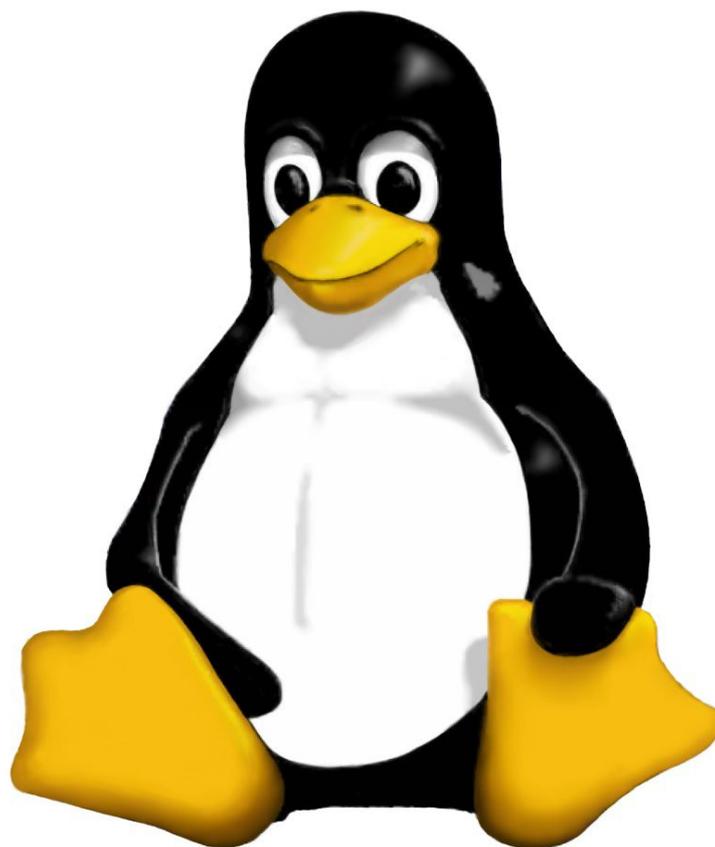


LINUX

Da Teoria à Prática



Antonio Carlos Feitosa Costa (antonio@cbpf.br)

Prefácio

Há alguns anos, a mentalidade dos usuários de computador se resumia apenas em ligá-lo e usar alguns programas, transformando o computador numa ótima ferramenta de trabalho. Hoje em dia, o panorama da informática é muito mais complexo do que se imagina. O mundo inteiro, involuntariamente, está completamente dependente do computador. E com o surgimento do movimento *Free Software* ou Software Livre, o universo da informática nunca mais foi o mesmo. Emergindo deste admirável mundo novo surgiram uma infinidade de projetos de software livre com as mais diversas aplicações, indo de mero celular com internet até softwares de posicionamento global, servidores de grandes empresas e etc. Os projetos de software livre que mais se destacaram neste cenário tecnológico foram o Linux e o projeto GNU.

Esta nota técnica dá uma visão geral do que é o Linux, como ele funciona e como podemos usá-lo em nosso dia-a-dia.

Conteúdo

1. Introdução ao Linux.....	3
2. Estrutura básica de uma distribuição Linux.....	6
3. Como funciona o Linux.....	9
4. Estrutura de arquivos e diretórios.....	11
5. Comandos básicos do Sistema.....	16
6. Instalação de uma distribuição linux.....	17

1. Introdução ao Linux

O que é Linux?

O Linux é um sistema operacional derivado do Unix[1] feito para rodar em computadores pessoais.

O Linux faz tudo o que você poderia esperar de um Unix moderno e completo. Suporta multitarefa real, memória virtual, bibliotecas dinâmicas, redes TCP/IP, nomes de arquivos com até 255 caracteres e proteção entre processos (crash protection), além de muitas outras funcionalidades que deixariam esta lista extensa demais.

Um grande atrativo que o Linux oferece é o fato de poder trabalhar tanto como servidor de aplicações quanto como estação de trabalho, sem que haja necessidade de grandes modificações no seu sistema.

Como surgiu o Linux?

O Linux foi originalmente desenvolvido como um passatempo de Linus Torvalds. Ele queria um sistema operacional que fosse semelhante a um Unix, com todas as suas funcionalidades e, ainda, que pudesse utilizá-lo num PC.

A partir dessa idéia, Linus começou a trabalhar nesse que seria o futuro kernel do sistema operacional que hoje é chamado de Linux. Isso tudo aconteceu em meados de 1991, quando Linus cursava a faculdade de Computação na Finlândia.

Em 5 de outubro de 1991 a seguinte mensagem circulou na usenet[2]:

" ...Como eu mencionei há um mês, estou trabalhando em uma versão free de um sistema semelhante ao Minix para computadores AT-386. Ele já alcançou o estágio de ser usável (embora possa não ser, dependendo do que você quer fazer), e pretendo distribuir o código fonte. É apenas a versão 0.02... mas já consegui rodar nele o bash, gcc, gnumake, gnu-sed, compress, etc."

Esta mensagem era assinada por Linus Torvalds, e ninguém adivinharia que ela estaria marcando o início de um movimento que, menos de dez anos depois, já teria mais de trinta milhões de seguidores.

Assim surgiu o que seria o primeiro kernel utilizável do Linux. Faz-se necessária aqui uma explicação sobre o que é o kernel do Linux, para evitar concepções errôneas do que é o kernel e o que é o Linux propriamente dito. O kernel é o núcleo do sistema operacional, é a parte que controla diretamente o hardware[3] da máquina. Quando se fala de Linux, está se referindo somente ao kernel do sistema. Tudo que existe ao redor do kernel são aplicativos que compõem uma distribuição do Linux.

[1] Um sistema operacional comercial baseado no padrão POSIX.

[2] Sistema de News, mural on-line.

[3] Parte física do computador

O que é uma Distribuição Linux?

Pelo fato de o Linux ser um software de livre distribuição, muitas pessoas e até mesmo empresas se empenham em organizar o kernel e mais uma série de aplicativos e manuais para que o sistema fique cada vez mais amigável.

A esse conjunto de aplicativos mais o kernel dá-se o nome de distribuição Linux. Algumas distribuições Linux são maiores que outras, dependendo da quantidade de aplicativos e a finalidade a que se propõem. Existem desde distribuições que cabem num disquete de 1.44Mb até distribuições que ocupam vários Cds.

Cada uma delas têm seu público-alvo e finalidades específicas. As minidistribuições[4] têm como objetivo desde a recuperação de um sistema danificado até o monitoramento de uma rede de computadores.

Entre as "grandes"[5] distribuições pode-se citar: Slackware, Debian, Ubuntu, Mandriva, SuSE, Red Hat Enterprise, Fedora e Gentoo. O que diferencia uma distribuição de outra é a maneira como são organizados e préconfigurados os aplicativos que cada uma contém.

Um exemplo: o Ubuntu Desktop Linux tem quase a totalidade de seus aplicativos traduzidos para as mais diversas línguas: português, espanhol, inglês, Italiano, facilitando a integração com os usuários espalhados pelo mundo. O que não quer dizer que "todos" os aplicativos estejam disponíveis no idioma local também.

Algumas distribuições incluem ferramentas de configuração que facilitam a vida do administrador do sistema.

[4] Distribuições que cabem em unidades poucos disquetes ou Pendrives.

[5] Distribuições que são fornecidas por imagens de CD (.iso) via Internet ou por venda do pacote com manual e mídias (CD ou DVD).

2. Estrutura básica de uma distribuição Linux

O sistema operacional Linux é composto por três partes:

1. Kernel.
2. Aplicações de Sistema.
3. Aplicações de Usuário.

O kernel do Linux

Embora o kernel seja uma parte importante do Linux, ele sozinho não constitui o sistema GNU/Linux. O kernel é o "núcleo" do sistema e é responsável pelas funções de mais baixo nível, como o gerenciamento de memória, gerenciamento de processos e da CPU. O kernel também é o responsável pelo suporte aos sistemas de arquivos, dispositivos e periféricos conectados ao computador, como placas SCSI, placas de rede, de som, portas seriais, etc. É chamado Linux o conjunto do kernel e demais programas, como shells, compiladores, bibliotecas de funções, etc.

Alguns cuidados devem ser tomados ao se verificar que versão de Linux está sendo utilizada, porque o kernel possui um número de versão e a sua distribuição pode possuir outro. O número que identifica a versão da distribuição é decidido pela empresa responsável pela versão, de acordo com padrões próprios. Já o kernel possui um número de versão composto por 3 partes: um número maior, um número menor e um número de release, e este número é dado pelo grupo de programadores que cuidam do desenvolvimento do kernel. O número de série do seu kernel pode ser facilmente identificado digitando-se no prompt o comando `uname -r`, como no exemplo abaixo:

```
# uname -r 2.6.17-10-generic
```

O exemplo diz que este kernel pertence a quarta release da série 2.6. Em alguns casos é adicionado também um número que identifica uma compilação feita por uma determinada distribuição (2.6.17-10-generic no exemplo, ou seja, é a compilação padrão, ou generic, da distribuição Ubuntu Linux Versão 6.10). O número menor é particularmente importante, pois números pares identificam versões de kernel testadas e consideradas estáveis, enquanto que números ímpares identificam versões de desenvolvimento, onde novos recursos estão sendo testados.

O kernel do Linux pode ser compilado[6] para se adequar melhor ao tipo de máquina e ao tipo de tarefa que essa máquina vai executar. Por exemplo, se o servidor precisa se comunicar com outras máquinas usando o protocolo IPX, o administrador poderá compilar o kernel com suporte a esse protocolo, ou, se não houver necessidade de usar um determinado tipo de placa de rede, é possível compilar o kernel sem suporte a essa placa, resultando assim em um kernel de menor tamanho.

[6] Compilação é o processo de transformação de um código-fonte em um programa executável.

Aplicações do Sistema

O kernel faz muito pouco sozinho, uma vez que ele só provê os recursos que são necessários para que outros programas sejam executados. Logo, é necessária a utilização de outros programas para implementar os vários serviços necessários ao sistema operacional.

Do "ponto de vista" do kernel, as aplicações do sistema, bem como qualquer outro programa, rodam no que é chamado "modo usuário", logo, a diferença entre aplicações de sistema e aplicações do usuário se dá pelo propósito de cada aplicação. Aplicações do sistema são necessárias para fazer o sistema funcionar, enquanto as aplicações do usuário são todos programas utilizados pelo usuário para realizar uma determinada tarefa (como um processador de texto, por exemplo).

Entre as aplicações de sistema pode-se citar o `init`, o `getty` e o `syslog`.

Aplicações do usuário

As aplicações do usuário são todas aquelas utilizadas pelo usuário para executar uma determinada tarefa. Editores de texto, editores de imagens, navegadores e leitores de email se encaixam nessa categoria.

O ambiente gráfico

No Linux a responsabilidade pelo ambiente gráfico não é do kernel e sim de um programa especial, o `XFree86`. No entanto, este programa provê apenas as funções de desenho de elementos gráficos e interação com a placa de vídeo. A interação final do usuário com a interface gráfica se dá através de programas gerenciadores de janelas, como o `KDE`, o `GNOME`, o `XFCE`, e são eles os responsáveis pela "aparência" do seu Linux.

A separação do ambiente gráfico do resto do sistema apresenta muitas vantagens. Como o ambiente gráfico consome recursos do sistema, é possível desativá-lo, principalmente em servidores, resultando assim em um melhor desempenho de outras aplicações, uma vez que a quantidade de processamento da CPU que seria utilizado para o `XFree86`, poderá ser utilizado para essas aplicações. Além do mais, o desenvolvimento do ambiente gráfico pode ocorrer de maneira independente ao do kernel.

O Linux também pode funcionar em modo texto. Nesse caso a interação com o usuário se dá por meio de um shell, como o `Bash`, que é capaz de interpretar e executar comandos digitados pelo usuário.

As Licenças no Mundo Linux

Antes de se definir em quais licenças e condições o Linux se encaixa, serão vistas as principais licenças utilizadas atualmente para os softwares em geral.

As licenças no mundo da Informática podem ser divididas, de um modo geral, em licenças de software proprietário e licenças de software livre. As licenças em softwares proprietários são geralmente comerciais e não permitem a cópia, modificação ou distribuição do software em questão.

O software livre é utilizado de um modo completamente diferente. Para que um software seja livre, ele deve possuir os seguintes itens:

- ✓ Liberdade para executar o software, seja qual for a sua finalidade.
- ✓ Liberdade para acessar o código-fonte do programa e modificá-lo conforme sua necessidade.
- ✓ Liberdade para fazer cópias e distribuí-las para quem desejar.
- ✓ Liberdade para melhorar o programa e distribuir suas melhorias ao público, de modo que elas fiquem disponíveis para a comunidade.

Com isso é definido qual o principal objetivo da Fundação do Software Livre (Free Software Foundation): promover a disseminação do software livre no mundo da Informática, eliminar restrições de cópias e distribuição de programas, entre outros pontos. Para mais informações, verifique o site da Free Software Foundation(<http://www.fsf.org>).

As licenças de software livre podem ser divididas, de modo geral, em dois grupos: licenças de documentação e licenças de software. A licença GNU GPL[7] é uma das mais conhecidas, e talvez uma das mais utilizadas como licença de software. A licença GPL foi criada para garantir que cópias de softwares livres possam ser distribuídas, alteradas ou utilizadas (na sua totalidade ou em parte) por novos programas. Um outro exemplo de licença é a GNU FDL[8], que segue a mesma linha da GPL, mas utilizada para a documentação. Existem muitas outras licenças, e o escopo destas pode variar muito.

Este assunto é muito extenso e abrange uma série de outras questões. Se você deseja conhecer mais detalhes sobre as licenças, sobre software livre ou sobre a Fundação do Software Livre, verifique a página do Projeto GNU(<http://www.gnu.org>).

[7] General Public License.

[8] Free Documentation License.

3. Como funciona o Linux

Para entendermos como o Linux inicia, temos de entender como um sistema operacional inicia.

O processo de inicialização (também conhecido por processo de boot) de um sistema linux é composto por várias etapas, sendo que a explicação a seguir é simplificada.

A primeira etapa é comum a outros sistemas operacionais: quando o computador é ligado, a BIOS efetua o POST (Power-On Self-Test) e, ao final do processo, procura um dispositivo que possa ser usado para inicializar o sistema operacional. Caso este dispositivo seja um disco rígido, a BIOS vai carregar as informações existentes no MBR (Master Boot Record), que é o primeiro setor do disco (cilindro 0, cabeça 0, setor 1). No MBR existirá um pequeno programa que se encarregará de iniciar a carga do sistema operacional. Eventualmente, este programa poderá passar o controle para um outro, armazenado em uma partição inicializável existente no disco.

Inicialização de uma distribuição Linux

Para carregar o Linux, é necessário instalar um gerenciador de inicialização no MBR ou na partição raiz do sistema, desde que esta esteja marcada como ATIVA. Este gerenciador irá, então, carregar o kernel. Ao final desta etapa, o kernel irá carregar o programa init, o qual se encarregará de iniciar os serviços necessários para o funcionamento do sistema.

Analogia com o DOS:

DOS	LINUX
io.sys	vmlinuz
msdos.sys	/etc/lilo.conf ou /boot/grub/menu.lst
config.sys	/etc/inittab
autoexec.bat	rc<runlevel>.d/...
--	login
command.com	bash

Assim como o DOS e o Windows, o Linux possui níveis de execução. Ou seja, ele possui opções de boot normal, em modo seguro, em modo texto e etc. A diferença é que no linux você pode personalizar, mais abaixo possui o que chamamos de padrão System V, herdado do Unix.

OBS: O init é o primeiro processo lançado após o carregamento do kernel na memória, e é ele o responsável por continuar o processo de boot lançando os outros programas. É o init o responsável, também, por garantir que o getty esteja sendo executado (para que os usuários possam entrar no sistema) e por adotar processos órfãos (processos filhos no qual o pai morreu), pois no Linux todos os processos devem estar em uma mesma árvore, e possuem um pai (excluindo o processo init, que não tem pai).

O `getty` provê o serviço responsável pelo login dos usuários em terminais textos (virtuais ou não). É ele que lê o nome do usuário e a senha e chama o programa `login`[9] para validá-los; caso estejam corretos é lançado um shell, caso contrário o processo todo é reiniciado.

O `syslog` é responsável por capturar as mensagens de erro geradas pelo kernel ou por outras aplicações de sistema, e por mostrá-las posteriormente quando o administrador do sistema solicitá-las.

Níveis de Execução

Os Níveis de execução (run levels) são diferentes modos de funcionamento do GNU/Linux com programas, daemons e recursos específicos. Em geral, os sistemas GNU/Linux possuem sete níveis de execução numerados de 0 a 6. O daemon `init` é o primeiro programa executado no GNU/Linux (veja através do `ps ax|grep init`) e responsável pela carga de todos daemons de inicialização e configuração do sistema.

O nível de execução padrão em uma distribuição GNU/Linux é definido através do arquivo de configuração do `/etc/inittab` através da linha:

```
id:2:initdefault:
```

Entendendo o funcionamento dos níveis de execução do sistema (runlevels)

Os nível de execução atual do sistema pode ser visualizado através do comando `runlevel` e modificado através dos programas `init` ou `telinit`. Quando é executado, o `runlevel` lê o arquivo `/var/run/utmp` e adicionalmente lista o nível de execução anterior ou a letra N em seu lugar (caso ainda não tenha ocorrido a mudança do nível de execução do sistema).

No Debian, os diretórios `/etc/rc0.d` a `/etc/rc6.d` contém os links simbólicos para arquivos em `/etc/init.d` que são acionados pelo nível de execução correspondente.

Por exemplo, o arquivo `S10sysklogd` em `/etc/rc2.d`, é um link simbólico para `/etc/init.d/sysklogd`. O que aconteceria se você removesse o arquivo `/etc/rc2.d/S10sysklogd` ?

Simplesmente o daemon `sysklogd` deixaria de ser executado no nível de execução 2 (runlevel padrão da Debian) do sistema.

O Debian segue o seguinte padrão para definir se um link simbólico em `/etc/rc[0-6].d` iniciará ou interromperá a execução de um serviço em `/etc/init.d`, que é o seguinte:

- Se um link é iniciado com a letra K (kill), quer dizer que o serviço será interrompido naquele nível de execução. O que ele faz é executar o daemon em `/etc/init.d` seguido de `stop`.
- Se um link é iniciado com a letra S (start), quer dizer que o serviço será iniciado naquele nível de execução (é equivalente a executar o daemon seguido de `start`).

[9] O kernel não se envolve no processo de login. Tudo é feito por outros programas.

É por isso que primeiro os links com a letra K são executados e depois os S. A ordem que os links são executados dependem do valor numérico que acompanha o link, por exemplo, os seguintes arquivos são executados em seqüência:

```
S10syslogd
S12kerneld
S20inetd
S20linuxlogo
S20logoutd
S20lprng
S89cron
S99xdm
```

Note que os arquivos que iniciam com o mesmo número (S20*) são executados alfabeticamente. O nível de execução do sistema pode ser modificado usando-se o comando `init` ou `telinit`. Os seguintes níveis de execução estão disponíveis na Debian:

- 0 - Interrompe a execução do sistema. todos os programas e daemons finalizados. É acionado pelo comando `shutdown -h`
- 1 - Modo monousuário, útil para manutenção dos sistema.
- 2 - Modo multiusuário (padrão da Debian)
- 3 - Modo multiusuário
- 4 - Modo multiusuário
- 5 - Modo multiusuário com login gráfico
- 6 - Reinicialização do sistema. Todos os programas e daemons são encerrados e o sistema é reiniciado. É acionado pelo comando `shutdown -r` e o pressionamento de CTRL+ALT+DEL.

Por exemplo, para listar o nível de execução atual do sistema digite: `runlevel`. O `runlevel` deverá listar algo como:

```
N 2
```

Agora para mudar para o nível de execução 1, digite: `init 3`. Agora confira a mudança digitando: `runlevel`. Você deverá ver este resultado:

```
2 3
```

Isto quer dizer que o nível de execução anterior era o 2 e o atual é o 3.

4. Estrutura de arquivos de diretórios

Numa distribuição Linux, temos uma estrutura de diretórios diferente de outros sistemas operacionais. Os arquivos ali dispostos, a princípio, parecem estar jogados aleatoriamente nos diversos diretórios existentes. Mas, felizmente, há uma certa ordem e uma lógica, fazendo com que cada diretório do sistema tenha uma finalidade específica.

Sistema de arquivos

Sistema de arquivos é a forma que o sistema operacional usa para representar determinada informação em um espaço de armazenagem. É o método de identificar e indexar as informações que estão armazenadas em qualquer mídia: disquetes, discos rígidos, drives em memória, CDs, etc.

Quando se prepara um disco para o trabalho através do processo de formatação física, criam-se os meios magnéticos necessários para armazenar os dados. Este processo faz uma preparação do dispositivo de armazenagem para que ele possa receber um sistema de arquivos e futuramente os dados do usuário. Um sistema de arquivos, portanto, é necessário para manter padrões, para controlar o tamanho das partições, permissões de arquivos, tamanho dos arquivos e sua organização, entre muitas outras funções.

Veremos primeiramente quais os principais tipos de sistemas de arquivos que existem e que são criados em um processo de formatação:

- **ext:** sistema de arquivos estendido (extended filesystem). É o sistema de arquivos mais utilizado no Linux. Ele fornece padrões para arquivos regulares, diretórios, arquivos de dispositivos e links simbólicos, entre outras características avançadas. Desenvolvedores trabalham para ampliar a capacidade deste sistema de arquivos para o suporte a controle de acesso conforme o padrão POSIX e compressão de arquivos on the fly (o arquivo é descompactado no momento em que é acessado). Suas principais ramificações são os sistemas ext2 e ext3.
- **swap:** sistema de arquivos para memória virtual. O tamanho pode variar. Ex.: 512MB
- **vfat:** este é o sistema de arquivos (volume FAT) dos sistemas Windows®9x e Windows NT®, usado também para formatar disquetes.
- **nfs:** sistema de arquivos de rede Unix, utilizado para acessar diretórios de máquinas remotas, que permite o compartilhamento de dados na rede.
- **smb:** sistema de arquivos de rede Windows, utilizado para acessar diretórios de máquinas remotas, que permite o compartilhamento de dados na rede.
- **reiserfs:** sistema de arquivos com suporte a características mais avançadas, como por exemplo melhor performance para diretórios muito grandes e suporte a transações (journaling).
- **iso9660:** sistema de arquivos do CD-ROM.
- **hpfs:** sistema de arquivos do OS/2®.

Existem muitos outros tipos de sistemas de arquivos, dependendo da sua aplicação e de que sistema você precisará. Examine as páginas de manual do `fstab` e do `mount` para mais detalhes sobre os vários tipos de sistema de arquivos.

Juntamente com o conceito de sistema de arquivos existe o conceito de partição. Particionar um dispositivo é dividi-lo de forma que cada uma das suas partes, denominadas partições, possam receber um tipo de sistema de arquivo e estejam preparadas para receber as informações.

Sistema de arquivos e partições são normalmente confundidos, quando na verdade são conceitos totalmente diferentes. As partições são áreas de armazenamento, criadas durante o processo de particionamento, sendo que cada partição funciona como se fosse um disco rígido (ou dispositivo utilizado). Para se utilizar uma partição, entretanto, deve-se criar um sistema de arquivos, ou seja, um sistema que organize e controle os arquivos e diretórios desta partição. Uma partição só pode ter apenas um sistema de arquivo.

Apesar da diferença, muitas vezes os termos são utilizados de modo similar. Não é uma linguagem totalmente errada, pois é muito utilizada diariamente, mas deve-se ter em mente a diferença entre os termos.

O sistema de arquivos mais importante em um sistema Linux é o sistema de arquivos raiz. Ele geralmente está montado no diretório "/", também chamado de diretório raiz. Um sistema de arquivos raiz deve conter o necessário para suportar um sistema Linux completo, e para tanto deve incluir alguns requisitos básicos:

- A estrutura básica do sistema de arquivos, geralmente agrupando os diretórios em árvores e obedecendo a alguns padrões.
- Conjunto mínimo de diretórios: /dev, /proc e /bin, entre outros.
- Conjunto básico de utilitários e comandos: ls, cp, mv.
- Conjunto mínimo de arquivos de configuração: rc, inittab, fstab.
- Arquivos de dispositivo: /dev/hd*, /dev/tty*, /dev/fd0.
- Bibliotecas que disponibilizem as funções básicas necessárias aos utilitários.

A estrutura dos sistemas de arquivos do Linux prevê um agrupamento que permite maior organização de dados, o que aumenta a funcionalidade do sistema. Os comandos estão todos em uma determinada área, todos os arquivos de dados em uma outra, documentação em uma terceira, e assim por diante. Além disso, o diretório raiz geralmente não contém nenhum arquivo, exceto, em algumas distribuições, pela imagem de inicialização do sistema. Todos os outros arquivos estão em subdiretórios do raiz.

Estrutura de diretórios

Após esta primeira apresentação, passamos agora a mostrar como é formado o diretório raiz de um sistema Linux. Veja a estrutura básica de diretórios abaixo:

```
/bin  
/dev  
/home  
/proc  
/usr  
/boot  
/etc  
/lib  
/var  
/sbin
```

- **"/"**, é específico de cada máquina. Pode ficar tanto em um disco físico quanto na memória da máquina ou em uma unidade de rede. É o diretório principal, que contém todos os arquivos e diretórios do sistema.
- **/bin** é o diretório que contém o mínimo de programas necessários para a inicialização e utilização do sistema.
- **/dev** é o local onde ficam armazenadas as referências aos dispositivos presentes na máquina, para o controle destes dispositivos. Esse diretório contém apontadores para, por exemplo, o drive de disquetes, os discos da máquina, terminais virtuais, portas de acesso seriais e paralelas, etc. Os controladores são automaticamente criados durante a instalação do sistema e posteriormente podem ser criados através do comando MAKEDEV.
- **/home** contém os diretórios pessoais dos usuários e suas configurações.
- **/proc** fornece informações sobre o kernel e sobre os processos que estão rodando no momento, além de informações sobre a utilização de alguns dispositivos. Alguns parâmetros do kernel podem ser alterados diretamente nesses arquivos, fazendo com que as modificações passem a valer imediatamente. Esse diretório não ocupa espaço nenhum em disco e as informações ali presentes são geradas apenas quando solicitadas.
- **/usr** contém comandos, bibliotecas, programas, páginas de manual e outros arquivos que não mudam mas que se fazem necessários para a operação normal do sistema. Como são estáticos, é interessante compartilhar esses arquivos pela rede, o que resulta numa grande economia de espaço em disco utilizado.
- **/boot** contém informações para o gerenciador de inicialização do sistema. É aqui que normalmente ficam as informações para o carregador do sistema operacional e também a imagem para a pré-carga do sistema (initrd).
- **/etc** é um dos mais importantes diretórios da máquina. Nele ficam a maioria dos arquivos de configuração e manipulação dos serviços essenciais ao sistema, a maioria dos arquivos de configuração de acesso a rede e de comunicação, arquivos de configuração do Sistema de Janelas X[1], arquivos de configuração do idioma do sistema, de atualizações, enfim, de muitas funcionalidades da máquina.
- **/lib** é o diretório onde ficam as bibliotecas básicas do sistema. Elas são compartilhadas por diversos programas, principalmente os que se encontram no diretório raiz.
- **/var** contém arquivos que possuem dados variáveis. Neste diretório estão arquivos e diretórios de spool, arquivos de log, arquivos de configuração de correio eletrônico e de news, entre outros. O diretório **/var** também guarda arquivos que precisam de uma freqüente atualização, como os arquivos de conteúdo do servidor Apache ou do servidor de FTP.
- **/sbin** contém ferramentas de uso do superusuário e que geralmente são usadas por serviços básicos da máquina. Ficam nesse diretório programas como os responsáveis pela carga de módulos do kernel, ativação e interrupção das interfaces de rede, manutenção dos sistemas de arquivos e de outras atividades.

Existem, além destes, alguns diretórios que também são importantes para a complementação da funcionalidade do sistema, e estão presentes em praticamente todos os sistemas Linux, apesar de não serem estritamente obrigatórios. São eles:

- **/mnt** é o diretório utilizado para o acesso a dispositivos de mídia, como disquetes e CDROM. Ele é utilizado como ponto de montagem para a maioria destes dispositivos.
- **/tmp** serve como repositório para arquivos temporários, sendo utilizado para programas que são executados após a ativação do sistema, ou seja, este diretório serve como espaço extra para vários programas e aplicações.
- **/opt** pode ser útil por questões de espaço ou programas opcionais

Apesar das diferentes partes acima serem chamadas de diretórios, não há obrigatoriedade que elas estejam separadas. Elas podem estar facilmente no mesmo sistema de arquivos em uma pequena máquina utilizada por um único usuário que deseje mantê-las de uma forma mais simplificada.

Alguns destes diretórios podem ser montados em suas próprias partições. Suponha que, por questões de espaço ou segurança, é desejável que o diretório dos usuários esteja em uma outra partição. Então, é criada uma outra partição que será montada no diretório /home e que terá, por exemplo, o mesmo sistema de arquivos utilizado pelo sistema de arquivos raiz. Essa separação em um sistema de arquivos à parte é interessante pois facilita determinadas tarefas administrativas, como gerenciamento da quantidade de espaço que cada usuário pode utilizar e a manutenção de cópias de segurança.

Portanto, a árvore de diretórios pode estar dividida ainda em diferentes sistemas de arquivos, dependendo do tamanho de cada disco e de quanto espaço será alocado para cada finalidade.

5. Comandos básicos do Sistema

DOS	LINUX	FUNÇÃO
cls	clear	Limpa tela
dir	ls	Lista arquivos e diretórios
cd	cd	Entra e sai de diretórios
md	mkdir	Cria diretório
del	rm	Apaga arquivo
rd	rm -r	Apaga diretório vazio
deltree	rm -rf	Apaga árvore de diretórios
copy	cp	Copia arquivo
xcopy	cp -r	Copia diretório / árvore
ren	mv	Renomeia arquivo ou diretório
move	mv	Move arquivo / diretório / árvore
type	cat	Mostra cont. de arq.
more	more / less	Mostra cont. de arq. paginando
edit	mcedit / nano / vi	Edita arquivo
ver	uname -r	Versão do Sistema
date	date	Mostra / Muda data
time	date	Mostra / Muda hora
telnet	telnet	Conecta em portas remotas
ftp	ftp	Cliente de ftp
hostname	hostname	Mostra / muda nome da máquina
ping	ping	Testa conectividade entre hosts
tracert	tracert	Traça o caminho de roteadores
netstat	netstat	Mostra status das portas locais

6. Instalação de uma distribuição

1. Baixe o CD de instalação do Ubuntu Desktop Linux
<http://espelhos.edugraf.ufsc.br/ubuntu-releases/edgy/ubuntu-6.10-desktop-i386.iso>
2. Isto é uma imagem de CD iso9660. Você precisa gravar esta imagem num CD. Se não sabe como se faz isso, entre num site de procura e digite “gravar imagem iso CD”.
3. Dê boot com este CD gravado.
4. Após entrar no sistema é só clicar no Ícone Install e seguir as instruções.

OBS: Se você não sabe exatamente como se faz essa instalação, chame um técnico. É melhor chamar alguém que conheça do que você perder os seus dados.

Bibliografia

<http://focalinux.cipsga.org.br>

<http://help.ubuntu.com/>