaurar e/ou recuperar o sistema. Isso quer dizer que esse diretório também é de comandos essenciais, mas os mesmos são utilizados apenas pelo usuário root.

Entre os comandos estão:

- halt
- ifconfig
- init
- iptables

/srv

Diretório para dados de serviços fornecidos pelo sistema cuja aplicação é de alcance geral, ou seja, os dados não são específicos de um usuário.

Por exemplo:

- /srv/www (servidor web)
- /srv/ftp (servidor ftp)

/tmp

Diretório para armazenamento de arquivos temporários. É utilizado principalmente para guardar pequenas informações que precisam estar em algum lugar até que a operação seja completada, como é o caso de um download.

Enquanto não for concluído, o arquivo fica registrado em /tmp, e, assim que é finalizado, é encaminhado para o local correto.

/usr

O diretório /usr contém programas que não são essenciais ao sistema e que seguem o padrão GNU/Linux, como, por exemplo, navegadores, gerenciadores de janelas, etc.



O diretório /usr é portátil, perceba que dentro dele, existe praticamente uma outra arvore de diretórios independente da primeira, contendo, lib, bin, sbin dentre outras coisas.

/var

O diretório /var contém arquivos de dados variáveis. Por padrão, os programas que geram um arquivo de registro para consulta, mais conhecido como log, ficam armazenados nesse diretório. Além do log, os arquivos que estão aguardando em filas, também ficam localizados em /var/spool.

Os principais arquivos que se utilizam do diretório /var são :

- mensagens de e-mail;
- arquivos a serem impressos;

1.5. Diretório Recomendado

/proc

O /proc é um diretório virtual, mantido pelo kernel, onde encontramos a configuração atual do sistema, dados estatísticos, dispositivos já montados, interrupções, endereços e estados das portas físicas, dados sobre as redes, etc.

Aqui, temos subdiretórios com o nome que corresponde ao PID (Process ID) de cada processo.

Dentro deles, vamos encontrar diversos arquivos texto contendo várias informações sobre o respectivo processo em execução.

1.6. O diretório /sys

Pode-se dizer que esse diretório é um primo do diretório /proc. Dentro do diretório /sys podemos encontrar o quase o mesmo conteúdo do proc, mas de uma forma bem mais organizada para nós administradores.

Esse diretório está presente desde a versão 2.6 do kernel e traz novas funcionalidades o que se diz respeito a dispositivos PnP.

1.7. Diretórios Opcionais

Os diretórios /root e /home podem estar disponíveis no sistema, mas não precisam obrigatoriamente possuir este nome.

Por exemplo, o diretório /home poderia se chamar /casa, que não causaria nenhum impacto na estrutura do sistema.

`/home`

O /home contém os diretórios pessoais dos usuários cadastrados no sistema.

`/root`

Diretório pessoal do superusuário root.

O root é o administrador do sistema, e pode alterar a configuração (dele), configurar interfaces de rede, manipular usuários e grupos, alterar a prioridade dos processos, entre outras.

Dica: Utilize uma conta de usuário normal em vez da conta root para operar seu sistema.



Uma razão para evitar usar privilégios root é por causa da facilidade de se cometer danos irreparáveis como root; além do que, você pode ser enganado e rodar um programa Cavalo de Tróia (programa que obtém poderes do super usuário) comprometendo a segurança do seu sistema sem que você saiba.

1.8. Comandos de Movimentação

Vamos aprender agora alguns comandos essenciais para a nossa movimentação dentro do sistema.

O comando `pwd` exibe o diretório corrente. Ele é muito útil quando estamos navegando pelo sistema e não lembramos qual o diretório atual.



```
# pwd
```

O comando `cd` é utilizado para mudar o diretório atual de onde o usuário está.

Ir para o diretório home do usuário logado:



```
# cd
```

```
# cd ~
```

Ir para o início da árvore de diretórios, ou seja, o diretório `/` :



```
# cd /
```

Ir para um diretório específico:



```
# cd /etc
```

Sobe um nível na árvore de diretórios:



```
# cd ..
```

Retorna ao diretório anterior:



```
# cd -  
# ls
```

Entra em um diretório específico:



```
# cd /usr/include/X11
```

Sobe 2 níveis da árvore de diretórios



```
# cd ../../
```



Atenção! Note a diferença entre caminhos absolutos e relativos:

Absolutos: /etc/ppp; /usr/share/doc; /lib/modules

Relativos: etc/ppp; ../doc; ../../usr;



Fique esperto para conhecer as diferenças entre o . e o .. e o que eles representam para o sistema. Os comandos de movimentação muitas vezes são grandes alvos nas provas, uma boa interpretação desses comandos pode ser necessária, pois você pode precisar deles para resolver uma questão maior.