

# Introdução ao Linux

## Um Guia Prático

Machtelt Garrels

Garrels.be

[<tille\\_wants\\_no\\_spam\\_at\\_garrels\\_dot\\_be>](mailto:tille_wants_no_spam_at_garrels_dot_be)

Este guia foi traduzido para o português por

**André Nunes**

tagesuhu.wordpress.com

[<andrenbatista\\_at\\_gmail\\_dot\\_com>](mailto:andrenbatista_at_gmail_dot_com)

Copyright © 2002, 2003, 2004, 2005, 2006, 2007, 2008 Machtelt Garrels

20080606

---

### Table of Contents

#### [Introdução](#)

- [Por que este guia?](#)
- [Quem deveria ler este livro?](#)
- [Novas versões e disponibilidade](#)
- [Histórico de revisões](#)
- [Contribuições](#)
- [Retorno](#)
- [Informações de direitos autorais](#)
- [O que você precisa?](#)
- [Convenções usadas neste documento](#)
- [Organização deste documento](#)

#### [1. O que é o Linux?](#)

##### [História](#)

- [UNIX](#)
- [Linus e Linux](#)
- [O atual uso dos sistemas Linux](#)

##### [A interface com o usuário](#)

- [O Linux é difícil?](#)
- [Linux para usuários inexperientes](#)

##### [O Linux tem futuro?](#)

- [Open Source](#)
- [Dez anos de experiência a seu dispor](#)

##### [Propriedades do Linux](#)

- [Os prós do Linux](#)
- [Os contras do Linux](#)

##### [Os Sabores do Linux](#)

- [O Linux e o GNU](#)
- [O GNU/Linux](#)
- [Qual distribuição devo instalar?](#)

##### [Resumo](#)

##### [Exercícios](#)

#### [2. Início Rápido](#)

[Como efetuar o login, ativar a interface de usuário e efetuar o logout](#)

##### [Introdução](#)

[Modo Gráfico](#)  
[Modo Texto](#)

## [O básico do básico](#)

[Os comandos](#)  
[Observações](#)  
[Usando os recursos Bash](#)

## [Buscando ajuda](#)

[Esteja Ciente](#)  
[As páginas man](#)  
[Mais informações](#)

## [Resumo](#) [Exercícios](#)

[Conectando e desconectando](#)  
[Senhas de acesso](#)  
[Diretórios](#)  
[Arquivos](#)  
[Buscando ajuda](#)

## [3. Dos arquivos e do sistema de arquivos](#)

### [Panorama geral do sistema de arquivos do Linux](#)

[Arquivos](#)  
[Do particionamento](#)  
[Mais sobre a disposição do sistema de arquivos](#)

### [Orientação no sistema de arquivos](#)

[O caminho \(path\)](#)  
[Caminhos absolutos e relativos](#)  
[Os arquivos e diretórios mais importantes](#)  
[Os arquivos de configuração mais importantes](#)  
[Os dispositivos mais comuns](#)  
[Os arquivos de variáveis mais comuns](#)

### [Manipulando arquivos](#)

[Exibindo as propriedades de um arquivo](#)  
[Criando e deletando arquivos e diretórios](#)  
[Encontrando arquivos](#)  
[Outros modos de averiguar o conteúdo de arquivos](#)  
[Interligando arquivos](#)

### [Segurança de arquivos](#)

[Direitos de acesso: A primeira linha de defesa do Linux](#)  
[As ferramentas](#)

## [Resumo](#) [Exercícios](#)

[Partições](#)  
[Caminhos](#)  
[Passeio pelo sistema](#)  
[Manipulando arquivos](#)  
[Permissões de arquivo](#)

## [4. Processos](#)

### [Processos virados do avesso](#)

[Multiusuário e multitarefa](#)  
[Tipos de processos](#)  
[Atributos de processos](#)  
[Exibindo informações de processos](#)  
[Vida e morte de um processo](#)  
[SUID and SGID](#)

### [Processo de inicialização, iniciar e desligar](#)

[Introdução](#)  
[O processo de inicialização](#)  
[Recursos do GRUB](#)  
[Init](#)  
[Níveis de execução do init](#)

[Desligamento](#)

[Gerenciando processos](#)

[Trabalho para o administrador do sistema](#)

[Quanto tempo leva?](#)

[Performance](#)

[Carga](#)

[Eu posso fazer alguma coisa como um usuário?](#)

[Agendando processos](#)

[Use aquele tempo ocioso!](#)

[O comando sleep](#)

[O comando at](#)

[Cron e crontab](#)

[Resumo](#)

[Exercícios](#)

[Geral](#)

[Inicialização, init etc.](#)

[Agendamento](#)

## [5. Redirecionamento de E/S](#)

[Redirecionamentos simples](#)

[O que são entrada e saída padrão?](#)

[Os operadores de redirecionamento](#)

[Recursos avançados de redireção](#)

[Uso dos descritores de arquivo](#)

[Exemplos](#)

[Filtros](#)

[Mais sobre o grep](#)

[Filtrando a saída](#)

[Resumo](#)

[Exercícios](#)

## [6. Editores de texto](#)

[Editores de texto](#)

[Por que eu deveria usar um editor?](#)

[Qual editor devo usar?](#)

[Usando o editor Vim](#)

[Dois modos](#)

[Comandos básicos](#)

[O modo fácil](#)

[O Linux no escritório](#)

[História](#)

[Suítes e programas](#)

[Observações](#)

[Resumo](#)

[Exercícios](#)

## [7. Lar doce /lar \(/home\)](#)

[Boa manutenção do lar](#)

[Introdução](#)

[Crie espaço](#)

[Seu ambiente texto](#)

[Variáveis do ambiente](#)

[Arquivos de configuração do shell](#)

[Um conjunto típico de arquivos de configuração](#)

[O prompt do Bash](#)

[Scripts do shell](#)

[O ambiente gráfico](#)

[Introdução](#)  
[O Sistema de Janelas X](#)  
[Configuração do servidor X](#)

## [Configurações de Região](#)

[Configuração de teclado](#)  
[Fontes](#)  
[Data e Fuso Horário](#)  
[Idioma](#)  
[Informações específicas do país](#)

## [Instalando novos programas](#)

[Geral](#)  
[Formatos de pacotes](#)  
[Automatizando o gerenciamento de pacotes e atualizações](#)  
[Atualizando o seu kernel](#)  
[Instalando pacotes extras a partir do CD de instalação](#)

[Resumo](#)  
[Exercícios](#)

[Ambiente shell](#)  
[Ambiente gráfico](#)

## 8. Impressoras e impressão

### [Imprimir arquivos](#)

[Imprimir a partir da linha de comando](#)  
[Formatação](#)

### [O lado do servidor](#)

[Geral](#)  
[Configuração gráfica da impressora](#)  
[Comprando uma impressora para o Linux](#)

### [Problemas de impressão](#)

[Arquivo errado](#)  
[Minha impressão não saiu](#)

[Resumo](#)  
[Exercícios](#)

## 9. Técnicas Fundamentais de Backup

### [Introdução](#)

[Preparando seus dados](#)

### [Movendo seus dados para um dispositivo de backup](#)

[Criando uma cópia em um disquete](#)  
[Criando uma cópia com o gravador de CDs](#)  
[Backups em/a partir de jazz drives, dispositivos USB e outros removíveis](#)  
[Fazendo backup de dados com um dispositivo de fita](#)  
[Ferramentas para a sua distribuição](#)

### [Usando o rsync](#)

[Introdução](#)  
[Um exemplo: rsync em um dispositivo de armazenamento USB](#)

### [Criptografia](#)

[Notas gerais](#)  
[Gerando uma chave](#)  
[Sobre a sua chave](#)  
[Criptografando seus dados](#)  
[Descriptografando arquivos](#)

[Resumo](#)  
[Exercícios](#)

## 10. Redes

### [Panorama de Redes](#)

[O modelos OSI](#)  
[Alguns protocolos populares de rede](#)

## [Configurações e informações sobre a rede](#)

[Configuração de interfaces de rede](#)  
[Arquivos de configuração de rede](#)  
[Comandos de configuração de rede](#)  
[Nomes das interfaces de rede](#)  
[Verificando a configuração da máquina com o nstat](#)  
[Outras máquinas](#)

## [Aplicativos Internet/Intranet](#)

[Tipos de servidores](#)  
[Email](#)  
[Web](#)  
[File Transfer Protocol \(Protocolo de Transferência de Arquivos\)](#)  
[Conferências e Bate-papo](#)  
[Serviços de notícias](#)  
[O Domain Name System](#)  
[DHCP](#)  
[Serviços de autenticação](#)

## [Execução remota de aplicativos](#)

[Introdução](#)  
[Rsh, rlogin e telnet](#)  
[O Sistema de Janelas X](#)  
[O suíte SSH](#)  
[VNC](#)  
[O protocolo rdesktop](#)  
[Cygwin](#)

## [Segurança](#)

[Introdução](#)  
[Serviços](#)  
[Atualize o sistema com regularidade](#)  
[Firewalls e política de acesso](#)  
[Detecção de invasões](#)  
[Mais dicas](#)  
[Eu fui invadido?](#)  
[Recuperando-se de uma invasão](#)

## [Resumo](#)

### [Exercícios](#)

[Redes em geral](#)  
[Conexões remotas](#)  
[Segurança](#)

## [11. Som e vídeo](#)

### [Áudio Básico](#)

[Instalação](#)  
[Driver e arquitetura](#)

### [Execução de som e vídeo](#)

[Execução e cópia de CDs](#)  
[Tocando arquivos de música](#)  
[Gravação](#)

### [Execução de vídeo, fluxos e televisão](#) [Telefonia de Internet](#)

[O que é?](#)  
[O que você vai precisar?](#)

## [Resumo](#)

### [Exercícios](#)

## [A. Para aonde ir a partir daqui?](#)

### [Livros úteis](#)

[Linux em geral](#)  
[Editores](#)  
[Shells](#)

[X Window](#)  
[Redes](#)

[Sites úteis](#)

[Informações gerais](#)  
[Referências por arquitetura](#)  
[Distribuições](#)  
[Software](#)

[B. Comandos DOS vs comandos Linux](#)

[C. Recursos do Shell](#)

[Recursos comuns](#)  
[Recursos específicos](#)

[Glossário](#)

[Index](#)

## List of Figures

1. [Capa do Introdução ao Linux](#)
- 1.1. [Planilha de cálculo do OpenOffice \(compatível com o MS Excell\)](#)
- 2.1. [Janela de terminal](#)
- 2.2. [Konqueror como um navegador de ajuda](#)
- 3.1. [Disposição do sistema de arquivos no Linux](#)
- 3.2. [Mecanismos dos links rígidos e simbólicos](#)
- 4.1. [Mecanismo bifurcar-e-executar](#)
- 4.2. [Você não pode ir mais rápido?](#)
- 4.3. [Gnome System Monitor](#)
- 8.1. [Status da Impressora através da interface web](#)
- 9.1. [Floppy formatter](#)
- 10.1. [O leitor de emails e notícias Evolution](#)
- 10.2. [X-Chat](#)
- 10.3. [SSH X11 forwarding](#)
- 11.1. [O tocador mp3 XMMS](#)

## List of Tables

1. [Convenções tipográficas e de uso](#)
- 2.1. [Comandos básicos](#)
- 2.2. [Combinação de teclas no Bash](#)
- 2.3. [Novos comandos no capítulo 2: o Básico](#)
- 3.1. [Tipos de arquivos em uma lista longa](#)
- 3.2. [Subdiretórios do diretório raiz](#)
- 3.3. [Os arquivos de configuração mais comuns](#)
- 3.4. [Dispositivos comuns](#)
- 3.5. [Esquema padrão do ls-em-cores](#)
- 3.6. [Esquema de sufixos padrão para o ls](#)
- 3.7. [Códigos de modo de acesso](#)
- 3.8. [Códigos de grupos de usuários](#)
- 3.9. [Proteção de arquivos com o chmod](#)
- 3.10. [Novos comandos no capítulo 3: Arquivos e o sistema de arquivos](#)
- 3.11. [Permissões de arquivo](#)
- 4.1. [Controlando processos](#)
- 4.2. [Sinais comuns](#)
- 4.3. [Novos comandos no capítulo 4: Processos](#)
- 5.1. [Novos comando no capítulo 5: Redirecionamento E/S](#)
- 7.1. [Variáveis de ambiente comuns](#)
- 7.2. [Novos comandos no capítulo 7: Fazendo-se sentir em casa](#)
- 8.1. [Novos comandos no capítulo 8: Impressão](#)
- 9.1. [Novos comandos no capítulo 9: Backup](#)
- 10.1. [O modelo OSI simplificado](#)
- 10.2. [Novos comandos no capítulo 10: Redes](#)
- 11.1. [Novos comandos no capítulo 11: Áudio](#)
- B.1. [Panorama dos comandos ODS/Linux](#)
- C.1. [Recursos comuns do Shell](#)
- C.2. [Recursos específicos do shell](#)

# Introdução

## Table of Contents

[Por que este guia?](#)  
[Quem deveria ler este livro?](#)  
[Novas versões e disponibilidade](#)  
[Histórico de revisões](#)  
[Contribuições](#)  
[Retorno](#)

## Por que este guia?

Muitas pessoas ainda acreditam que aprender Linux é difícil ou que apenas peritos podem entender como um sistema Linux funciona. Embora exista muita documentação livre à disposição, ela se encontra espalhada pela Web e é muitas vezes confusa, vez que em geral é destinada a usuários experientes do UNIX e do Linux. Hoje, graças aos avanços em desenvolvimento, o Linux cresceu em popularidade tanto em casa quanto no trabalho. O objetivo deste guia é mostrar às pessoas de todas as idades que o Linux pode ser simples e divertido, bem como usado para os mais diversos propósitos

## Quem deveria ler este livro?

Este guia foi criado para apresentar um panorama do Sistema Operacional Linux, focado em usuários novatos, como um tour de exploração e um guia de primeiros passos, com exercícios ao final de cada capítulo. Para trainees mais avançados ele pode servir de referência, uma coleção de conhecimentos básicos necessários à administração do sistema e da rede. Este livro contém muitos exemplos reais, provenientes da experiência pessoal da autora como administradora, professora e consultora. Esperamos que estes exemplos o ajudem a ter um melhor entendimento do sistema Linux e que você se sinta encorajado a tentar novas coisas por conta própria.

Todos que quiserem ter uma “CLUE”, a Command Line User Experience, com o Linux e (o UNIX em geral) acharam este livro útil.

## Novas versões e disponibilidade

Este documento é publicado na seção de Guias da coleção do Linux Documentation Project no <http://www.tldp.org/guides.html>; lá também é possível baixar as versões formatadas em PDF e Postscript.

A versão mais recente encontra-se disponível em <http://tille.garrels.be/training/tldp/>.

A segunda edição deste guia está disponível impressa no [Fultus.com Books](#) no formato capa mole e é impresso de acordo com a demanda. A Fultus [distribui este documento](#) por meio da Ingram and Baker & Taylor para muitas livrarias, incluindo [Amazon.com](#), [Amazon.co.uk](#), [BarnesAndNoble.com](#) e [Google's Froogle](#) portal de compras global [Google Book Search](#).

### Figure 1. Capa do Introdução ao Linux



Este guia foi [traduzido ao Hindi](#) por:

- Alok Kumar
- Dhananjay Sharma
- Kapil
- Puneet Goel
- Ravikant Yuyutsu

Andrea Montagner [traduziu o guia para o italiano](#).

## Histórico de revisões

### Revision History

Revision 1.27	20080606	MG
atualizações.		
Revision 1.26	20070919	MG
Comentários dos leitores, licença.		
Revision 1.25	20070511	MG
Comentários dos leitores, pequenas atualizações, etiqueta de emails, informações atualizadas sobre a disponibilidade (graças a Oleg).		
Revision 1.24	2006-11-01	MG
termos de indexação adicionados, preparados para a segunda edição impressa, adicionadas informações sobre o gpg e proxy.		
Revision 1.23	2006-07-25	MG e FK
Atualizações e correções, nova remoção do app5, licença atualizada para permitir a inclusão na documentação do Debian.		
Revision 1.22	2006-04-06	MG

chap8 completamente revisado, chap10: exemplos mais claros, adicionada info sobre o ifconfig e o cygwin, aplicativos de rede revisados.

Revision 1.21 2006-03-14 MG

Exercícios adicionados no chap11, corrigidos erros de nova linha, panorama de comandos completada no capítulo 9, pequenas correções no capítulo 10.

Revision 1.20 2006-01-06 MG

Divisão do chap7: a parte de áudio agora está em um capítulo próprio, chap11.xml. Pequenas revisões, atualizações para comandos como o aptitude, mais sobre armazenamento USB, telefonia por Internet, correções dos leitores.

Revision 1.13 2004-04-27 MG

Última releitura antes do envio para a Fultus para impressão. Adicionada referência à Fultus na seção de Novas Versões, seções de Convenções e Organização atualizadas. Pequenas alterações nos capítulos 4, 5, 6 e 8, adicionada informação sobre sobre o rdesktop no capítulo 10, atualização do glossário, substituição das referências aos fileutils para o coreutils, um obrigado aos tradutores para o Hindi.

## Contribuições

Meus agradecimentos a todos aqueles que compartilharam suas experiências. E especialmente aos usuários Belgas do Linux, por me ouvirem todos os dias e por sua generosidade nos comentários.

Também, um agradecimento especial para Tabatha Marshall por ter feito uma revisão cuidadosa, verificando a ortografia e a formatação, bem como a Eugene Crosser, por captar os erros que nós duas deixamos passar.

E obrigado a todos os leitores que me apontaram assuntos não abordados e que perceberam os últimos erros, definições vagas e erros de digitação, tendo o trabalho de me enviar por email todas as suas considerações. Estas são também as pessoas que me auxiliam a manter este guia atualizado, como Filipus Klutiero que fez uma revisão completa em 2005 e 2006 e me ajuda a colocar o guia na coleção de documentos do Debian ou Alexey Eremenko que me enviou a base para o capítulo 11.

Em 2006, Suresh Rajashekara criou um pacote Debian para esta documentação.

Por fim, um muito obrigado aos voluntários que estão atualmente traduzindo este documento para o Francês, Sueco, Alemão, Farsi, Hindu e mais. É um grande trabalho que não deve ser menosprezado; eu admiro sua coragem.

## Retorno

Falta de informações, links ou caracteres? Envie um email para a mantenedora deste documento:

[<tille.wants.no.spam.at.garrels.be>](mailto:tille.wants.no.spam.at.garrels.be)

Não se esqueça de verificar a [versão mais recente!](#)

## Informações de direitos autorais

- \* Copyright (c) 2002-2007, Machtelt Garrels
- \* Todos os direitos reservados.
- \* A redistribuição e o uso nos formatos fonte e binário, com ou sem
- \* modificações, são permitidos, desde que aceitas as seguintes condições:
- \*
  - \* A redistribuição do código fonte deve conter a nota de direitos autorais acima, esta lista de condições e a cláusula de isenção de responsabilidade.
  - \* A redistribuição em binário deve conter a nota de direitos autorais acima, esta lista de condições e a cláusula de isenção de responsabilidade na documentação e/ou outros materiais inclusos na distribuição.
  - \* O nome da autora, Machtelt Garrels, ou dos contribuidores não poderão ser usados para a promoção de produtos derivados deste software sem permissão prévia e por escrito.
- \*  
\* ESTE SOFTWARE É FORNECIDO PELA AUTORA E CONTRIBUIDORES "COMO ESTÁ" E NEGA-SE
- \* QUAISQUER GARANTIAS, EXPRESSAS OU IMPLÍCITAS, INCLUÍDAS, PORÉM NÃO LIMITADAS A, AS
- \* GARANTIAS IMPLÍCITAS DE COMERCIALIZABILIDADE E ADEQUAÇÃO A UM DADO PROPÓSITO.
- \* EM NENHUMA HIPÓTESE SERÃO OS AUTORES OU CONTRIBUIDORES RESPONSÁVEIS POR QUAISQUER
- \* DANOS, DIRETOS, INDIRETOS, INCIDENTAIS, ESPECIAIS, EXEMPLARES OU CAUSAIS
- \* (INCLUSOS, PORÉM NÃO LIMITADOS A, DISPONIBILIDADE DE BENS OU SERVIÇOS SUBSTITUTOS;
- \* PERDA DE UTILIDADE, DADOS OU LUCROS; OU INTERRUPÇÃO DE NEGÓCIOS) QUAISQUER QUE SEJAM
- \* SUAS CAUSAS E QUALQUER QUE SEJA A TEORIA DA RESPONSABILIDADE, CONTRATUAL, ESTRITA,
- \* ILÍCITO CIVIL (INCLUINDO A NEGLIGÊNCIA OU ASSEMELHADOS), ADVINDOS DE QUALQUER FORMA
- \* DESDE SOFTWARE, MESMO QUANDO ALERTADA A POSSIBILIDADE DE TAL DANO.

Os logos, marcas e símbolos usados neste livro são de propriedade de seus titulares.

## O que você precisa?

Será necessário um computador e uma mídia contendo uma distribuição Linux. A maior parte deste guia é



aplicável a todas as distribuições Linux - e ao UNIX em geral. Além do tempo, não há quaisquer outras exigências.

O [Installation HOWTO](#) contém informações úteis de como obter softwares Linux e instalá-los no seu computador. Requisitos de hardware e a coexistência com outros sistemas operacionais também é discutida.

Imagens de CDs podem ser baixadas no [linux-iso.com](http://linux-iso.com) e em muitos outros locais, veja [Appendix A, Para onde ir a partir daqui?](#).

Uma alternativa interessante para aqueles que não ousam instalar o Linux em suas máquinas é o uso de uma distribuição a partir de um CD, como a distribuição [Knoppix](#).

## Convenções usadas neste documento

As seguintes convenções foram adotadas neste documento:

**Table 1. Convenções tipográficas e de uso**

Tipo de texto	Significado
“Texto citado”	Citações de pessoas ou saída do computador.
visão do terminal	Entrada literal para o computador e saída capturada no terminal, em geral apresentada com um fundo cinza claro.
<b>comando</b>	Nome de um comando que pode ser informado na linha de comando.
VARIÁVEL	Nome de uma variável ou ponteiro para o conteúdo de uma variável, como em \$VARIABLE.
opção	Opção a um comando, como na “opção -a ao comando <b>ls</b> ”.
argumento	Argumento para um comando, como em “read <b>man ls</b> ”.
prompt	Prompt do usuário, em geral seguido de um comando que se informa em uma janela de terminal, como em <code>hilda@home&gt;ls -l</code>
command options arguments	Sinopse de um comando ou uso geral, em uma linha separada.
nome do arquivo	Nome de um arquivo ou diretório, por exemplo “Mudar para o diretório /usr/bin.”
<b>Key</b>	Teclas para se pressionar no teclado, como “digitar <b>Q</b> para sair”.
Botão	Botão gráfico a ser clicado, como o botão OK.
Menu → Escolha	Escolha em um menu gráfico, como, por exemplo: “SelecionarAjuda → Sobre o Mozilla no seu navegador.”
Terminologia	Termo ou conceito importante: “O <i>kernel</i> do Linux é o coração do sistema.”
\	A barra invertida em uma visão de terminal ou sinopse de comando indica uma linha não acabada. Em outras palavras, se você vir um comando longo que está dividido em várias linhas, \ significa “Não aperte <b>Enter</b> ainda!”
Veja <a href="#">Chapter 1, O que é o Linux?</a>	link para assuntos relacionados neste guia.
<a href="#">O autor</a>	Link clicável para uma fonte externa na web.

As imagens a seguir são usadas:

### Isto é uma nota

Ela contém informações ou observações adicionais.

### Isto é um aviso

Significa que é preciso tomar cuidado.

### Isto é uma advertência

Tenha *muito* cuidado.

### Isto é uma dica

Dicas e truques.

## Organização deste documento

Este guia é parte do Linux Documentation Project e seu objetivo é ser a base para todos os outros materiais que se pode obter do Projeto. Desta forma, ele fornece os conhecimentos fundamentais para qualquer um que queira começar a trabalhar em um sistema Linux, ao mesmo tempo em que tenta evitar reinventar a água quente. Assim, você deve esperar que este livro seja incompleto e cheio de links para outras fontes de informações complementares no seu sistema, na Internet e na documentação do seu sistema.

Este primeiro capítulo é uma introdução ao assunto Linux; os dois subseqüentes discutem os comandos mais elementares. Os capítulos 4 e 5 discutem alguns assuntos mais avançados, porém ainda básicos. O capítulo 6 é necessário para acompanhar os demais, vez que discute a edição de arquivos, uma habilidade necessária para tornar-se um usuário Linux. Os capítulos que se seguem discutem alguns assuntos um pouco mais avançados que você terá que lidar na sua rotina de uso do Linux.

Todos os capítulos vêm com exercícios que testam seu preparo para o capítulo subseqüente.

- [Chapter 1, O que é o Linux?](#): O que é o Linux, como ele veio a existir, suas vantagens e desvantagens, o que o futuro reserva para ele, quem deve usá-lo, como instalá-lo em seu computador.
- [Chapter 2, Início Rápido](#): Primeiros passos, conectando-se ao sistema, comandos básicos, onde encontrar ajuda.
- [Chapter 3, Dos arquivos e do sistema de arquivos](#): O sistema de arquivos, arquivos e diretórios importantes, gerenciamento de arquivos e diretórios e proteção dos seus dados.
- [Chapter 4, Processos](#): Entendendo e gerenciando processos, procedimentos de inicialização e desligamento, agendamento de tarefas, tarefas repetitivas.
- [Chapter 5, Redirecionamento de E/S](#): Quais são os padrões de entrada, saída e erro e como estes recursos são usados na linha de comando.
- [Chapter 6, Editores de texto](#): Por que você deveria aprender a trabalhar com um editor, uma discussão sobre os editores mais comuns.
- [Chapter 7, Lar doce /lar \(/home\)](#): Configuração dos seus ambientes gráfico, texto e de áudio, configurações para os usuários de outros idiomas que não o inglês, dicas para a adição de outros softwares.
- [Chapter 8, Impressoras e impressão](#): Conversão de arquivos para um formato de impressão, imprimindo-os, dicas para a solução de problemas de impressão.
- [Chapter 9, Técnicas Fundamentais de Backup](#): Preparando os dados para backup, discussão de diversas ferramentas, backup remoto.
- [Chapter 10, Redes](#): Panorama das ferramentas de rede do Linux e aplicativos de usuário, com uma pequena discussão sobre os programas daemons por detrás, bem como redes seguras.
- [Chapter 11, Som e vídeo](#): Vídeo e som, incluindo Voz sobre IP e gravação de som, são discutidos neste capítulo.
- [Appendix A, Para aonde ir a partir daqui?](#): Quais livros ler e sites visitar quando você tiver terminado de ler este.
- [Appendix B, Comandos DOS vs comandos Linux](#): Uma comparação.
- [Appendix C, Recursos do Shell](#): Se você algum dia ficar perdido, estas tabelas podem ser uma saída. Servem também de argumento quando o seu chefe insistir que VOCÊ deveria usar o shell preferido DELE.

## Chapter 1. O que é o Linux?

### Table of Contents

#### [História](#)

- [UNIX](#)
- [Linus e Linux](#)
- [O atual uso dos sistemas Linux](#)

#### [A interface com o usuário](#)

- [O Linux é difícil?](#)
- [Linux para usuários inexperientes](#)

#### [O Linux tem futuro?](#)

- [Open Source](#)
- [Dez anos de experiência a seu dispor](#)

#### [Propriedades do Linux](#)

- [Os prós do Linux](#)
- [Os contras do Linux](#)

#### [Os Sabores do Linux](#)

- [O Linux e o GNU](#)

## Abstract

Começaremos com um panorama de como o Linux se tornou o sistema operacional que é hoje. Discutiremos seu desenvolvimento passado e futuro e daremos uma olhada de perto nas vantagens e desvantagens do sistema. Trataremos das distribuições, do Open Source em geral e tentaremos explicar um pouco do que é o GNU.

Este capítulo responde questões como:

- O que é o Linux?
- Onde e como o Linux começou?
- O Linux não é um sistema onde tudo é feito no modo texto?
- O Linux tem futuro ou é apenas uma moda?
- Quais são as vantagens de usar o Linux?
- Quais são as desvantagens?
- Quais tipos de Linux existem e como eu escolho o mais adequado às minhas necessidades?
- O que são os movimentos do Open Source e do GNU?

## História

### UNIX

Para entender a popularidade do Linux, é necessário retroceder no tempo, cerca de 30 anos atrás...

Imagine computadores grandes como casas, até mesmo estádios. Embora o tamanho daqueles computadores fosse a causa de grandes problemas, uma questão específica os tornava ainda piores: cada um deles tinha um sistema operacional diferente. Os softwares eram sempre adequados à execução de uma tarefa específica e os softwares de um sistema não podiam ser executados em outros. Ser capaz de trabalhar com um sistema não implicava automaticamente ser capaz de lidar com outro. Era difícil, tanto para os usuários, como para os administradores.

Os computadores eram extremamente caros à época e eram necessários sacrifícios mesmo após a compra, apenas para fazer com que os usuários entendessem como eles funcionavam. O custo global de cada unidade era enorme.

Em termos tecnológicos, o mundo não era muito avançado, então tivemos que conviver com o tamanho por mais uma década. Em 1969, uma equipe de desenvolvedores nos laboratórios da Bell Labs começou a trabalhar em uma solução para o problema dos softwares, enfrentando os problemas de compatibilidade. Eles desenvolveram um novo sistema operacional, o , que era

1. Simples e elegante.
2. Escrito na linguagem de programação C, ao invés de código assembly.
3. Capaz de reciclar códigos.

Os desenvolvedores da Bell Labs nomearam seu projeto "UNIX".

O recurso de reciclagem de códigos foi muito importante. Até então, todos os sistemas operacionais disponíveis eram escritos em códigos especificamente direcionados a um sistema. O UNIX, por outro lado, somente fazia uso de uma pequena porção deste código especial, que hoje é ordinariamente denominado kernel. O kernel é o único trecho de código que precisa ser adaptado para cada sistema específico, é o núcleo do sistema UNIX. O sistema operacional e todas as demais funções foram construídos em torno deste kernel e escritos em uma linguagem de programação de mais alto nível, C. Esta linguagem foi especialmente desenvolvida para criar o sistema UNIX. Empregando esta nova técnica, foi muito mais fácil desenvolver um sistema operacional que fosse compatível com tipos diferentes de hardware.

Os vendedores de software adaptaram-se rapidamente, uma vez que com isso podiam vender dez vezes mais softwares quase sem esforço. Novas e estranhas situações então começaram a aparecer: imagine, por exemplo, computadores de fabricantes diferentes comunicando-se em uma mesma rede ou usuários trabalhando em sistemas diferentes sem a necessidade de uma reeducação para usar o computador novo. O UNIX, em grande parte, foi o responsável por tornar os usuários compatíveis com sistemas diferentes.

Ao longo das duas décadas seguintes, o desenvolvimento do UNIX teve continuidade. Outras tarefas foram viabilizadas e mais comerciantes de hardware e software passaram a dar suporte em seus produtos para o UNIX.

Inicialmente, só era possível encontrar o UNIX em enormes ambientes com mainframes e minicomputadores (note que um PC é um “micro” computador). Era necessário trabalhar para uma universidade, para o governo ou para uma grande corporação para ter contato com um sistema UNIX.

Todavia, computadores menores vinham sendo desenvolvidos e, próximo ao final dos anos oitenta, muitas pessoas já tinham computadores em suas casas. À época, existiam diversas versões do UNIX disponíveis para a arquitetura dos PCs, mas nenhuma delas era verdadeiramente livre e, principalmente, todas eram extremamente lentas, de modo que a maior parte das pessoas usava o MS DOS ou o Windows 3.1 em seus PCs domésticos.

## Linus e Linux

No começo da década de 1990, os PC's domésticos finalmente tornaram-se suficientemente poderosos para executarem um UNIX. Linus Torvalds, um jovem estudante de ciências da computação da Universidade de Helsinque, percebeu que seria ótimo se existisse uma versão acadêmica, livremente disponível, do UNIX e imediatamente começou a programá-la.

Ele começou a levantar questões, em busca de respostas e soluções que iriam ajudá-lo a ter um UNIX em seu PC. Abaixo, lê-se um de seus primeiros comentários no comp.os.minix, datado de 1991:

De: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)

Grupo: comp.os.minix

Assunto: Gcc-1.40 and a posix-question

Mensagem: <1991Jul3.100050.9886@klaava.Helsinki.FI>

Data: 3 Jul 91 10:00:50 GMT

Olá habitantes da net,

Por conta de um projeto em que estou trabalhando (no minix), estou interessado na definição padrão posix. Será que alguém poderia me indicar (preferencialmente) em um formato legível por máquina um local onde encontrar as regras posix mais recentes? Sites FTP seriam ótimos

Desde o início, o objetivo de Linux era um sistema livre que fosse completamente compatível com o UNIX original. Por esta razão ele precisava dos padrões POSIX, uma vez que estes são os padrões do UNIX.

À época, o “plug-and-play” ainda não havia sido inventado, mas tantas pessoas estavam interessadas em ter um sistema UNIX próprio, que este era apenas um pequeno obstáculo. Novos drivers foram sendo disponibilizados para todos os tipos de hardware, a uma velocidade sempre crescente. Tão logo um novo tipo de hardware fosse disponibilizado, alguém o comprava e testava com o Linux, denominação pela qual o sistema gradualmente vinha sendo chamado, disponibilizando mais código livre para uma sempre maior quantidade de hardwares. Estes programadores não paravam com seus PC's; compravam cada peça de hardware que achassem ser útil para o Linux.

Naquele tempo, essas pessoas eram rotuladas “nerds” ou “freaks”, mas isto não lhes incomodava, desde que sua lista de hardwares suportados continuasse a crescer sempre mais e mais. Graças a essas pessoas, o Linux é hoje ideal não apenas para ser executado em máquinas novas, como é também a melhor escolha para as velhas e exóticas máquinas que hoje em dia seriam inúteis, não fosse por ele.

Dois anos após o comentário de Linus, havia cerca de 12000 usuários de Linux. O projeto, muito popular entre hobbistas, crescia constantemente, permanecendo sempre dentro dos limites do padrão POSIX. Todas as funções do UNIX foram adicionadas ao longo dos anos seguintes, resultando no sistema operacional maduro em que o Linux hoje se tornou. O Linux é um clone completo do UNIX, adequado para uso em estações de trabalho, bem como em servidores de médio e grande porte. Atualmente, muitos participantes importantes tanto no mercado de hardwares como no de softwares têm sua equipe de desenvolvedores Linux; através de seu distribuidor local é até mesmo possível comprar um computador com o Linux pré-instalado e ter suporte oficial, embora ainda existam muitos hardwares e softwares incompatíveis com o Linux.

## O atual uso dos sistemas Linux

Atualmente, o Linux entrou no mercado de desktops. Os desenvolvedores do Linux, no início, concentraram-se em redes e serviços, e os aplicativos de escritório foram a última barreira a ser transposta. Nós não gostamos de admitir que a Microsoft dá as regras neste mercado, então muitas alternativas foram iniciadas ao longo dos últimos anos como forma de tornar o Linux uma escolha aceitável para as estações de trabalho, disponibilizando uma interface simples para o usuário e aplicativos compatíveis com os programas de escritório da Microsoft, como os processadores de texto, planilhas e apresentações.

No mundo dos servidores, o Linux é muito conhecido por ser uma plataforma estável e confiável, fornecendo base de dados e serviços de comércio para empresas como a Amazon, a mundialmente conhecida livraria online, os Correios dos Estados Unidos, o Exército da Alemanha e muitos outros. Em especial, os provedores de acesso à Internet simpatizaram com o Linux como firewall, servidor proxy e web, e você encontrará um equipamento Linux perto de todo administrador de sistemas UNIX que aprecia uma estação administrativa confortável. Os clusters de máquinas Linux são usados na criação de filmes como “Titanic”, “Shrek” e outros. Nos correios, eles são os centros nevrálgicos que distribuem as cartas, e em grandes motores de pesquisa, os clusters são usados para realizar buscas na internet. Estes são apenas alguns dos milhares de trabalhos pesados que o Linux atualmente tem realizado ao redor do mundo todos os dias.

É importante frisar que os sistemas Linux modernos não podem ser executados apenas em estações de trabalho, servidores de médio e grande porte, mas também em “aparelhos” como os PDA's, celulares,

uma centena de aplicativos incorporados e até mesmo relógios de pulso experimentais. Isto torna o Linux o único sistema operacional no mundo inteiro capaz de ser executado em tantos tipos diferentes de hardware.

## A interface com o usuário

### O Linux é difícil?

Se o Linux é difícil de ser aprendido depende da pessoa de quem se quer saber a resposta. Usuários experientes de UNIX dirão que não, porquanto o Linux é o sistema operacional ideal para usuários avançados e programadores, vez que tem sido desenvolvido por e para estas pessoas.

Tudo o que um programador poderia desejar encontra-se disponível: compiladores, bibliotecas, ferramentas de desenvolvimento e depuradores. Estes pacotes vêm com toda distribuição de Linux padrão. O compilador C encontra-se incluído livremente - ao contrário do que ocorre em muitas distribuições do UNIX que requerem o pagamento de licenças por esta ferramenta. Toda a documentação e os manuais são disponibilizados e geralmente encontram-se incluídos exemplos para o rápido aprendizado. O Linux parece-se com o UNIX e mudar deste para o aquele é um processo muito natural.

Nos primórdios do Linux, ser um perito era quase um pré-requisito para começar a usar o sistema. Aqueles que dominavam o Linux sentiam-se melhores do que os demais "users" (trocadilho com as palavras perdedor e usuário em inglês) que ainda não tinham visto a luz. Era uma prática comum dizer a um usuário iniciante "LPDM" (leia a \* dos manuais). Embora os manuais estivessem em todos os sistemas, era difícil encontrar a documentação e mesmo que alguém conseguisse, as explicações eram feitas por meio de tantos termos técnicos que os novos usuários facilmente sentiam-se desencorajados de aprender o sistema.

A comunidade de usuários de Linux começou a se dar conta de que, se o Linux pretendia algum dia ser um competidor importante no mercado de sistemas operacionais, seriam necessárias algumas mudanças sérias na acessibilidade do sistema.

### Linux para usuários inexperientes

Empresas como a RedHat, a SuSE e a Mandriva surgiram, fornecendo distribuições empacotadas do Linux adequadas ao consumo em massa. Elas integraram interfaces gráficas (GUIs), desenvolvidas pela comunidade, de forma a facilitar o gerenciamento de programas e serviços. Como um usuário Linux hoje, temos todos os meios para aprender os meandros do sistema, porém não é mais primordial ter este conhecimento para fazer com que o sistema se adeque às necessidades de cada um.

Atualmente, é possível usar um ambiente gráfico, com todos os aplicativos necessários, sem nem ao menos precisar digitar um único caractere, ao passo em que ainda é possível acessar o núcleo do seu sistema se necessário for. Por conta de sua estrutura, o Linux permite ao usuário crescer com o sistema: ele se adequa igualmente aos usuários novatos e aos mais experientes. Os novatos não são obrigados a realizar tarefas muito complexas, ao passo em que os mais experientes não são obrigados a trabalhar da mesma forma que eles faziam quando começaram a aprender o Linux.

Embora o desenvolvimento na área de servidores continue ativo, grandes feitos vêm sendo realizados para os usuários de desktop, em geral considerados o grupo menos provável de se encontrar usuários que entendam como o sistema funciona. Desenvolvedores de aplicativos desktop têm realizado um esforço incrível para fornecer os desktops mais belos já vistos ou para fazer com que a sua máquina Linux pareça-se com a sua antiga estação de trabalho MS Windows ou Apple. Os últimos avanços incluem suporte para aceleração 3D, equipamentos USB, atualizações do sistema e dos pacotes em um clique e assim por diante. O Linux fornece isto e tenta fornecer todos os serviços possíveis de uma forma lógica para que pessoas comuns possam entendê-lo. Abaixo segue uma breve lista com alguns grandes exemplos; estes sites possuem muitas imagens de captura de tela que irão dar uma amostra do que a experiência com Linux em um desktop pode ser:

- <http://www.gnome.org>
- <http://kde.org/screenshots/>
- <http://www.openoffice.org>
- <http://www.mozilla.org>

## O Linux tem futuro?

### Open Source

A idéia por trás do software Open Source é muito simples: quando programadores podem ler, distribuir e modificar um código, tal código amadurecerá. As pessoas podem adaptá-lo, consertá-lo, procurar por erros, e podem fazê-lo a uma velocidade assustadoramente superior àquela dos desenvolvedores de software em empresas tradicionais. O software resultante será, portanto, mais flexível e de uma maior qualidade quando comparado àquele desenvolvido pelas vias convencionais, haja vista que mais pessoas o terão testado em uma maior variedade de condições do que os desenvolvedores de software fechado jamais poderiam fazer.

A Iniciativa Open Source começou a tornar isto claro para o mundo empresarial e, aos poucos, os

comerciantes estão entendendo o ponto. Embora muitos acadêmicos e técnicos já estivessem convencidos há mais de 20 anos de que este é o caminho a seguir, as empresas precisavam de aplicativos como a Internet para que perceber que podem obter lucro a partir do Open Source. Hoje o Linux cresceu além do estágio no qual era apenas um sistema acadêmico, útil somente para um punhado de pessoas com formação técnica. Hoje ele fornece mais do que um sistema operacional: existe toda uma infra-estrutura de união de esforços para criar o sistema, elaborando e testando programas para ele, trazendo tudo aos usuários, fornecendo manutenção, atualizações, suporte, personalização e etc. Hoje, o Linux está pronto para aceitar o desafio de um mundo em constante mudança.

## Dez anos de experiência a seu dispor

Embora o Linux provavelmente seja a iniciativa Open Source mais conhecida, existe um outro projeto que contribuiu grandemente para a popularidade do sistema operacional. Este projeto chama-se SAMBA e sua grande façanha foi ter realizado a engenharia reversa do protocolo Server Message Block (SMB)/Common Internet File System (CIFS) no fornecimento de arquivos e impressoras em equipamentos relacionados aos PCs, originalmente suportados pelo MS Windows NT, pelo OS/2 e pelo Linux. Os pacotes agora são disponibilizados para quase todos os sistemas e fornecem soluções de interação em ambientes mistos usando protocolos do MS Windows: compatíveis com os servidores de arquivos e impressão do Windows (até e incluindo o Windows XP).

Talvez até mais famoso do que o projeto SAMBA, outro importante projeto é o de servidores Apache HTTP. O servidor pode ser executado no UNIX, no Windows NT e muitos outros sistemas operacionais. Inicialmente conhecido como "A PAtCHy server" (um servidor corrigido), baseado em trechos de código pré-existent e uma série de "patch files" (arquivos de correção), o nome para o código desenvolvido faz menção à tribo nativa dos Estados Unidos, os índios Apache, famosos por suas excelentes habilidades em estratégia de guerra, bem como inesgotável energia. O Apache demonstrou ser substancialmente mais rápido, estável e completo do que muitos outros servidores web. Ele é usado em sites que têm milhões de acessos diários e, embora não exista suporte oficial dos desenvolvedores, a comunidade de usuários Apache provê as respostas para todas as perguntas a ele relacionadas. Atualmente, também o suporte comercial tem sido fornecido por uma extensa gama de terceiros.

Na categoria de aplicativos de escritório, alternativas à suíte MS Office encontram-se disponíveis, possuindo desde implementações parciais até completas dos aplicativos existentes nas máquinas MS Windows. Estas iniciativas ajudaram muitos a tornar o Linux aceitável para o mercado de desktops, porquanto os usuários não precisam de treinamento extra para aprender a trabalhar com os novos sistemas. Com o desktop vêm o reconhecimento dos usuários comuns e, também, suas exigências específicas, que têm se tornado cada dia mais complexas.

A comunidade Open Source, formada em grande parte por pessoas que vêm contribuindo por mais de uma década, garante a posição do Linux enquanto um importante participante no mercado de desktops, como também no mercado de TI em geral. Funcionários remunerados e voluntários têm trabalhado diligentemente para que o Linux mantenha sua posição no mercado. Quanto mais usuários, mais perguntas. A comunidade Open Source garante que as respostas continuem surgindo, averiguando sempre a qualidade destas com um ar crítico, do que resulta em mais estabilidade e acessibilidade.

Apresentar uma lista completa de todos os softwares Linux disponíveis é algo que se encontra além do escopo deste guia, uma vez que existem milhares de pacotes. Ao longo deste curso nós apresentaremos os pacotes mais comuns, os quais se encontram, em grande parte, livremente disponíveis. De forma a afastar o medo do usuário iniciante, aqui segue uma captura de tela de um de seus programas prediletos. Veja por si mesmo que não foram poupados esforços para fazer com que os usuários que estão migrando do Windows sintam-se em casa:

**Figure 1.1. Planilha de cálculo do OpenOffice (compatível com o MS Excell)**



## Propriedades do Linux

### Os prós do Linux

Muitas das vantagens do Linux são uma consequência de suas origens, profundamente arraigadas no UNIX, excetuando a sua primeira vantagem, é claro:

- O Linux é livre:

Como na cerveja livre, dizem. Se não se quer gastar absolutamente nada, não é necessário nem ao menos dispendir o valor de um CD. O Linux pode ser integral e gratuitamente baixado da Internet. Sem taxas de adesão, custo por cabeça, atualizações gratuitas e códigos-fonte livremente disponíveis, para a eventualidade de o usuário desejar alterar o funcionamento do sistema.

Acima de tudo, o Linux é livre no mesmo sentido em que o é a expressão do pensamento

A licença geralmente utilizada é a Licença Pública Geral GNU (GPL). A licença determina que qualquer um tem o direito de mudar o Linux, bem como redistribuir a versão alterada, sob a condição de que o código continue disponível após a redistribuição. Na prática, você é livre para pegar uma imagem do kernel, por exemplo, para adicionar suporte para máquinas de tele-transporte ou viagem no tempo e vender seu novo código, contanto que a comunidade continue a poder ter uma cópia do

código.

- O Linux é portátil para qualquer plataforma de hardware:

Um empresário que intencione comercializar um novo modelo de computador, porém que não saiba que tipo de sistema operacional sua nova máquina irá executar (por exemplo, a CPU no seu carro ou máquina de lavar), pode usar o kernel do Linux e fazê-lo funcionar no novo hardware, porquanto a documentação relacionada a esta tarefa encontra-se livremente disponível.

- O Linux foi desenvolvido para um uso contínuo

Assim como o UNIX, um sistema Linux é preparado para ser executado sem ser frequentemente reiniciado. É por isto que muitas tarefas são executadas durante a noite ou então são automaticamente agendadas para outros momentos tranquilos, do que resulta em uma maior disponibilidade ao longo de períodos mais agitados e em um uso mais balanceado do hardware. Esta característica permite que o Linux seja utilizado também em ambientes onde as pessoas não dispõem de tempo ou têm a possibilidade de controlar seus sistemas noite e dia.

- O Linux é seguro e versátil:

O modelo de segurança usado no Linux é alicerçado no conceito de segurança do UNIX, famoso por ser forte e de comprovada qualidade. Mas o Linux não é adequado apenas para a defesa de ataques inimigos na Internet: ele é igualmente adaptado para outras situações, usando os mesmos elevados padrões de segurança. Sua máquina de desenvolvimento ou estação de controle será tão segura quanto seu firewall.

- O Linux é plástico:

Desde um "Palmtop" com 2 MB de memória até um cluster de um petabyte com centenas de nós: adicione ou remova os pacotes apropriados e o Linux adequar-se-á a ambos. Não é mais necessário um supercomputador, porque agora é possível usar o Linux para realizar grandes coisas usando os tijolos providos junto com o sistema. Se você quer fazer coisas pequenas, como um sistema operacional para um processador incorporado ou apenas reciclar seu antigo 486, o Linux servirá igualmente.

- O sistema Linux e a maior parte de seus aplicativos têm um tempo de depuração muito curto:

Em razão de o Linux ser desenvolvido e testado por milhares de pessoas, ambos, os erros e as pessoas para consertá-los, são rapidamente encontrados. Às vezes acontece de existir apenas um lapso de algumas horas entre a descoberta e o conserto de um erro.

## Os contras do Linux

- Existem infindáveis distribuições:

"Quot capites, tot rationes", como os romanos já diziam: quanto mais pessoas, mais opiniões. Logo de cara, a quantidade de distribuições do Linux pode ser assustadora ou ridícula, dependendo do ponto de vista. Mas também significa que todos irão encontrar o que precisam. Não é necessário ser um perito para encontrar uma distribuição adequada.

Quando perguntado, via de regra todo usuário Linux dirá que a melhor distribuição é aquela específica que ele vem usando. Então qual deve ser escolhida? Não se preocupe muito com isto: todas as distribuições contêm mais ou menos o mesmo conjunto de pacotes básicos. A partir do básico, softwares especializados de terceiros são adicionados produzindo, por exemplo, o TurboLinux que é mais adequado às pequenas e médias empresas, o RedHat para os servidores e o SuSE para as estações de trabalho. Todavia, as diferenças em geral são muito superficiais. A melhor estratégia é testar algumas distribuições; infelizmente, nem todos dispõem de tempo para isto. Por sorte, existem muitos conselhos a se observar quando da escolha de uma distribuição do Linux. Uma busca rápida no [Google](#), com as palavras-chave "escolhendo a sua distribuição" traz milhares de links para um bom aconselhamento. O [Installation HOWTO](#) também disserta sobre a escolha de uma distribuição (em inglês).

- O Linux não é muito amigável e pode ser confuso para iniciantes:

É preciso dizer que o Linux, ao menos o núcleo do sistema, é menos amigável do que o MS Windows e certamente mais difícil do que o MacOSX, mas... Por conta de sua popularidade, um esforço considerável vem sendo feito para tornar o Linux mais fácil de usar, especialmente para os novatos. Mais informação tem sido publicada diariamente, como este guia, por exemplo, para ajudar a preencher a falta de documentação disponível aos usuários de todos os níveis.

- Um produto de código-fonte aberto é confiável?

Como é possível que algo livre seja também confiável? Os usuários de Linux têm a escolha entre usá-lo ou não, o que lhes garante uma enorme vantagem sobre os usuários de softwares proprietários, que não têm esta liberdade. Após longos períodos de testes, a maior parte dos usuários de Linux chega à conclusão de que o Linux não é apenas bom, mas, em muitos casos, é melhor e mais rápido do que as soluções tradicionais. Se o Linux não fosse confiável, ele teria deixado de existir há muito tempo, jamais alcançando a popularidade que tem atualmente, com milhões de usuários. Agora os usuários podem influenciar seus sistemas e compartilhar suas observações com a comunidade, de modo que o sistema torna-se melhor a cada dia. É um projeto que jamais terminará, isto é verdade,

mas em um ambiente em contínua mutação, o Linux é também um projeto que continua a perseguir a perfeição.

## Os Sabores do Linux

### O Linux e o GNU

Embora exista um grande número de implementações do Linux, você encontrará muitas semelhanças entre as diferentes distribuições, justamente porque toda a máquina Linux é apenas uma caixa com tijolos que permite o uso de acordo com necessidades e perspectivas individualizadas. Instalar o sistema é apenas o começo de uma relação duradoura. Assim que você acha que tem um ótimo sistema, o Linux irá estimular sua imaginação e criatividade, e quanto mais você percebe o poder que o sistema pode lhe dar, mais você tentará redefinir seus limites.

O Linux poderá parecer diferente dependendo da distribuição, do hardware e gosto pessoal de cada um, mas os fundamentos sobre os quais todos os gráficos e outras interfaces são construídas, permanecem os mesmos. O sistema Linux é baseado nas ferramentas do GNU (Gnu Não é UNIX), que, por sua vez, disponibilizam um conjunto de modos padrão de lidar com e usar o sistema. Todas as ferramentas do GNU são de código-fonte aberto, de modo que elas podem ser instaladas em qualquer novo sistema. A maioria das distribuições oferece pacotes pré-compilados das ferramentas mais comuns, como, por exemplo pacotes RPM no RedHat e pacotes Debian (também chamados de deb ou dpkg) no Debian, de forma que não é necessário ser um programador para instalar um pacote no seu sistema. Entretanto, se você é e gosta de fazer as coisas você mesmo, você aproveitará o Linux tanto mais, já que a maior parte das distribuições vêm com um conjunto completo de ferramentas de desenvolvimento, que permitem a instalação de novos softwares diretamente a partir do código-fonte. Esta configuração também permite a instalação de softwares ainda que não existam pacotes pré-configurados para o seu sistema.

Uma lista de softwares GNU:

- Bash: O Shell do GNU
- GCC: O Compilador C do GNU
- GDB: O Depurador do GNU
- Coreutils: um conjunto de utilitários básicos do estilo UNIX, como o **ls**, o **cat** e o **chmod**
- Findutils: para procurar por arquivos
- Fontutils: para transformar fontes de um formato para outro ou criar novas fontes
- O Gimp: O programa de manipulação de imagens do GNU
- Gnome: o ambiente de trabalho do GNU
- Emacs: um editor muito potente
- Ghostscript e Ghostview: interpretador e front-end gráfico para arquivos PostScript.
- GNU Photo: software para interação com câmeras digitais cameras
- Octave: uma linguagem de programação, focada em realizar computação de números e processamento de imagens.
- GNU SQL: base de dados relacional
- Radius: um servidor remoto de autenticação de contas
- ...

Muitos aplicativos comerciais estão disponíveis para o Linux e para obter mais informações sobre estes pacotes nos iremos fazer referência à sua documentação específica. Ao longo deste guia focaremos em discutir apenas softwares livres, os quais vêm (na maior parte das vezes) com uma licença GNU.

Para instalar softwares ausentes ou novos pacotes, é necessário algum método de gerenciamento de softwares. As soluções mais comuns são o RPM e o dpkg. RPM é o Gerenciador de pacotes da RedHat, usado em uma grande variedade de sistemas Linux, embora o nome não indique. Dpkg sistema de gerenciamento de pacotes do Debian, que usa uma interface chamada **apt-get**, que também consegue gerenciar pacotes RPM. Novell Ximian Red Carpet é uma implementação de terceiros de uma interface gráfica para o gerenciador RPM. Outros comerciantes de softwares poderão usar de métodos de instalação específicos, algumas vezes parecidos com o InstallShield e seus semelhantes, conhecidos no MS Windows e outras plataformas. A medida em que o usuário cresce no Linux, é inevitável deparar-se com um ou mais destes programas.

### O GNU/Linux

O kernel do Linux (os ossos do sistema, veja [the section called "O kernel"](#)) não faz parte do projeto GNU, mas faz uso das mesmas licenças que os softwares do GNU. A esmagadora maioria dos utilitários e ferramentas de desenvolvimento (a carne do sistema), que não são específicos do Linux, são



emprestados do GNU. Porquanto qualquer sistema utilizável precisa ter ambos, o kernel e um conjunto mínimo de utilitários, algumas pessoas sustentam que o sistema deveria ser chamado de *GNU/Linux*.

Buscando o maior grau de independência possível em relação às distribuições, este será o modelo de Linux que iremos abordar ao longo deste curso. Quando não estiver em tela um sistema GNU/Linux, a distribuição específica, versão ou nome do programa será mencionado.

## Qual distribuição devo instalar?

Antes da instalação, o fator mais importante é o hardware. Uma vez que todas as distribuições do Linux contém todos os pacotes básicos e podem ser configuradas para atender a quase todas as necessidades (porque todas usam o kernel do Linux), somente é necessário atentar se a distribuição poderá ser executada no seu hardware. O LinuxPPC, por exemplo, foi criado para ser executado em máquinas Apple e outros PowerPCs e não poderá ser executado em um PC x86 comum. O LinuxPPC pode ser executado em novos Macs, mas não é possível executá-lo em alguns mais antigos, que fazem uso da arcaica tecnologia de barramento. Outro caso delicado é o dos hardwares da Sun, que podem ser tanto um velho CPU Sparc ou um novo UltraSparc, cada um necessitando de versões específicas do Linux.

Algumas distribuições do Linux são otimizadas para processadores específicos, como os CPU Athlon, embora eles também possam ser executados decentemente em qualquer processador 486, 586 ou 686 da Intel. Acontece, às vezes, de algumas versões para máquinas especiais não serem tão confiáveis, uma vez que são testadas por um grupo menor de pessoas.

A maior partes das distribuições do Linux oferece um conjunto de programas para PCs genéricos com pacotes especiais contendo kernels otimizados para aqueles com processadores Intel x86 Intel. Estas distribuições são regularmente testadas e bem mantidas, com foco em implementações de servidor confiáveis e processos simplificados para instalação e atualização. Exemplos destas distribuições são: o Debian, o Ubuntu, o Fedora, o SuSE e o Mandriva, que são, de longe, as distribuições mais populares do sistema Linux e, em geral, consideradas fáceis de lidar para um usuário iniciante, embora não impeçam os mais experientes de extrair o desempenho máximo de suas máquinas. O Linux também é bem executado em laptops e servidores de médio porte. Drivers para novos hardwares são incluídos apenas após longos testes, o que acresce em estabilidade ao sistema.

Embora muitos sistemas ofereçam o Gnome como ambiente desktop padrão, existem também aqueles que adotam o KDE. Em geral, ambos, Gnome e KDE encontram-se disponíveis nas principais distribuições do Linux. Outros gerenciadores de janelas e desktop encontram-se disponíveis para os usuários mais avançados.

O processo de instalação padrão permite aos usuários a escolha entre diferentes configurações básicas, como estação de trabalho, onde todos os pacotes necessários para as tarefas do dia-a-dia e de desenvolvimento são instalados, ou configuração de servidor, onde diferentes pacotes de rede e serviços podem ser selecionados. Os peritos pode instalar quaisquer combinações de pacotes que desejarem ao longo do processo de instalação inicial.

O objetivo deste guia é abranger todas as distribuições do Linux. Para a sua própria conveniência, entretanto, é altamente recomendado que usuários iniciantes adiram a uma distribuição popular, que dê suporte aos hardwares e aplicações mais comuns. As seguintes são ótimas escolhas para novatos:

- [Fedora Core](#)
- [Debian](#)
- [SuSE Linux](#)
- [Mandriva \(antigo MandrakeSoft\)](#)
- [Knoppix](#): um sistema operacional que pode ser executado direto do CD-ROM, dispensando a instalação.

É possível baixar imagens ISO através do site [LinuxISO.org](#). As principais distribuições podem ser adquiridas em qualquer loja de informática decente.

## Resumo

Neste capítulo, aprendemos que:

- O Linux é uma implementação do UNIX.
- O sistema operacional Linux é escrito na linguagem de programação C.
- “De gustibus et coloribus non disputandum est”: existe um Linux para cada um.
- O Linux faz uso das ferramentas do GNU, um conjunto de ferramentas padrão livremente disponíveis para lidar com o sistema operacional.

## Exercícios

Um exercício prático para iniciantes: instale o Linux no seu PC. Leia o manual de instalação da sua distribuição e/ou o HOWTO de instalação e então instale.

## Leia a documentação!

A maior parte dos erros advém da falta de leitura das informações disponibilizadas ao longo da instalação. Ler as mensagens de instalação com cuidado é o primeiro passo no caminho para o sucesso.

É preciso saber algumas coisas ANTES de começar a instalar o Linux:

- A distribuição escolhida pode ser executada no seu hardware?

Confira em <http://www.tldp.org/HOWTO/Hardware-HOWTO/index.html> (em inglês) se estiver em dúvida a respeito da compatibilidade do seu equipamento.

- Qual tipo de teclado você usa (número de teclas, formato)? Qual tipo de mouse (serial/paralelo, número de botões)? Quantos MB de RAM?
- Qual o padrão da instalação: um servidor, uma estação de trabalho ou será preciso escolher e instalar pacotes específicos?
- A instalação será feita a partir do meu HD, de um CD-ROM, ou de uma rede? É necessário adaptar o BIOS para quaisquer destes modos? O método de instalação requer um disco de inicialização?
- O Linux será o único sistema no computador ou será o segundo em uma instalação dupla? Será preciso criar uma partição grande para poder instalar sistemas virtuais depois ou esta é uma instalação virtual ela mesma?
- Este computador está em uma rede? Qual é o seu nome de máquina (hostname) e endereço IP? Existem outros servidores de portas ou outras máquinas importantes na rede com as quais esta máquina deve conseguir se comunicar?

## O Linux espera estar em rede

Não usar a rede ou configurá-la erroneamente pode resultar em uma inicialização lenta.

- Este computador é uma porta/roteador/firewall? (Se você precisa pensar sobre esta questão, provavelmente não é.)
- Particionamento: deixe o programa de instalação fazê-lo apenas desta vez, discutiremos as partições em detalhes no [Chapter 3, Dos arquivos e do sistema de arquivos](#). Existe documentação específica para cada sistema se você quiser saber tudo sobre isto. Se a sua distribuição do Linux não oferece um particionamento padrão, este é um indício de que ela não é recomendada para iniciantes.
- A máquina irá iniciar no modo texto ou gráfico?
- Pense em uma boa senha para o administrador da máquina (root). Crie uma conta de usuário não-root (acesso sem privilégios ao sistema).
- Você precisa de um disco de resgate? (recomendado)
- Quais idiomas você deseja?

A lista completa pode ser encontrada no <http://www.tldp.org/HOWTO/Installation-HOWTO/index.html> (em inglês).

Nos capítulos a seguir descobriremos se a instalação foi um sucesso.

## Chapter 2. Início Rápido

### Table of Contents

[Como efetuar o login, ativar a interface de usuário e efetuar o logout](#)

[Introdução](#)  
[Modo Gráfico](#)  
[Modo Texto](#)

[O básico do básico](#)

[Os comandos](#)  
[Observações](#)  
[Usando os recursos Bash](#)

[Buscando ajuda](#)

[Esteja Ciente](#)  
[As páginas man](#)  
[Mais informações](#)

[Resumo](#)  
[Exercícios](#)

[Conectando e desconectando](#)  
[Senhas de acesso](#)  
[Diretórios](#)  
[Arquivos](#)  
[Buscando ajuda](#)

## Abstract

De forma a extrair o máximo deste guia, começaremos imediatamente com um capítulo prático sobre como conectar-se ao sistema Linux e realizar algumas operações básicas.

Discutiremos:

- Como conectar-se com o sistema
- Como desconectar-se do sistema
- Modos texto e gráfico
- Como mudar a senha de acesso
- Como navegar pelo sistema de arquivos
- Como determinar o tipo de arquivo
- Como analisar arquivos-texto
- Como encontrar ajuda

## Como efetuar o login, ativar a interface de usuário e efetuar o logout

### Introdução

Para trabalhar diretamente em um sistema Linux, será necessário fornecer um nome de usuário e uma senha de acesso. Será sempre necessário identificar-se ao sistema. Como previamente mencionado no exercício do [Chapter 1, O que é o Linux?](#), a maior parte dos sistemas Linux em PCs tem dois modos básicos de execução: um rápido e sóbrio modo texto, que se parece com um DOS com mouse e recursos multitarefa e multi-usuário, ou um modo gráfico, com um aspecto mais agradável, mas que também consome mais recursos.

### Modo Gráfico

Atualmente, este é o padrão na maior parte dos computadores desktop (de mesa). Você sabe que irá conectar-se ao sistema usando o modo gráfico quando inicialmente é perguntado o nome de usuário e, então, em uma nova tela, a senha de acesso.

Para efetuar o login, certifique-se de que o cursor do mouse encontra-se na janela de login, forneça o seu nome de usuário e senha de acesso ao sistema e clique OK ou aperte **Enter**.

### Cuidado com a conta root!

Em geral, considera-se uma má idéia conectar (graficamente) usando o usuário *root*, a conta administrativa do sistema, vez que o uso de gráficos inclui a execução de muitos outros programas, no caso do *root*, com muitas outras permissões. Para manter os riscos tão baixos quanto possível, use uma conta de usuário normal para conectar-se graficamente. Para simplificar, uma regra geral para o uso da conta *root*: apenas efetue o login como *root* quando tiver necessidade de obter privilégios extraordinários.

Após informar seu nome de usuário e senha de acesso, pode demorar um pouco antes do ambiente gráfico iniciar-se, o que varia de acordo com a velocidade do CPU da sua máquina, os softwares que você usa e suas configurações pessoais.

Para continuar, será necessário abrir um *terminal* ou, abreviando, *xterm* (X sendo o nome do software que dá suporte ao ambiente gráfico). Este programa pode ser encontrado menu Aplicativos → Utilitários, Ferramentas do Sistema ou Internet, de acordo com o gerenciador de janelas em uso. Pode também haver ícones de atalho para iniciar uma janela *xterm* e, normalmente, clicar com o botão direito do mouse exibe um menu com um aplicativo para iniciar um terminal.

Enquanto estiver navegando pelos menus, ficará claro que muitas tarefas podem ser realizadas sem a necessidade de informar comandos pelo teclado. Para a maior parte dos usuários, o bom e velho método do apontar-clicar é suficiente. Este guia, entretanto, é destinado a futuros administradores de sistemas e redes, que precisarão lidar com o coração do sistema. Eles precisam de uma ferramenta mais poderosa do que um mouse para lidar com todas as tarefas com as quais irão se deparar. Esta ferramenta é o shell e, no modo gráfico, ativa-se o shell através do terminal.

O terminal é o painel de controle para o sistema. Quase tudo daqui em diante é feito através desta simples, porém poderosa ferramenta. Um terminal deve sempre mostrar um prompt de comando quando

aberto. Este terminal mostra um prompt padrão, que apresenta o nome de usuário e o diretório atual, representado pelo acento til (~):

### Figure 2.1. Janela de terminal



Outra forma comum de prompt é esta:

```
[user@host dir]
```

No exemplo acima, *usuário* é o seu nome de usuário, *hosts* o nome da máquina em que se está trabalhando e *dir* uma indicação do presente local dentro do sistema de arquivos.

No futuro discutiremos os prompts e seu comportamento em detalhes. Por ora, é suficiente saber que os prompts podem apresentar todos os tipos de informação, mas não são parte dos comandos que você fornece ao sistema.

Para desconectar do sistema no modo gráfico, será preciso fechar todos os terminais e demais aplicativos. Após, clique no ícone de logout ou encontre o Encerrar no menu. Fechar tudo não é realmente necessário, o sistema pode fazê-lo por você. Porém, o gerenciador de sessões poderá reabrir todos os aplicativos atuais quando de um novo login, o que demora mais e pode não ser o efeito desejado. Ressalve-se apenas que este comportamento é configurável.

Quando você vir a tela de login novamente, pedindo pelo seu nome de usuário e senha de acesso, o logout foi concluído com sucesso.

## Gnome ou KDE?

Mencionamos ambos Gnome e KDE já algumas vezes. Estes são os dois meios mais populares de gerenciar o seu ambiente de trabalho, embora existam muitos, muitos outros. O ambiente de trabalho escolhido é indiferente, desde que se saiba como abrir um terminal. Entretanto, continuaremos a nos referir a ambos, Gnome e KDE para indicar os modos mais populares de alcançar alguns objetivos.

## Modo Texto

Percebe-se que se está usando o modo texto quando a tela inteira é preta, apresentando (na maior parte dos casos) caracteres brancos. Uma tela de login no modo texto normalmente apresenta algumas informações sobre a máquina em que se está trabalhando, o nome da máquina e um prompt aguardando que usuário efetue o login:

```
RedHat Linux Release 8.0 (Psyche)
```

```
blast login: _
```

O login difere do login gráfico, no sentido de que será preciso teclar **Enter** após fornecer o nome de usuário, porquanto não existem botões na tela para se clicar com o mouse. Então será preciso teclar a senha de acesso, seguida de um novo **Enter**. Não haverá qualquer indício de que se está entrando em algum lugar, nem mesmo um asterisco, nem ao menos se perceberá o cursor mover-se. Isto, porém, é normal no Linux e assim é por motivos de segurança.

Quando o sistema reconhecê-lo como um usuário válido, ele poderá apresentar mais informações, denominadas *mensagem do dia*, que podem ser qualquer coisa. Além disto, é muito comum que sistemas UNIX apresentem uma mensagem de ventura, com pensamentos, sábios ou não (o que cabe a você decidir). Então, será apresentado o shell, com o mesmo prompt que seria apresentado no modo gráfico.

## Não efetue o login como root

Também no modo texto: apenas efetue o login como root quando precisar realizar configurações que somente podem ser feitas com privilégios administrativos, como, por exemplo, criar novas contas de usuário, instalar novos pacotes de softwares e configurar a rede ou o sistema. Após realizar tais tarefas, saia imediatamente da conta especial e volte a operar como um usuário normal. Como alternativa, alguns sistemas, como o Ubuntu, obrigam o usuário a utilizar o comando **sudo**, de modo que se torna dispensável o acesso direto à conta administrativa.

O logout é realizado através do comando **logout**, seguido da tecla **Enter**. Quando o sistema apresentar novamente a tela de login, o logout terá sido realizado com sucesso.

## O botão de energia (power)

Embora o Linux não tenha sido desenvolvido para ser desligado sem os procedimentos adequados de paralisação do sistema, apertar o botão de energia, *em sistemas mais recentes*, é equivalente a dar início a tais procedimentos. Contudo, desligar um sistema mais antigo sem

realizar todo o processo de paralisação poderá resultar em graves danos! Para garantir-se, use sempre a opção de Desligar para realizar o logout ambientes gráficos ou, quando estiver na tela de acesso ao sistema, ao invés de fornecer o nome de usuário e a senha, procure pelo botão de desligar.

Agora que sabemos como conectar e desconectar do sistema, estamos prontos para nossos primeiros comandos.

## O básico do básico

### Os comandos

Esta é apenas uma prévia daquilo que precisaremos para começar; posteriormente discutiremos estes comandos em detalhes.

**Table 2.1. Comandos básicos**

Comando	Significado
<b>ls</b>	Apresenta uma lista dos arquivos no diretório atual, como o comando <b>dir</b> do DOS
<b>cd</b> <i>directory</i>	muda de diretório (Change Directory)
<b>passwd</b>	muda a senha do usuário atual
<b>file</b> <i>nome do arquivo</i>	exibe o tipo de arquivo do arquivo com o nome <i>nome do arquivo</i>
<b>cat</b> <i>arquivotexto</i>	joga o conteúdo do <i>arquivotexto</i> na tela
<b>pwd</b>	exibe o atual diretório de trabalho
<b>exit</b> or <b>logout</b>	sai desta sessão
<b>man</b> <i>comando</i>	lê páginas man sobre o <b>comando</b>
<b>info</b> <i>comando</i>	lê páginas Info do <b>comando</b>
<b>apropos</b> <i>seqüência</i>	procura pela seqüência na base de dados <i>whatis</i>

### Observações

Digite estes comandos após o prompt, em um terminal no modo gráfico ou no modo texto, e em seguida tecla **Enter**.

Comandos podem ser informados sozinhos, como, por exemplo **ls**. Um comando se comporta de modo diverso quando é informada uma *opção*, normalmente precedida por um hífen (-), como no comando **ls -a**. O mesmo caractere de opção pode ter um significado diferente para outro comando. Os programas GNU aceitam opções longas, que devem ser precedidas por dois hífen (--), como em **ls --all**. Alguns podem não ter opções.

O(s) argumento(s) para um comando são especificações para o(s) objeto(s) sobre o(s) qua(l)(is) deseja-se que o comando aja. Um exemplo é **ls /etc**, no qual o diretório */etc* é o argumento para o comando **ls**. Isto indica que deseja-se saber o conteúdo do diretório indicado, ao invés do padrão, que seria aquele do atual diretório, obtido quando se digita apenas o **ls** seguido da tecla **Enter**. Alguns comandos necessitam de argumentos, em outros casos eles são opcionais.

É possível descobrir se um comando aceita opções e argumentos, bem como quais lhe são válidos, buscando ajuda online para aquele comando, veja [the section called "Buscando ajuda"](#).

No Linux, como no UNIX, os diretórios são separados por barras, como as que são usadas em endereços web (URL). Mais adiante, discutiremos a estrutura dos diretórios em profundidade.

Os símbolos `.` e `..` têm um significado especial em se tratando de diretórios. Tentaremos entendê-los ao longo dos exercícios e ao longo do próximo capítulo.

Tente evitar efetuar logins ou usar o sistema com a conta administrativa, *root*. Além de realizar seu trabalho cotidiano, a maior parte das tarefas, incluindo checar o sistema, reunir informações e etc, podem ser realizadas utilizando-se uma conta de usuário comum, sem quaisquer permissões especiais. Quando necessário, por exemplo para criar uma nova conta de usuário ou instalar um novo software, o melhor modo de obter o acesso ao *root* é mudando a ID do usuário, como em [the section called "O caminho \(path\)"](#), por exemplo.

Quase todos os comandos neste livro podem ser executados sem os privilégios administrativos. Na maior parte dos casos, quando se fornece um comando ou se inicia um programa como um usuário sem privilégios, o sistema irá avisá-lo sobre a necessidade de privilégios administrativos ou pedir pela senha *root* quando necessário for. Saia imediatamente do aplicativo ou sessão que conta com privilégios do *root* quando tiver terminado a tarefa.

Ler a documentação deve tornar-se a sua segunda natureza. Principalmente no começo, é importante ler a documentação do sistema, os manuais para os comandos básicos, os HOWTOs e etc. Uma vez que a quantidade de documentos é imensa, é impossível incluir toda a documentação relacionada. Este livro tentará servir de guia aos documentos mais apropriados a cada um dos tópicos discutidos, de forma a estimular o hábito de ler as páginas man.

## Usando os recursos Bash

Várias combinações especiais de teclas permitem agilizar e facilitar a realização de tarefas com o shell do GNU, Bash, que é o padrão na maior parte dos sistemas Linux, veja [the section called "O shell"](#). Abaixo encontra-se uma lista dos recursos mais comuns; é altamente recomendado que se torne um hábito usá-los, de forma a extrair o máximo da sua experiência com o Linux desde o início.

**Table 2.2. Combinação de teclas no Bash**

Tecla ou combinação de teclas	Função
<b>Ctrl+A</b>	Movê o cursor para o começo da linha de comando.
<b>Ctrl+C</b>	Finaliza um programa em execução e retorna ao prompt, veja <a href="#">Chapter 4, Processos</a> .
<b>Ctrl+D</b>	Efetua o logout da atual sessão do shell, o mesmo que digitar <b>exit</b> ou <b>logout</b> .
<b>Ctrl+E</b>	Movê o cursor para o final da linha de comando.
<b>Ctrl+H</b>	Gera o caractere de backspace.
<b>Ctrl+L</b>	Limpa o terminal
<b>Ctrl+R</b>	Busca pelo histórico de comandos, veja <a href="#">the section called "O comando grep"</a> .
<b>Ctrl+Z</b>	Suspende um programa, veja <a href="#">Chapter 4, Processos</a> .
<b>ArrowLeft e ArrowRight</b>	Movê o cursor uma posição para a esquerda ou direita na linha de comando, de modo que você possa inserir caracteres em outros locais ao invés do começo e do final.
<b>ArrowUp e ArrowDown</b>	Navega pelo histórico. Vai para a linha que se deseja repetir, onde se pode editar os detalhes se necessário, seguido de <b>Enter</b> para economizar tempo.
<b>Shift+PageUp e Shift+PageDown</b>	Navega pelo buffer do terminal do terminal (para visualizar o texto que "saiu" da tela).
<b>Tab</b>	Completa comando ou nome de arquivo; quando houver mais de uma escolha possível, o sistema irá sinalizar por áudio ou vídeo ou, se muitas escolhas forem possíveis, perguntar se o usuário deseja visualizar todas elas.
<b>Tab Tab</b>	Mostra as possibilidades de complemento de arquivo ou comando.

Os últimos dois itens da tabela acima carecem de maiores explicações. Por exemplo, quando se deseja mudar para o diretório `diretório_com_um_nome_muito_extenso`, não é preciso digitar o nome completo. Digita-se apenas na linha de comando `cd dir` e pressionando, então, a tecla **Tab**, o shell irá completar o nome do diretório, se nenhum outro arquivo começar por estes mesmos três caracteres. Claro que se não houver outros itens iniciados por "d", então será também possível atingir o mesmo objetivo tecendo apenas `cd d` seguido de **Tab**. Se mais de um arquivo começar com os mesmos caracteres, o shell emitirá um sinal, de forma que você poderá pressionar **Tab**, duas vezes seguidas em um curto intervalo, e o shell apresentará as possíveis escolhas:

```
seu_prompt> cd st
comece aqui   coisas   coisinhas
```

No exemplo acima, se for digitado "m" após os dois primeiros caracteres e então **Tab** novamente, nenhuma outra possibilidade sobra, de modo que o shell completa o nome do diretório, sem precisar digitar a seqüência "ece aqui":

```
your_prompt> cd comece aqui
```

É óbvio que também é necessário pressionar **Enter** para informar a escolha.

No mesmo exemplo, se for digitado "i" e então a tecla **Tab**, o shell irá adicionar "s" e protestar novamente, porquanto múltiplas alternativas são possíveis. Digitando-se, então, **TabTab** novamente, serão apresentadas as escolhas; se for digitado um ou mais caracteres de forma a tornar única a escolha para o sistema, seguido novamente de **Tab** ou **Enter** quando finalizado o nome que se deseja escolher, o shell completa o nome e adentra no diretório - se de fato se tratar de um nome de diretório.

Isto funciona para todos os nome que são argumentos para comandos.

O mesmo serve para completar os nomes de comandos. Digitar **ls** e então pressionar a tecla **Tab** duas vezes, lista todos os comandos no PATH (veja [???](#)) iniciados por estes dois caracteres:

```
seu_prompt> ls
ls          lsdev      lspci      lsraid     lsw
lsattr      lsmod      lspgpot    lss16toppm
lsb_release lsof       lspnp      lsusb
```

## Buscando ajuda

### Esteja Ciente

O GNU/Linux está intimamente ligado com tornar-se mais autônomo. E como é comum neste sistema,

existem várias maneiras de se alcançar um objetivo. Um modo comum de conseguir ajuda é achando alguém que saiba, mas ainda que a comunidade de usuários GNU seja muito pacífica e paciente, quase todos esperam que você tenha tentado um ou mais dentre os métodos nesta seção antes de perguntar, e a maneira como este ponto de vista é expresso pode ser agressiva se você demonstrar que não seguiu esta regra básica.

## As páginas man

Muitos usuários iniciantes têm medo das páginas man (de manual), porque elas são uma fonte prolixa de documentação. Entretanto, elas são, também, bem estruturadas, como se pode ver no exemplo que segue: **man man**.

A leitura das páginas man é geralmente feita através de um terminal, quando se está em um ambiente gráfico, ou no modo texto para os que o preferem. Digitando-se o comando no prompt, seguido de **Enter**:

```
seunome@seucomp ~-> man man
```

Lança na tela a documentação para o **man**, após ser pressionada a tecla **Enter**:

```
man(1) man(1)
```

### NOME

```
man - formata e exibe as páginas de manual online
manpath - determina o caminho de busca do usuário por páginas man
```

### SINOPSE

```
man [-acdfFhkKtW] [--path] [-m system] [-p string] [-C config_file]
[-M pathlist] [-P pager] [-S section_list] [section] nome ...
```

### DESCRIÇÃO

O man formata e exibe as páginas de manual online. Se você especifica a seção, o man somente buscará a seção apropriada. Nome é normalmente o nome da página de manual, que é em geral o nome de um comando, função, ou arquivo. Entretanto, se o nome tiver uma barra (/) então man a interpreta como uma especificação de arquivo, de forma que se pode usar man ./foo.5 ou mesmo man /cd/foo/bar.1.gz.

Veja abaixo uma descrição de onde o man procura pelos arquivos de páginas de manual.

### OPÇÕES

```
-C config_file
linhas 1-27
```

Navegue para a próxima página usando a barra de espaço. A tecla b serve para retornar à página anterior. Quando for atingido o final, o **man** normalmente se encerra e remete o usuário devolta ao prompt. Caso este comportamento não seja o padrão ou se deseje abandonar a página man antes de atingir o seu final, pode-se fazê-lo teclando **q** a qualquer momento.

## Paginadores

A combinação de teclas usadas para manipular as páginas man depende do *paginador* ("pager") usado na distribuição escolhida. A maior parte das distribuições usa o **less** para ver as páginas man e navegar por elas. Veja [the section called "less is more"](#) para saber mais sobre os paginadores.

Cada página man geralmente contém um conjunto padrão de seções, tal como podemos observar através do exemplo **man man**:

- A primeira linha contém o nome do comando sobre o qual se está lendo, bem como a identificação da seção em que se localiza a página man. As páginas estão ordenadas em capítulos. Em geral, os comandos têm múltiplas páginas man, por exemplo, uma para a seção do usuário, uma para a seção do administrador do sistema e uma para a seção de programadores.
- O nome do comando e uma descrição curta são dados, o que se usa para construir um índice das páginas man. É possível buscar por qualquer seqüência neste índice usando o comando **apropos**.
- A sinopse do comando proporciona uma nota técnica sobre todas as opções e/ou argumentos aceitos pelo comando. Pode-se pensar na opção como um modo de executar o comando. O argumento é aquilo sobre o que o comando é executado. Alguns comandos não aceitam opções ou argumentos. Opções ou argumentos facultativos são colocados entre "[" e "]" como forma de indicar que eles podem ser omitidos.
- Uma descrição detalhada do comando é dada.
- São listadas as opções com as suas descrições. Normalmente as opções podem ser combinadas. Quando não puderem, a seção explicitamente indicará.

- Ambiente são as variáveis do shell que influenciam o comportamento deste comando (nem todos os comandos têm esta seção).
- Ocorre também de haver seções específicas para o comando.
- Referências para outras páginas man são dadas na seção "VEJA TAMBÉM". Entre parênteses são fornecidos os números das seções de páginas man em que se encontra o comando. Usuários experientes costumam ir para a parte do "VEJA TAMBÉM" usando o comando / seguido da sequência de busca VEJA, seguido de **Enter**.
- Normalmente também se fornece informação a respeito de erros conhecidos (anomalias) e onde informar novos erros que venham a ser descobertos.
- Pode também haver informação sobre o autor e os direitos autorais.

Alguns comandos possuem múltiplas páginas man. Por exemplo, o comando **passwd** tem uma página man na seção 1 e outra na seção 5. O padrão é que seja exibida a página man com o menor número. Caso se queira ver outra seção que não a padrão, deve-se especificar o seu número após o comando **man**:

```
man 5 passwd
```

Caso se queira acessar todas as páginas man sobre um comando, uma após a outra, usa-se o man com a -a

```
man -a passwd
```

Desta forma, quando for alcançado o final da primeira página man e pressionada a **barra de espaço** novamente, a página man da próxima seção será exibida.

## Mais informações

### As páginas Info

Além das páginas man, é possível ler as páginas Info sobre um comando, o que é feito através do comando **info**. Estas páginas em geral contém informações mais recentes e são um pouco mais fáceis de usar. As páginas man de alguns comandos fazem referência às páginas Info.

Comece teclando **info info** em um terminal:

```
Arquivo: info.info, Node: Topo, Próximo: Básico, Cima: (dir)
```

```
Info: Uma Introdução
*****
```

O Info é um programa, que você está usando agora, feito para ler a documentação de programas de computador. O Projeto GNU distribui a maior parte de seus manuais online no formato Info, de modo que é necessário usar um programa chamado "leitor Info" para lê-los. Como o programa que você está usando agora.

Se você é novo no Info e quer aprender a usá-lo, digite o comando `h' agora. Isto exibe uma sequência de instruções programadas.

Para aprender comandos avançados do Info, digite `n' duas vezes. Isto leva você ao `Info para Experts', pulando o capítulo `Básico'.

\* Menu:

```
* Básico::          \t Aprendendo a usar um leitor Info.
* Info Avançado::   Comandos avançados para o Info.
* Criando um Arquivo Info:: Como criar seu próprio arquivo Info.
--zz-Info: (info.info.gz)Topo, 24 linhas --Topo-----
Bem vindo ao Info versão 4.2. Digite C-h para obter ajuda, e m para menu.
```

Use as setas do teclado para navegar pelo texto e mover o cursor em uma linha iniciada por um asterisco, contendo a palavra-chave sobre a qual você deseja saber, então aperte **Enter**. Use as teclas **P** e **N** para ir para o tópico anterior ou posterior. A barra de espaço muda para a próxima página, indiferentemente desta ser um novo tópico ou uma página Info de outro comando. Use **Q** para sair. O programa **info** tem mais informações.

### Os comandos whatis e apropos

Uma explicação resumida para os comandos pode ser acessada usando o **whatis**, como nos exemplos a seguir:

```
[seu_prompt] whatis ls
ls          (1) - lista o conteúdo de um diretório
```

Isto mostra um breve resumo sobre o comando e a primeira seção na coleção de páginas man que contém uma página apropriada.



Se você não sabe por onde começar e qual página man deve ler, **apropos** fornece mais informações. Por exemplo, se você não souber como iniciar um navegador, então seria possível usar o seguinte comando:

```
outro_prompt> apropos navegador
Galeon [galeon](1) - navegador GNOME baseado no gecko
lynx (1) - um navegador genérico de informações distribuídas
para a World Wide Web
ncftp (1) - Um navegador para o File Transfer Protocol
opera (1) - um navegador gráfico
pilot (1) - um navegador de sistema no mesmo estilo do
Pine Composer
pinfo (1) - um navegador baseado em curses e no estilo lynx
pinfo [pman] (1) - um navegador baseado em curses e no estilo lynx
viewres (1x) - navegador gráfico de classe para o Xt
```

Após teclar **Enter** você descobrirá que existe muita coisa relacionada a navegadores na sua máquina: não apenas navegadores web, mas também navegadores de arquivos e FTP, bem como navegadores de documentos. Se houver pacotes de desenvolvimento instalados, provavelmente também haverá páginas man relativas a programação relacionada a navegadores. Em geral, um comando com página man na seção um, portanto, marcada com "(1)", é adequado a experimentações por um usuário. O usuário que teclar o comando acima **apropos**, provavelmente tentará em seguida iniciar os comandos **galeon**, **lynx** ou **opera**, vez que eles têm a ver com navegar pela web.

## A opção --help

A maior parte dos comandos do GNU aceita `--help`, que fornece uma explicação resumida sobre a forma de uso do comando e uma lista das opções que são aceitas. Abaixo encontra-se a saída desta opção para o **cat** command:

```
promptusuario@maquina: cat --help
Uso: cat [OPÇÃO] [ARQUIVO]...
Concatena ARQUIVO(s), ou entrada padrão, para saída padrão.
```

```
-A, --show-all          equivalente a -vET
-b, --number-nonblank   numera as linhas não-vazias de saída
-e                      equivalente a -vE
-E, --show-ends        exibe $ ao final de cada linha
-n, --number            numera todas as linhas de saída
-s, --squeeze-blank    nunca mais de uma linha vazia
-t                      equivalente a -vT
-T, --show-tabs        exibe caracteres TAB como ^I
-u                      (ignorado)
-v, --show-nonprinting usa as notações ^ e M-,
                       exceto para LFD e TAB
--help                 exibe esta ajuda e sai
--version              exibe a versão e sai
```

Sem Arquivo, ou quando ARQUIVO é -, lê a entrada padrão.

Relate erros para <bug-textutils@gnu.org>.

## Ajuda gráfica

Não se desespere caso você prefira uma interface gráfica. O Konqueror, que é o gerenciador de arquivos padrão do KDE, provê um acesso colorido e indolor às páginas man e Info. Digitando-se "info:info" no *Local* da barra de endereços, o Konqueror apresentará uma página navegável sobre o comando **info**. Da mesma maneira, "man:ls" apresenta a página man para o comando **ls**. Consegue-se até mesmo usar o recurso de auto-completar: serão apresentados todos os comandos iniciados por "ls" em um menu em cascata. Informar "info:/dir" na barra de endereços faz com que o navegador apresente todas as páginas Info, elencadas em categorias de utilidade. Um excelente material de Ajuda, incluindo um guia prático para o Konqueror, encontra-se incluso. Pode-se iniciar o programa a partir do menu ou digitando-se **konqueror** em um terminal, seguido de **Enter**; veja uma captura de tela abaixo.

## Figure 2.2. Konqueror como um navegador de ajuda



O Navegador de Ajuda do Gnome também é muito amigável. Pode-se iniciá-lo selecionando Aplicativos → Ajuda do menu do Gnome, clicando no ícone de salva-vidas em seu desktop ou informando o comando **gnome-help** em um terminal. A documentação do sistema e as páginas man são facilmente navegáveis e têm uma interface simples.

O gerenciador de arquivos **nautilus** um índice das páginas man e Info, que é interligado e de fácil navegação. O Nautilus pode ser iniciado a partir da linha de comando, clicando no ícone do diretório "Home" ou a partir do menu do Gnome.

A grande vantagem das "GUIs" (interfaces gráficas) para a documentação do sistema é que a informação encontra-se completamente interligada, de forma que é possível clicar na seção "VEJA TAMBÉM" e quaisquer links para outras páginas man que aparecerem e, assim, navegar e adquirir conhecimento sem interrupção por horas a fio.

## Exceções

Alguns comandos não têm documentação individual, porque são parte de outros comandos. **cd**, **exit**, **logout** e **pwd** são exemplos destas exceções. Eles são parte do seu programa shell e recebem a alcunha de *comandos internos do shell* ("built-in"). Para obter informações sobre tais comandos é necessário acessar a página man ou info do shell. A maior parte dos usuários iniciantes do Linux usam a shell Bash. Veja [the section called "O shell"](#) para saber mais sobre shells.

Se você tiver mudado as configurações originais do seu sistema, é possível que as páginas man ainda encontrem-se presentes, porém invisíveis por conta da alteração em seu ambiente shell. Neste caso, é preciso verificar a variável MANPATH. Na [the section called "Exportando variáveis"](#) é explicado como fazer isto.

Alguns programas ou pacotes têm somente um conjunto de instruções ou referências no diretório /usr/share/doc. Veja a [the section called "Outros modos de averiguar o conteúdo de arquivos"](#) sobre exibição.

Na pior das hipóteses, é possível que você tenha apagado a documentação do seu sistema por acidente (tomara que por acidente, porque fazer isto de propósito é uma péssima idéia). Neste caso, primeiro certifique-se de que não sobrou nada usando uma ferramenta de busca, leia na [the section called "Encontrando arquivos"](#). Se assim for, é possível que você tenha que reinstalar o pacote que contém o comando ao qual se aplica a documentação, veja a [the section called "Instalando novos programas"](#).

## Resumo

Tradicionalmente o Linux opera tanto no modo texto como no gráfico. Uma vez que a potência dos CPUs e das memórias RAM não são mais o custo atualmente, todos os usuários Linux podem usá-lo no modo gráfico e geralmente o fazem. O que não implica que ser desnecessário saber usá-lo no modo texto: trabalharemos no modo texto ao longo deste curso, utilizando um terminal.

O Linux encoraja os seus usuários a adquirir conhecimento e tornarem-se independentes. Inevitavelmente, é necessária uma vasta leitura de documentos para alcançar tal objetivo; este é o motivo pelo qual, como você irá notar, iremos fazer referência a outros documentos para quase todos os comandos, ferramentas e problemas listados neste livro. Quanto mais documentos você ler, mais fácil será fazê-lo e mais rápido você os lerá. Torne a leitura da documentação um hábito o quanto antes. Quando não souber a resposta para um problema, buscá-la na documentação deve tornar-se a sua segunda natureza.

Nós já aprendemos alguns comandos:

**Table 2.3. Novos comandos no capítulo 2: o Básico**

Comando	Significado
<b>apropos</b>	Procura mais informações sobre um comando ou tópico.
<b>cat</b>	Exibe o conteúdo de um ou mais arquivos.
<b>cd</b>	Muda para outro diretório.
<b>exit</b>	Encerra uma sessão do shell.
<b>file</b>	Obtém informações sobre o conteúdo de um arquivo.
<b>info</b>	Lê páginas Info sobre um comando.
<b>logout</b>	Encerra uma sessão do shell.
<b>ls</b>	Lista o conteúdo de um diretório.
<b>man</b>	Lê as páginas de manual sobre um comando.
<b>passwd</b>	Muda a senha.
<b>pwd</b>	Mostra qual o diretório atual.

## Exercícios

Nós aprendemos muitas coisas errando e percebendo como as coisas podem dar errado. Estes exercícios foram elaborados para que você leia algumas mensagens de erro. A ordem em que eles estão apresentados é importante.

Não se esqueça de utilizar os recursos do Bash na linha de comando: tente fazer os exercícios usando a menor quantidade de caracteres possível!

### Conectando e desconectando

- Determine qual o ambiente você está trabalhando, texto ou gráfico.

Estou trabalhando no modo texto/gráfico. (risque o que não for aplicável)

- Efetue o login com o nome de usuário e senha de acesso criados durante a instalação.
- Efetue o logout.

- Efetue o login novamente, usando um nome de usuário inexistente

-> O que acontece?

## Senhas de acesso

Efetue o login novamente com o seu nome de usuário e senha de acesso.

- Mude a sua senha de acesso para *P6p3.aa!* e pressione a tecla **Enter**.

-> O que acontece?

- Tente novamente, mas desta vez com uma senha de acesso que seja ridiculamente fácil, como *123* ou *aaa*.

-> O que acontece?

- Tente novamente, mas desta vez não forneça uma senha de acesso, pressionando somente a tecla **Enter**.

-> O que acontece?

- Tente o comando **psswd** ao invés de **passwd**

-> O que acontece?

### Nova senha de acesso

A menos que você mude novamente a sua senha de acesso para a original, ela continuará a ser "P6p3.aa!". Mude a senha de acesso após concluir este exercício.

Note que alguns sistemas não permitem a reciclagem de senhas de acesso, p. ex., retornar à senha original antes de decorrido certo tempo ou quantidade de mudanças de senhas ou ambos.

## Diretórios

Estes são alguns exercícios para ajudá-lo a sentir como é o Linux.

- Informe o comando **cd blah**

-> O que acontece?

- Informe o comando **cd ..**

Observe que há um espaço entre "cd" e ".."! Use o comando **pwd**.

-> O que acontece?

- Liste o conteúdo de um diretório com o comando **ls**.

-> O que você percebe?

-> O que você acha que eles são?

-> Verifique usando o comando **pwd**.

- Informe o comando **cd**.

-> O que acontece?

- Repita o passo 2 duas vezes.

-> O que acontece?

- Exiba o conteúdo deste diretório.

- Tente o comando **cd root**

-> O que acontece?

-> A quais diretórios você tem acesso?

- Repita o passo 4.

Você conhece alguma outra possibilidade para chegar onde você está agora?

## Arquivos

- Mude para o diretório `/` e então para o `etc`. Digite **ls**; se a saída for maior do que a sua tela, torne a janela maior ou tente **Shift+PageUp** e **Shift+PageDown**.

O arquivo `inittab` contém a resposta para a primeira pergunta desta lista. Tente usar o comando **file** nele.

-> O tipo de arquivo do meu `inittab` é ....

- Use o comando **cat inittab** e leia o arquivo.

-> Qual é o modo padrão no seu computador?

- Retorne ao seu diretório `home` usando o comando **cd**.

- Informe o comando **file** .

-> Isto ajuda você a descobrir o significado de "."?

- É possível olhar no "." usando o comando **cat**?

- Exiba a ajuda para o programa **cat**, usando a opção `--help`. Use a opção de numerar as linhas de saída para contar a quantidade de usuário listados no arquivo `/etc/passwd`.

## Buscando ajuda

- Leia **man intro**

- Leia **man ls**

- Leia **info passwd**

- Informe o comando **apropos pwd**.

- Tente os comandos **man** e **info** no **cd**.

-> O que fazer para descobrir mais sobre o **cd**?

- Leia **ls --help** e tente usá-lo de várias formas.

## Chapter 3. Dos arquivos e do sistema de arquivos

### Table of Contents

[Panorama geral do sistema de arquivos do Linux](#)

[Arquivos](#)

[Do particionamento](#)

[Mais sobre a disposição do sistema de arquivos](#)

[Orientação no sistema de arquivos](#)

[O caminho \(path\)](#)

[Caminhos absolutos e relativos](#)

[Os arquivos e diretórios mais importantes](#)

[Os arquivos de configuração mais importantes](#)

[Os dispositivos mais comuns](#)

[Os arquivos de variáveis mais comuns](#)

[Manipulando arquivos](#)

[Exibindo as propriedades de um arquivo](#)

[Criando e deletando arquivos e diretórios](#)

[Encontrando arquivos](#)

[Outros modos de averiguar o conteúdo de arquivos](#)

[Interligando arquivos](#)

[Segurança de arquivos](#)

[Direitos de acesso: A primeira linha de defesa do Linux](#)

[As ferramentas](#)

[Resumo](#)

[Exercícios](#)

[Partições](#)

[Caminhos](#)

[Passeio pelo sistema](#)

[Manipulando arquivos](#)

[Permissões de arquivo](#)

### Abstract

Após a abordagem inicial no [Chapter 2, Início Rápido](#), estamos prontos para discutir os arquivos e diretórios em um sistema Linux em detalhes. Muitos usuários têm dificuldades com o Linux porque

carecem de uma visão panorâmica que lhes indique quais dados cada local armazena. Tentaremos, então, trazer alguma luz sobre o tema.

Listaremos também os arquivos e diretórios mais importantes e apresentaremos diferentes métodos para a exibição do conteúdo destes arquivos. Aprenderemos, ainda, como criar, mover e apagar arquivos.

Após completar os exercícios neste capítulo você será capaz de:

- Descrever a estrutura do sistema de arquivos no Linux
- Exibir e configurar caminhos
- Descrever os arquivos mais importantes, incluindo o kernel e o shell
- Encontrar arquivos desaparecidos ou ocultos
- Criar, mover e apagar arquivos e diretórios
- Exibir o conteúdo dos arquivos
- Compreender e usar diferentes tipos de links
- Aprender sobre as propriedades do arquivos e suas permissões

## Panorama geral do sistema de arquivos do Linux

### Arquivos

#### Geral

Um descrição simples do sistema UNIX, também aplicável ao Linux, é esta:

“Em um sistema UNIX, todas as coisas são arquivos; se algo não é um arquivo, então é um processo.”

Esta afirmação não é verdadeira porque existem arquivos especiais que são mais do que arquivos (named pipes e sockets, por exemplo), mas para manter as coisas simples, dizer que tudo é arquivo é uma generalização aceitável. Um sistema Linux, assim como o UNIX, não diferencia arquivos e diretórios, vez que um diretório não passa de um arquivo contendo nomes de outros arquivos. Programas, serviços, textos, imagens e assim por diante, todos são arquivos. Dispositivos de entrada e saída e, em geral, quaisquer outros aparelhos são também considerados arquivos.

A fim de gerenciar todos estes arquivos de uma forma ordenada, é útil encará-los como se estivessem organizados em uma estrutura semelhante a um árvore no disco rígido, assim como acontece no MS-DOS (Sistema Operacional de Disco). Os grandes galhos contém galhos menores e os galhos menores contém as folhas ou arquivos normais. Por enquanto, recorreremos a esta imagem de uma árvore. Mais adiante, porém, aprenderemos sobre as inexactidões desta metáfora.

#### Tipos de arquivos

ME sua maioria os arquivos são apenas isto e são denominados arquivos *normais*; eles contém dados usuais, como, por exemplo, arquivos-texto, arquivos autoexecutáveis ou programas, entrada para ou saída de um programa e assim por diante.

Embora seja razoavelmente seguro supor que tudo o que você encontra em um sistema Linux é um arquivo, existem algumas exceções.

- *Diretórios*: arquivos que listam outros arquivos.
- *Arquivos especiais*: o mecanismo usado para entrada e saída. Grande parte dos arquivos especiais estão no `/dev` e os discutiremos mais adiante.
- *Links*: um sistema para tornar um arquivo ou diretório visível em múltiplas partes da árvore de arquivos. Discutiremos os links em detalhes.
- *Sockets (de domínio)*: um tipo especial de arquivo, semelhante aos sockets de TCP/IP, provê uma de rede de processos protegida pelo controle de acesso do sistema.
- *Named pipes*: atuam mais ou menos como sockets e perfazem um modo para que os processos se comuniquem uns com os outros, sem que seja necessária a semântica de sockets de rede.

A opção `-l` ao comando `ls` exibe o tipo de arquivo usando o primeiro caracter de cada linha de entrada:

```
jaime:~/Documents> ls -l
total 80
-rw-rw-r-- 1 jaime jaime 31744 Feb 21 17:56 intro Linux.doc
-rw-rw-r-- 1 jaime jaime 41472 Feb 21 17:56 Linux.doc
drwxrwxr-x 2 jaime jaime 4096 Feb 25 11:50 course
```

Esta tabela fornece um panorama dos caracteres determinando o tipo de arquivo:

**Table 3.1. Tipos de arquivos em uma lista longa**

Símbolo	Significado
-	Regular file
d	Diretório
l	Link
c	Arquivo especial
s	Socket
p	Named pipe
b	Dispositivo em bloco

A fim de não ser forçado a exibir uma lista longa para saber o tipo de arquivo, o padrão em muitos sistemas é não informar apenas **ls**, mas **ls -F**, que apõe aos nomes dos arquivos um dos caracteres `/=*|@` para indicar o tipo de arquivo. Para tornar as coisas ainda mais fáceis para o iniciante, ambos `-F` e `--color` são informados, veja [the section called "Mais sobre o ls"](#). Usaremos sempre o **ls -F** ao longo deste documento a fim de facilitar a leitura.

Como um usuário, você somente precisará lidar diretamente com arquivos simples, autoexecutáveis, diretórios e links. Os tipos especiais de arquivos estão lá somente para que o seu sistema faça aquilo que você quer que ele faça e são manipulados por administradores de sistemas e programadores.

Agora, antes de aprendermos sobre os arquivos e diretórios importantes, é preciso que saibamos um pouco sobre partições.

## Do particionamento

### Por que particionar?

A maior parte das pessoas tem uma vaga idéia do que são partições, já que qualquer sistema operacional é capaz de criá-las e removê-las. Pode parecer estranho que o Linux use mais de uma partição em um mesmo disco, mesmo quando se opta pelo procedimento padrão de instalação, de modo que se faz necessária uma explicação.

Um dos objetivos de ter diferentes partições é obter uma maior segurança para os dados no caso de um desastre. Dividindo o seu disco rígido em partições, os dados podem ser agrupados e separados. Quando ocorrer um acidente, apenas os dados na partição onde ocorreu o acidente é que serão afetados, enquanto os dados nas demais partições permanecerão intactos.

Este princípio data de uma época em que o Linux ainda não possuía um sistema de arquivos journaled, época em que falhas de energia podiam causar desastres. O uso de partições ainda permanece por razões de segurança e robustez, de forma que uma ruptura em uma parte do sistema não implique que o sistema inteiro seja exposto ao perigo. Esta é atualmente a principal razão para que se faça o particionamento. Um exemplo simples: um usuário cria um script, um programa ou aplicativo web que começa a encher o disco. Se nele existir somente uma partição, o sistema inteiro parará de funcionar quando o disco ficar cheio. Por outro lado, se o usuário tiver uma partição própria, então apenas os dados nesta partição é que serão afetados, enquanto a partição direcionada a arquivos do sistema e outros dados permanecerão intocadas.

Tenha em mente que ter um sistema de arquivos journaled provê segurança aos dados apenas para o caso de quedas de energia e desligamento súbito de dispositivos de armazenamento. Ele não fornece nenhum tipo de proteção contra blocos ruins e erros lógicos no sistema de arquivos. Para enfrentar estes problemas é aconselhável usar uma solução RAID (Redundant Array of Inexpensive Disks).

### Disposição e tipos de partições

Existem dois tipos principais de partições em um sistema Linux:

- *partição de dados*: dados usuais do sistema Linux, aqui inclusa a *partição root*, que contém todas as informações necessárias à inicialização e execução do sistema; e
- *partição swap*: uma expansão da memória física do computador, alocação disco rígido para a memória.

A maior parte dos sistemas contém uma partição root, uma ou mais partições de dados e uma ou mais partições de swap. Sistemas em ambientes mistos podem possuir partições destinadas aos dados de outros sistemas, como, por exemplo, partições FAT ou VFAT para dados do MS Windows.

A maioria dos sistemas Linux usa o **fdisk** quando da instalação para configurar o tipo de partição. Como você deve ter notado ao longo dos exercícios do Capítulo 1, isto normalmente acontece automaticamente. Em algumas situações, contudo, você pode não ter tanta sorte. Nestas hipóteses, você precisará selecionar os tipos de partições manualmente ou mesmo realizar o particionamento à mão. As partições Linux padrão possuem o número 82 para swap e 83 para dados, as quais, por sua vez, podem ser journaled (ext3) ou convencionais (ext2, em sistemas mais antigos). O utilitário **fdisk** tem um recurso embutido de ajuda, caso você se esqueça destes valores.

Além destes dois, o Linux também suporta uma larga gama de outros sistemas de arquivos, como, por

exemplo, o Reiser, o JFS, o NFS, o FATxx, bem como outros sistemas de arquivos disponíveis em outros sistemas operacionais (proprietários).

A partição root padrão (representada com uma barra simples, /) possui em torno de 100/500MB, e contém os arquivos de configuração do sistema, os comandos mais básicos e programas de servidor, as bibliotecas do sistema, espaço temporário e o diretório home do usuário administrador. Uma instalação padrão requer em torno de 250MB para a partição root.

O espaço de swap (representado por *swap*) somente pode ser acessado pelo próprio sistema, encontrando-se oculta durante uma operação normal. A swap é o sistema que garante, como ocorre em sistemas UNIX, que você pode continuar trabalhando, não importa o que aconteça. No Linux, você jamais verá mensagens como *Sem memória, por favor encerre alguns aplicativos e então tente novamente*, por conta desta memória adicional. Atualmente diversos outros sistemas operacionais fora do mundo UNIX já também adotaram a técnica da swap ou memória virtual.

Naturalmente, o uso da memória no disco rígido é mais lento do que o do chip de memória de um computador, mas este pequeno adicional é um grande conforto. Aprenderemos mais sobre a swap quando discutirmos os processos no [Chapter 4, Processos](#).

O Linux em geral espera ter o dobro da memória física na forma de swap. Quando da instalação do seu sistema convém que você saiba como fazer isto. Um exemplo em um sistema com 512MB de RAM:

- 1ª possibilidade: uma partição swap de 1GB
- 2ª possibilidade: duas partições swap de 512MB
- 3ª possibilidade: com dois discos rígidos: 1 partição de 512MB em cada um.

A última opção é a que fornece melhores resultados quando se espera realizar muitas operações de entrada e saída.

Leia a documentação do seu software para diretrizes específicas. Alguns aplicativos como bases de dados, podem precisar de mais swap. Outras, como alguns sistemas portáteis, podem dispensar a swap por falta de espaço em disco. O espaço de swap pode também depender da versão do seu kernel.

O kernel também está em uma partição separada em muitas distribuições, vez que ele é o arquivo mais importante no seu sistema. Neste caso, você descobrirá que também possui uma partição */boot*, guardando o seu kernel e os arquivos que lhe acompanham.

O restante do(s) disco(s) rígido(s) é normalmente dividido em partições de dados, embora pode bem ser que todos os dados não-críticos ao sistema residam em uma só partição, por exemplo, quando se instala uma estação de trabalho padrão. Quando os dados não-críticos são separados em diferentes partições, isto normalmente é feito segundo um padrão pré-estabelecido.

- uma partição para os programas de usuário (*/usr*)
- uma partição contem os dados pessoais dos usuários (*/home*)
- uma partição para o armazenamento temporário de dados como filas de email ou impressão (*/var*).
- uma partição para softwares de terceiros ou extras (*/opt*)

Criadas as partições, somente é possível criar mais. Mudar o tamanho ou propriedades das partições é possível, mas não é aconselhável.

A divisão dos discos rígidos em partições é determinada pelo administrador do sistema. Em sistemas maiores, ele(a) pode até mesmo decidir acomodar uma única partição em vários discos rígidos, usando os softwares apropriados. A maior parte das distribuições permite configurações padrão otimizadas para estações de trabalho (usuário médio) e também para servidores, mas também aceita partições personalizadas. Ao longo do processo de instalação você pode definir o formato de suas partições usando a ferramenta específica de sua distribuição, que normalmente será um aplicativo gráfico bem intuitivo ou com o **fdisk**, uma ferramenta-texto para a criação de partições e configuração de suas propriedades.

Uma estação de trabalho ou instalação cliente é destinada ao uso de uma única pessoa. Os softwares selecionados para a instalação são reflexo desta presunção e buscam os pacotes de usuário mais comuns, como temas de desktop agradáveis, ferramentas de desenvolvimento, clientes de email, softwares de multimídia, web e outros serviços. Tudo é acomodado em uma única partição, espaço de swap com o dobro da RAM é adicionado e a sua estação de trabalho genérica está completa, fornecendo a maior quantidade de espaço em disco possível para uso pessoal, porém com a desvantagem de perda da integridade dos dados em situações problemáticas.

Em um servidor, os dados do sistema são alocados em uma área separada daquela destinada aos dados de usuários. Os programas que fornecem serviços são mantidos em um local diferente daquele em que são guardados os dados manipulados por este serviço. Diferentes partições serão criadas em tais sistemas:

- uma partição com todos os dados necessários à inicialização do sistema
- uma partição com os dados de configuração e os programas de servidor

- uma ou mais partições contendo os dados dos serviços como as tabelas com bases de dados, emails de usuários e arquivos ftp e etc.
- uma partição com os programas de usuário e aplicativos
- uma ou mais partições para os arquivos específicos de usuários (diretório home)
- uma ou mais partições swap (memória virtual)

Servidores em geral possuem mais memória e mais espaço de swap. Certos processos de servidor, como bancos de dados, podem requerer mais espaço swap do que o normal; procure saber mais na documentação específica. Para uma melhor performance, a swap normalmente é dividida em duas partições distintas.

## Pontos de montagem

Todas as partições são acopladas ao sistema por meio de um ponto de montagem. O ponto de montagem define o local destinado a um conjunto particular de dados no sistema de arquivos. Normalmente, todas as partições estão conctadas por meio da partição *root*. Nesta partição, que é indicada por uma barra (/), diretórios são criados. Estes diretórios vazios serão o ponto de partida das partições que estão acopladas a eles. Um exemplo: dada uma partição que armazena os seguintes diretórios:

```
videos/           cd-images/       pictures/
```

Nós queremos acoplar esta partição no sistema de arquivos em um diretório chamado */opt/media*. Para fazer isto, o administrador do sistema tem que garantir que o diretório */opt/media* existe no sistema. De preferência, ele deve ser um diretório vazio. Como isto é feito será explicado mais adiante neste capítulo. Então, usando o comando **mount**, o administrador pode acoplar a partição ao sistema. Quando você olhar para o conteúdo do diretório anteriormente vazio **mount**, ele irá conter os arquivos e diretórios que estão no meio montado (disco rígido ou partição de um disco, CD, DVD, cartão flash, USB ou outro dispositivo de armazenamento).

Ao longo da inicialização do sistema, todas as partições são montadas, como descrito no arquivo */etc/fstab*. Algumas partições não são montadas por padrão, por exemplo quando elas não estão conectadas sempre ao sistema, como é o caso do espaço de armazenamento da sua câmera digital. Se bem configurado, o dispositivo será montado assim que o sistema perceber que ele está conectado, ou ele pode ser montado pelo usuário, p. ex., não é necessário privilégio administrativo para fazê-lo. Há um exemplo no [the section called "Usando o rsync"](#).

Em um sistema em execução, informações sobre as partições e seus pontos de montagem podem ser exibidos usando o comando **df** (sigla para *disco cheio* - full no inglês - ou *disco livre* - free no inglês). No Linux, o **df** é a versão GNU e aceita a opção **-h** ou *inteligível por humanos* que facilita muito a sua legibilidade. Perceba que todas as máquinas UNIX comerciais normalmente possuem suas próprias versões do **df** e muitos outros comandos. O seu comportamento é em geral o mesmo, embora as versões GNU das ferramentas mais comuns possuem mais e melhores recursos.

O comando **df** apenas exibe informações sobre partições ativas e não-swap. Aqui estão inclusas as partições de outros sistemas em rede, como no exemplo abaixo, em que os diretórios home estão montados em um arquivo de um servidor na rede, uma situação recorrente no ambiente empresarial.

```
freddy:~> df -h
Sistema de Arquivos  Tam  Uso  Dispon  Uso%  Montada em
/dev/hda8            496M 183M 288M    39%   /
/dev/hda1            124M 8.4M 109M    8%    /boot
/dev/hda5            19G  15G  2.7G   85%   /opt
/dev/hda6            7.0G 5.4G 1.2G   81%   /usr
/dev/hda7            3.7G 2.7G 867M   77%   /var
fs1:/home            8.9G 3.7G 4.7G   44%   /.automont/fs1/root/home
```

## Mais sobre a disposição do sistema de arquivos

### Visual

Por conveniência, o sistema de arquivos do Linux é comumente pensado como em uma estrutura de árvore. Em um sistema Linux padrão você perceberá que a disposição via de regra segue o esquema apresentado abaixo.

**Figure 3.1. Disposição do sistema de arquivos no Linux**



Esta é a apresentação de um sistema RedHat. Dependendo do administrador do sistema, deste e da missão da máquina UNIX, a estrutura pode variar, e alguns diretórios podem ser deixados de lado ou adicionados livremente. Os nomes não são sequer obrigatórios, mas mera convenção.

A árvore do sistema de arquivos começa no tronco ou *barra*, indicada por uma barra (/). Este diretório, contendo todos os diretórios e arquivos subjacentes, também é chamado de *diretório root* ou "a raiz" do sistema de arquivos.



Diretórios que são apenas um nível abaixo do diretório raiz são precedidos por uma barra, para indicar sua posição e evitar confusão com outros diretórios que podem ter o mesmo nome. Quando se começa com um novo sistema, é sempre um boa idéia dar uma olhada na pasta root. Vamos ver com o que você pode se deparar:

```
emmy:~> cd /
emmy: /> ls
bin/  dev/  home/  lib/      misc/  opt/    root/  tmp/  var/
boot/ etc/  initrd/ lost+found/ mnt/  proc/   sbin/  usr/
```

**Table 3.2. Subdiretórios do diretório raiz**

Diretório	Conteúdo
/bin	Programas comuns, compartilhados pelo sistema, pelo administrador e pelos usuários.
/boot	Os arquivos de inicialização e o kernel, vmlinuz. Em algumas distribuições mais recentes, também dados grub. O Grub é o carregador GRand Unified Boot e é uma tentativa de descartar todos os demais carregadores que conhecemos hoje em dia.
/dev	Contém referências para todos os periféricos da CPU, que são representados como arquivos com propriedades especiais.
/etc	Os arquivos de configuração mais importantes do sistema estão no /etc, este diretório contém dados semelhantes àqueles encontrados no Painel de Controle do Windows.
/home	Diretórios home dos usuários comuns.
/initrd	(em algumas distribuições) Informações para inicialização. Não remova!
/lib	Arquivos de bibliotecas, inclui arquivos para todos os tipos de programas precisos pelo sistemas e usuários.
/lost+found	Todas as partições têm um lost+found em seu diretório superior. Arquivos que são salvos durante falhas ficam guardados aqui.
/misc	Para propósitos variados
/mnt	Ponto de montagem padrão para sistemas de arquivos externos, p. ex., um CD-ROM ou uma câmera digital.
/net	Ponto de montagem padrão para sistemas de arquivos remotos.
/opt	Em geral, contém softwares extras ou de terceiros.
/proc	Um sistema de arquivos virtual contendo informações sobre os recursos do sistema. Mais informações sobre o significado dos arquivos no proc pode ser obtida informando o comando <b>man proc</b> em uma janela de terminal. O arquivo proc.txt discute o sistema de arquivos virtual em detalhes.
/root	O diretório home do usuário administrativo. Note a distinção entre /, o diretório root e /root, a pasta home do usuário root
/sbin	Programas para uso do sistema e do administrador do sistema.
/tmp	Espaço temporário para uso do sistema, é limpo quando da reinicialização, portanto, não use para armazenar nenhum tipo de trabalho.
/usr	Programas,, bibliotecas, documentação e etc para todos os programas de usuários.
/var	Espaço de armazenamento de todos os arquivos variáveis e temporários criados pelos usuários, como os arquivos de log, fila de email, área de spooler de impressão, espaço para armazenamento temporário de arquivos baixados da internet ou para gravar a imagem de um CD antes de gravá-la no disco.

Como descobrir em qual partição um diretório está? Usando o comando **df** com um ponto (.) de opção exibe a partição a que pertence o diretório atual e informa sobre a quantidade de espaço em uso na partição:

```
sandra:/lib> df -h .
Sistema de arquivos Tam  Uso  Dispon  Uso%  Montado em
/dev/hda7          980M 163M  767M   18%  /
```

Como regra geral, todos os diretórios dentro do diretório raiz estão na partição root, a não ser que ele possua uma entrada em separado na lista completa do comando **df** (ou **df -h** sem outras opções).

Leia mais em **man hier**.

## O sistema de arquivos na realidade

Para a maior parte dos usuários e para as tarefas mais comuns de um administrador de sistemas, é suficiente aceitar que os arquivos e diretórios estão ordenados em uma estrutura de árvore. O computador, contudo, não entende absolutamente nada sobre árvores ou estruturas em árvore.

Toda partição possui seu próprio sistema de arquivos. Imaginando todos estes sistemas de arquivos juntos, podemos ter uma idéia de um sistema em estrutura de árvore, mas ele não é tão simples assim. Em um sistema de arquivos, um arquivo é representado por um *inode*, uma espécie de número de série contendo informações sobre os dados que o arquivo contém: a quem ele pertence e em que local do disco ele está armazenado.

Toda partição possui seu próprio conjunto de inodes; em um sistema com múltiplas partições, podem existir arquivos com o mesmo número inode.

Cada inode descreve a estrutura dos dados no disco rígido, armazenando as propriedades de um arquivo, incluindo a localização física dos dados. Quando um disco rígido é inicializado para aceitar o armazenamento de dados, em geral durante o processo de instalação do sistema, um número fixo de inodes por partição é criado. Este número será a quantidade máxima de arquivos, de todos os tipos (incluindo diretórios, arquivos especiais, links e etc.) que podem existir ao mesmo tempo naquela partição. Via de regra existe 1 inode para cada 2 a 8 kilobytes de espaço.

Quando um novo arquivo é criado, ele recebe livremente um inode. Neste inode se encontram as seguintes informações:

- Proprietário e grupo proprietário do arquivo.
- Tipo de arquivo (normal, diretório, ...)
- Permissões do arquivo [the section called “Direitos de acesso: A primeira linha de defesa do Linux”](#)
- Data e hora da criação, do último acesso e mudança
- Data e hora em que estas informações foram alteradas no inode.
- Número de links para este arquivo (veja mais adiante neste capítulo).
- Tamanho do arquivo
- Um endereço para o local onde os dados do arquivo estão de fato armazenados.

A única informação que não está em um inode é o nome do arquivo e o diretório. Estas informações são guardadas em arquivos diretórios especiais. Comparando os nomes dos arquivos e os números de inode, o sistema pode construir uma estrutura de árvore que os usuários possam entender. É possível exibir os números de inode usando a opção `-i` ao comando `ls`. Os inodes possuem seu próprio espaço em disco.

## Orientação no sistema de arquivos

### O caminho (path)

Quando você quer que o sistema execute um comando, quase nunca é necessário informar o caminho completo para ele. Por exemplo, nós sabemos que o comando `ls` está no diretório `/bin` (verifique com o `which -als`), porém não é necessário informar o comando `/bin/ls` para que o computador exiba o conteúdo do diretório atual.

A variável de ambiente `PATH` toma conta disto. Esta variável lista os diretórios no sistema onde arquivos executáveis podem ser encontrados, poupando ao usuário o tempo de digitar e memorizar a localização dos comandos. Assim, o caminho contém diversos diretórios que possuem `bin` em alguma parte de seus nomes, como o usuário abaixo demonstra. O comando `echo` é usado para exibir o conteúdo (“\$”) da variável `PATH`:

```
rogier:> echo $PATH
/opt/local/bin:/usr/X11R6/bin:/usr/bin:/usr/sbin:/bin
```

Neste exemplo, os diretórios `/opt/local/bin`, `/usr/X11R6/bin`, `/usr/bin`, `/usr/sbin` e `/bin` são subsequencialmente pesquisados em busca do programa desejado. Assim que uma ocorrência é encontrada a busca é interrompida, ainda que nem todos os diretórios no caminho tenham sido pesquisados. Isto pode levar a alguns percalços. No primeiro exemplo abaixo, a usuária sabe que existe um programa chamado `sendsms` usado para o envio de mensagens SMS e que outro usuário no mesmo sistema consegue usá-lo, porém ela não consegue. A diferença está na configuração da variável `PATH`:

```
[jenny@blob jenny]$ sendsms
bash: sendsms: comando não encontrado
[jenny@blob jenny]$ echo $PATH
/bin:/usr/bin:/usr/bin/X11:/usr/X11R6/bin:/home/jenny/bin
[jenny@blob jenny]$ su - tony
Senha:
tony:~>which sendsms
sendsms é /usr/local/bin/sendsms

tony:~>echo $PATH
/home/tony/bin.Linux:/home/tony/bin:/usr/local/bin:/usr/local/sbin:\
/usr/X11R6/bin:/usr/bin:/usr/sbin:/bin:/sbin
```

Perceba o uso do `su` (switch user - alterar usuário), que permite a execução de um shell no ambiente de outro usuário, desde que a senha deste usuário seja conhecida.

Uma barra invertida indica a continuação de uma linha na próxima, sem um **Enter** separando ambas.

No próximo exemplo, um usuário quer chamar o `wc` (word count - contador de palavras) para verificar o número de linhas em um arquivo, mas nada acontece e ele tem que desistir da ação usando a combinação **Ctrl+C**.

```
jumper:~> wc -l test
```

```
(Ctrl-C)
```

```
jumper:~> which wc
```

```
wc is hashed (/home/jumper/bin/wc)
```

```
jumper:~> echo $PATH
```

```
/home/jumper/bin:/usr/local/bin:/usr/local/sbin:/usr/X11R6/bin:\
```

```
/usr/bin:/usr/sbin:/bin:/sbin
```

O uso do comando **which** demonstra que este usuário tem um diretório `bin` em sua pasta `home`, a qual, por sua vez, também tem um programa chamado **wc**. Uma vez que a sua pasta `home` precede o restante do `path`, quando da chamada pelo **wc**, este programa “caseiro” é executado, com uma entrada que ele provavelmente não entende, de forma que é preciso interrompê-lo. Há diversas formas de resolver este problema (sempre há diversas formas de resolver um problema no UNIX/Linux): uma saída seria renomear o **wc** da `home` do usuário, outra seria fornecer o caminho completo para o comando que ele deseja executar, usando o **which**, com a opção `-a` para saber os caminhos.

Se o usuário usa os programas em outros diretórios com mais frequência, é possível mudar o seu caminho para que ele busque em seus próprios diretórios por último :

```
jumper:~> export PATH=/usr/local/bin:/usr/local/sbin:/usr/X11R6/bin:\
```

```
/usr/bin:/usr/sbin:/bin:/sbin:/home/jumper/bin
```

## Mudanças não são definitivas!

Perceba que quando se usa o comando **export** em um shell, as mudanças realizadas são temporárias, válidas apenas durante esta sessão (até que você realize o `logout`). Ao abrir novas sessões, mesmo enquanto a atual ainda está em execução, não importará o novo caminho para esta nova sessão. Nós veremos na [the section called “Seu ambiente texto”](#) como nós podemos tornar permanentes estas mudanças ao ambiente, adicionando estas linhas aos arquivos de configuração do shell.

## Caminhos absolutos e relativos

Um caminho, que é a trilha que você deve seguir pela estrutura de árvore para alcançar um determinado arquivo, pode ser descrito como se começasse pelo tronco da árvore (o `/` ou diretório raiz). Nesta hipótese, o caminho se inicia com uma barra e denomina-se caminho absoluto, vez que não há possibilidade de erro: apenas um arquivo no sistema pode se adequar àquela descrição.

No outro caso, o caminho não se inicia com uma barra e há possibilidade de confusão entre `~/bin/wc` (na pasta `home` do usuário) e `bin/wc` no `/usr`, do exemplo anterior. Caminhos que não se iniciam com uma barra são sempre relativos .

Nos caminhos relativos nós também usamos as indicações `.` e `..` para os diretórios atual e pai. Um par de exemplos práticos:

- Quando você quer compilar o código fonte, a documentação de instalação muitas vezes o instrui a executar o comando **./configure**, que executa o programa **configure** localizado no diretório atual (que vem com novo código), ao invés de executar outro programa `configure` localizado em outra parte do sistema.
- Em arquivos HTML, caminhos relativos são muitas vezes usados para tornar um conjunto de páginas facilmente alocáveis em outro lugar:

```

```

- Perceba a diferença mais uma vez:

```
theo:~> ls /mp3
```

```
ls: /mp3: Arquivo ou diretório inexistente
```

```
theo:~>ls mp3/
```

```
oriental/  pop/  sixties/
```

## Os arquivos e diretórios mais importantes

### O kernel

O kernel é o coração de um sistema. Ele gerencia as comunicações entre o hardware e os periféricos. O kernel também garante que os processos e daemons (processos de serviços) são iniciados e finalizados exatamente quando necessário for. O kernel realiza uma série de outras tarefas importantes, tantas que há uma lista de emails especial para os desenvolvedores do kernel apenas para este assunto, na qual uma imensa quantidade de informações é compartilhada. Por enquanto, é basta saber que o kernel é o arquivo mais importante no sistema.

### O shell

#### O que é um shell?

Quando eu estava procurando por uma explicação apropriada do conceito de *shell*, encontrei mais

problemas do que o esperado. Todos os tipos de definições estão disponíveis, desde simples ilustrações como “o shell é o volante do carro”, até a vaga definição do manual do Bash que diz que “o bash é um intérprete da linha de comando compatível com o sh” ou a ainda mais obscura expressão, “o shell gerencia a interação entre o usuário e o sistema”. Um shell é muito mais do que isto.

A melhor comparação que se pode fazer é dizer que o shell é um meio de falar com o computador, um idioma. A maior parte dos usuários conhece um outro idioma, aquele de selecionar-clicar do desktop. Entretanto, naquele idioma o computador é quem direciona a conversa, enquanto o usuário tem o papel passivo de escolher as tarefas dentre as que lhe são apresentadas. É muito difícil para um programador incluir todas as opções e possíveis usos da linha de comando em um formato gráfico. Por conta disto, as interfaces gráficas são, quase sempre, menos capazes do que o comando ou os comandos que realizam as tarefas por detrás da casca.

O shell, por outro lado, é um meio avançado de comunicação com o sistema, porquanto ele proporciona a possibilidade de uma comunicação verdadeiramente bilateral, em que ambas as partes tomam iniciativas. Ambos os interlocutores são iguais, de modo que é possível testar novas idéias. O shell permite que o usuário manipule o sistema de uma forma bem flexível. Outro fator é a possibilidade de automação de tarefas pelo shell.

## Tipos de shell

Assim como as pessoas conhecem diferentes idiomas ou dialetos, o computador conhece diferentes tipos de shells:

- **sh** ou Bourne Shell: o shell original, ainda usado em sistemas UNIX e em ambientes nele baseados. Este é o shell básico, um pequeno programa com poucos recursos. Quando acionado em modo compatível com o POSIX, o **bash** irá emular este shell.
- **bash** ou Bourne Again SHell: é o shell padrão do GNU, intuitivo e flexível. Provavelmente o mais indicado a usuários iniciantes, embora também seja simultaneamente uma ferramenta poderosa para os usuários avançados e profissionais. No Linux, o **bash** é o shell padrão para usuários comuns. Este shell é chamado de *sucessor* do shell Bourne, com o incremento de muitos add-ons e plugins. Isto implica que o Bourne Again SHell é compatível com o shell Bourne: os comandos que funcionam no **sh** também irão funcionar no **bash**. Entretanto, a recíproca nem sempre é verdadeira. Todos os exemplos e exercícios neste livro usam o **bash**.
- **cs** ou C Shell: a sintaxe deste shell lembra aquela da linguagem de programação C. Às vezes requisitada por programadores.
- **tcsh** ou Turbo C Shell: um aprimoramento do C Shell, focado na agilidade e em uma interface amigável com o usuário.
- **ksh** ou o shell Korn: às vezes cumpre o gosto das pessoas com experiência em UNIX. É um sucessor do shell Bourne, com a configuração padrão, um pesadelo para iniciantes.

O arquivo `/etc/shells` fornece um panorama dos shells existentes em um sistema Linux:

```
mia:~> cat /etc/shells
/bin/bash
/bin/sh
/bin/tcsh
/bin/csh
```

## Shell bourne falso

Perceba que o `/bin/sh` é normalmente um link para o Bash, que será executado no modo compatível com o Bourne quando invocado desta forma.

O seu shell padrão está configurado no arquivo `/etc/passwd`, como no caso desta linha para a usuária *mia*:

```
mia:L2N0fqdlPrHwE:504:504:Mia Maya:/home/mia:/bin/bash
```

Para alternar entre um shell e outro, simplesmente informe o nome do novo shell no terminal ativo. O sistema irá encontrar o diretório onde o nome aparece usando a configuração do `PATH` e, considerando que o shell é um arquivo executável (programa), o shell atual o aciona e ele é executado. Normalmente um novo prompt é exibido, porquanto cada shell possui sua aparência própria:

```
mia:~> tcsh
[mia@post21 ~]$
```

## Qual shell estou usando?

Se você não sabe qual shell está usando, para descobri-lo você pode verificar a linha da sua conta no `/etc/passwd` ou informar o comando

```
echo $SHELL
```

## Sua pasta home

A sua pasta home é o seu destino padrão quando se conecta ao sistema. Na maior parte dos casos ela é

um subdiretório da `/home`, embora isto possa variar. Sua pasta home pode estar alocada em um disco rígido de um servidor remoto; neste caso sua pasta home poderá ser encontrada no `/nethome/seu_nome_de_usuario`. Pode ser também que o seu administrador tenha optado por uma configuração menos intuitiva e a sua home poderá estar no `/disk6/HU/07/jgillard`.

Qualquer que seja o caminho para a sua home, não é preciso se preocupar muito com isto. O caminho correto para a sua pasta home está armazenado na variável de ambiente `HOME`, caso algum programa precise dela. Com o comando **echo** você pode exibir o conteúdo desta variável:

```
orlando:~> echo $HOME
/nethome/orlando
```

Você pode fazer tudo o que quiser na sua home. Você pode colocar tantos arquivos em quantos diretórios desejar, embora o tamanho máximo de dados e arquivos esteja limitado pela capacidade do disco e da partição, sendo ainda possível que o administrador tenha estabelecido um regime de quotas. Impor restrições ao uso do disco era uma prática comum quando o espaço em disco era muito caro. Atualmente, os limites são aplicados quase que exclusivamente em ambientes grandes. Você pode verificar se foi estabelecido um limite usando o comando **quota**.

```
pierre@lamaison:~/> quota -v
Diskquotas para usuário pierre (uid 501): nenhuma
```

Caso tenham sido estabelecidas quotas, você pode obter uma lista das partições limitadas e suas especificações. Durante um período de graça pode ser que se tolere a extrapolação dos limites com poucas ou nenhuma restrição. Informações detalhadas podem ser encontradas usando os comandos **info quota** ou **man quota**.

## Nenhuma quota?

Nenhuma limitação ao uso do sistema de arquivos foi estabelecida se não é possível encontrar o comando **quota**.

O til (~) simboliza o seu diretório home, uma abreviação de `/path_to_home/user_name`. Este mesmo caminho está armazenado na variável `HOME`, de forma que não é necessário fazer nada para ativá-la. Uma demonstração simples: para mudar do `/var/music/albums/arno/2001` para o `imagens` na sua pasta home com um comando elegante:

```
rom:/var/music/albums/arno/2001> cd ~/images
```

```
rom:~/images> pwd
/home/rom/images
```

Mais adiante neste capítulo nós discutiremos os comandos para gerenciar arquivos e diretórios a fim de manter a sua pasta home organizada.

## Os arquivos de configuração mais importantes

Como já mencionado, a maior parte dos arquivos de configuração estão armazenados no diretório `/etc`. O conteúdo pode ser acessado com o comando **cat**, que envia arquivos texto para a saída padrão (normalmente o seu monitor). A sintaxe é intuitiva:

```
cat file1 file2 ... fileN
```

Nesta seção nós tentaremos apresentar um panorama dos arquivos de configuração mais comuns. Esta certamente não será uma listagem completa. Além disso, quando novos pacotes são adicionados, arquivos de configuração extra também podem ser incluídos no `/etc`. Quando estiver lendo arquivos de configuração, você descobrirá que eles são, via de regra, bem comentados e auto-explicativos. Alguns arquivos também possuem páginas `man` com documentação adicional, como no caso do **man group**.

**Table 3.3. Os arquivos de configuração mais comuns**

Arquivo	Informação/serviço
aliases	Os alias de email para uso dos servidores do sendmail e do postfix. Manter um servidor de email em cada um dos sistemas é uma prática comum no mundo UNIX e quase todas as distribuições do Linux ainda vêm com um pacote Sendmail. Neste arquivo os nomes dos usuários locais são agrupados com os seus nomes reais, como aparecem nos endereços de email ou outros endereços locais.
apache	Arquivos de configuração para o servidor web Apache
bashrc	O arquivo de configuração do Bourne Again Shell, válido para todo o sistema. Define funções e alias para todos os usuários. Outros shells podem ter seus próprios arquivos de configuração, como o <code>cshrc</code> .
Os diretórios <code>crontab</code> e o <code>cron.*</code>	A configuração de tarefas que precisam ser executadas periodicamente - backups, atualizações das bases de dados do sistema, limpeza, rotação dos logs e etc.
default	Padrão para opções de certos comandos, como o <b>useradd</b> .
filesystems	Sistemas de arquivos conhecidos: <code>ext3</code> , <code>vfat</code> , <code>iso9660</code> etc.
fstab	Lista partições e seus <i>pontos de montagem</i> .

ftp*	Configuração do servidor-ftp: quem pode conectar, quais partes do sistema são acessíveis e etc.
group	Arquivo de configuração para grupos de usuários. Use os utilitários <b>groupadd</b> , <b>groupmod</b> e <b>groupdel</b> para editar esse arquivo. Edite manualmente apenas se você de fato souber o que está fazendo.
hosts	Uma lista de máquinas que pode ser contatada usando a rede, mas sem a necessidade de um serviço de nomes de domínio. Isto não tem nada em comum com a configuração de rede do sistema, que é feita no /etc/sysconfig.
inittab	Informações para inicialização: modo, número de consoles texto e etc.
issue	Informações sobre a distribuição (versão e/ou informações sobre o kernel).
ld.so.conf	Localização dos arquivos de biblioteca
lilo.conf, silo.conf, about.conf etc.	Informação de inicialização para o Linux LOader, o sistema para inicialização do sistema que gradualmente vem sendo substituído pelo GRUB.
logrotate.*	Rotação dos logs, um sistema que previne que se acumulem grandes quantidades de arquivos de logs.
mail	Diretório que contém instruções para o comportamento do servidor de emails.
modules.conf	Configuração dos módulos que acionam recursos especiais (drivers).
motd	Mensagem do dia: Exibida a todos que conectam ao sistema (no modo texto), pode ser usada pelo administrador do sistema para anunciar recursos ou manutenção do sistema e etc.
mtab	Sistemas de arquivo atualmente montados. É aconselhável jamais editar este arquivo.
nsswitch.conf	Ordem para contatar os resolvers de nomes quando um processo precisa resolver o nome de uma máquina.
pam.d	Configuração dos módulos de autenticação
passwd	Lista os usuários locais. Use os utilitários <b>useradd</b> , <b>usermod</b> e <b>userdel</b> para editar este arquivo. Edite manualmente apenas se tiver muita certeza do que está fazendo.
printcap	Arquivo de configuração de impressora. Desatualizado, mas ainda muito usado. Não o edite a não ser que saiba o que está fazendo.
profile	Configurações, válidas para todo o sistema, do ambiente shell: variáveis, propriedades padrão de novos arquivos, limitação de recursos e etc.
rc*	Diretórios que definem os serviços ativos em cada um dos níveis de execução.
resolv.conf	Ordem para contatar os servidores de DNS (Servidores de Nomes de Domínio apenas).
sendmail.cf	Principal arquivo de configuração para o servidor Sendmail.
services	Conexões aceitas por esta máquina (portas abertas).
sndconfig OU sound	Configuração do cartão e eventos de som.
ssh	Diretório contendo os arquivos de configuração para shells seguros e servidores.
sysconfig	Diretório contendo os arquivos de configuração do sistema: mouse, teclado, rede, desktop, relógio do sistema, gerenciamento de energia e etc. (específico do RedHat).
X11	Configurações para o servidor gráfico, X. O RedHat usa o XFree, que aparece no nome deste arquivo de configuração, XFree6Config. Também contém diretrizes gerais para os gerentes de janelas disponíveis no sistema, por exemplo <b>gdm</b> , <b>fvwm</b> , <b>twm</b> , etc.
xinetd.* OU inetd.conf	Arquivos de configuração para serviços de Internet que são executados a partir dos daemons de serviços (estendidos) de Internet do sistema (servidores que não são executados com um daemon independente).

Ao longo deste guia, aprenderemos mais sobre estes arquivos e estudaremos alguns deles em detalhes.

## Os dispositivos mais comuns

Dispositivos, genericamente, todos os periféricos acoplados ao PC, que não o próprio CPU, são apresentados ao sistema com uma entrada no diretório /dev. Uma das vantagens desta forma UNIX de manipular os dispositivos é que nem o sistema, nem o usuário precisam se preocupar muito com as especificações destes dispositivos.

Usuários que são novos no Linux ou UNIX em geral são sufocados pela imensa quantidade de novos nomes e conceitos que precisam aprender. Por esta razão, uma lista de dispositivos comuns foi incluída nesta introdução.

**Table 3.4. Dispositivos comuns**

Nome	Dispositivo
cdrom	Drive de CD
console	Entrada especial para o console usado atualmente.
cua*	Portas seriais
dsp*	Dispositivos para amostragem e gravação

fd*	Entradas para grande parte dos drives de disquete, o padrão é /dev/fd0 um drive de entrada para disquetes de 1.44MB.
hd[a-t][1-16]	Suporte padrão para drives IDE com o número máximo de partições para cada um.
ir*	Dispositivos infravermelho
isdn*	Gerenciamento de conexões ISDN
js*	Joystick(s)
lp*	Impressoras
mem	Memória
midi*	tocador midi
mixer* e music	Modelo ideal de um mixer (combina ou adiciona sinais)
modem	Modem
mouse (também msmouse, logimouse, psmouse, input/mice, psaux)	Todos os tipos de mouses
null	Lata de lixo sem fundo
par*	Entradas para suporte de portas paralelas
pty*	Pseudo-terminais
radio*	Para radio-amadores (HAMs).
ram*	dispositivo de inicialização
sd*	discos SCSI com suas partições
sequencer	Para aplicativos de audio usando recursos de sintetizados do cartão de som controle do dispositivo-MIDI)
tty*	Consoles virtuais para simular terminais vt100.
usb*	Cartão USB e scanner
video*	Para uso com um cartão de gráficos que suporte vídeos.

## Os arquivos de variáveis mais comuns

No diretório `/var` encontramos um conjunto de diretórios para armazenar dados específicos não constantes (o oposto do programa `ls` ou dos arquivos de configuração do sistema, que se alteram com pouca frequência ou até mesmo nunca). Todos os arquivos que mudam frequentemente, como os arquivos de log, caixas de emails, arquivos cadeado, spooler e etc são mantidos em um subdiretório do `/var`.

Como medida de segurança, estes arquivos são normalmente mantidos em partes separadas do sistema de arquivos principal, de forma a facilitar o controle sobre eles impondo-lhes permissões mais estritas quando necessário. Muitos deles também precisam de mais permissões do que o normal, como o `/var/tmp`, quer precisa garantir acesso de gravação para todos os usuários. Muita atividade dos usuários é realizada por aqui, que pode até mesmo ser proveniente de usuários anônimos da internet conectados ao sistema. Este é o principal motivo pelo qual o diretório `/var`, bem como todos os seus subdiretórios, são comumente alocados em uma partição separada. Desta forma, não há risco de que, por exemplo, um email bomba encha completamente o sistema de arquivos alocado para arquivos mais importantes como seus programas e arquivos de configuração.

### **`/var/tmp` e `/tmp`**

Os arquivos no `/tmp` podem ser deletados sem aviso, por rotinas comuns do sistema ou quando de sua reinicialização. Em alguns sistemas (modificados), também o `/var/tmp` pode se comportar de forma imprevisível. De qualquer forma, uma vez que este não é o caso padrão, nós recomendamos que o `/var/tmp` seja usado para a gravação de arquivos temporários. Se estiver em dúvida, verifique com o administrador do seu sistema. Se você é que gerencia seu sistema, você pode estar seguro de que este é um local seguro, se você não mudou conscientemente as configurações do `/var/tmp` (como root, já que um usuário convencional não pode fazer isto).

O que quer que você faça, tente fazê-lo com os privilégios concedidos a um usuário normal - não se meta a salvar arquivos na raiz (`/`) do sistema de arquivos, não os salve no `/usr` ou algum subdiretório ou em um local reservado. Isto irá limitar o seu acesso a locais seguros do sistema de arquivos.

Um dos principais sistemas de segurança em um sistema UNIX, que certamente também é implementada em todas as máquinas Linux, é a manutenção de logs, que catalogam todas as ações dos usuários, processos, eventos do sistema e etc. O arquivo de configuração do chamado `syslogd` determina quais e por quanto tempo as informações serão mantidas. A localização padrão de todos os logs é o `/var/log`, que contém diferentes arquivos para acessos, servidores, mensagens do sistema e etc.

No `/var` normalmente encontramos dados de servidores, que são mantidos aqui para que fiquem separados de dados críticos como o programa de servidor e seus arquivos de configuração. Um exemplo típico em sistemas Linux é o `/var/www`, que contém as páginas HTML, os scripts e as imagens que um servidor web oferece. A árvore FTP em um servidor FTP (dados que podem ser baixados por clientes

remotos) também são mantidos em um dos subdiretórios do `/var`. Uma vez que estas informações são acessíveis ao público e até mesmo podem ser modificadas por usuários anônimos, é mais seguro mantê-las aqui, distante de partições ou diretórios com dados sensíveis.

Na maior parte das instalações em estações de trabalho, o `/var/spool` conterá ao menos um diretório at e um `cron`, contendo tarefas agendadas. Em ambientes de escritório, este diretório normalmente contém também um `lpd`, que armazena a fila de impressão, arquivos de configuração da impressora e seus arquivos de log.

Em servidores geralmente encontramos `/var/spool/mail`, contendo entrada de emails para os usuários locais, armazenados em um arquivo por usuário, a “caixa de entrada” do usuário. Um diretório relacionado é o `mqueue`, a área de spooler para as mensagens de email não enviadas. Estas partes do sistema podem ficar muito ocupadas em servidores de email com muitos usuários. Servidores de notícias também usam a área do `/var/spool` dada a grande quantidade de mensagens que eles precisam processar.

O diretório `/var/lib/rpm` é específico de distribuições baseadas no RPM (RedHat Package Manager); é onde são armazenadas as informações sobre os pacotes RPM. Outros gerentes de pacotes também armazenam suas informações em algum lugar do `/var`.

## Manipulando arquivos

### Exibindo as propriedades de um arquivo

#### Mais sobre o `ls`

Além do nome de um arquivo, o `ls` pode fornecer muitas outras informações, como o tipo de arquivo, como já dissemos anteriormente. Ele pode também exibir as permissões de um arquivo, o seu tamanho, número de inode, data e horário em que foi criado, proprietários e quantidade de links para o arquivo. Com a opção `-a` ao `ls`, arquivos que normalmente estão escondidos podem ser visualizados. Estes são os arquivos cujo nome se inicia com um ponto. Exemplos são os arquivos de configuração em sua pasta home. Quando você já trabalhou com um sistema durante algum tempo, você irá perceber que dezenas de arquivos e diretórios serão criados sem que sejam automaticamente listados no índice do diretório. Além disso, todos os diretórios contém um arquivo chamado simplesmente ponto (`.`) e outro dois pontos (`..`), que são usados combinados com seus números de inode para determinar a posição do diretório na árvore do sistema de arquivos.

Você de fato deveria ler as páginas Info do `ls`, já que ele é um comando muito comum com muitas opções bastante úteis. Opções podem ser combinadas, como é comum entre comandos UNIX e suas opções. Uma combinação muito comum é `ls -al`; que exibe uma lista longa dos arquivos e suas propriedades, bem como os destinos para os quais os links simbólicos apontam. `ls -ltr` exibe os mesmos arquivos, só que na ordem inversa das últimas alterações, de forma que o arquivo com as alterações mais recentes fique no topo da lista. Aqui vão alguns exemplos:

```
krissie:~/mp3> ls
Albums/ Radio/ Singles/ gene/ index.html

krissie:~/mp3> ls -a
./ .thumbs Radio gene/
../ Albums/ Singles/ index.html

krissie:~/mp3> ls -l Radio/
total 8
drwxr-xr-x  2 krissie krissie 4096 Oct 30 1999 Carolina/
drwxr-xr-x  2 krissie krissie 4096 Sep 24 1999 Slashdot/

krissie:~/mp3> ls -ld Radio/
drwxr-xr-x  4 krissie krissie 4096 Oct 30 1999 Radio/

krissie:~/mp3> ls -ltr
total 20
drwxr-xr-x  4 krissie krissie 4096 Oct 30 1999 Radio/
-rw-r--r--  1 krissie krissie  453 Jan  7 2001 index.html
drwxrwxr-x 30 krissie krissie 4096 Oct 20 17:32 Singles/
drwxr-xr-x  2 krissie krissie 4096 Dec  4 23:22 gene/
drwxrwxr-x 13 krissie krissie 4096 Dec 21 11:40 Albums/
```

Na maior parte das versões Linux o `ls` é aliado ao `ls-em-cores` por padrão. Este recurso permite saber o tipo de arquivo sem o uso de quaisquer opções ao `ls`. Para consegui-lo, cada tipo de arquivo recebe uma cor. O esquema padrão está no `/etc/DIR_COLORS`:

**Table 3.5. Esquema padrão do `ls-em-cores`**

Cor	Tipo de Arquivo
azul	diretórios
red	arquivos compactos
branco	arquivos texto
rosa	imagens



ciano	links
amarelo	dispositivos
verde	executáveis
destaque em vermelho	links quebrados

Mais informações podem ser encontradas na página man. Em outros tempos, as mesmas informações eram exibidas usando sufixos a cada tipo de nome de arquivo não padronizado. Quando do uso de terminal monocromático (como na impressão de uma listagem de um diretório) e para facilitar a leitura em geral, este esquema ainda é usado:

**Table 3.6. Esquema de sufixos padrão para o ls**

Caracter	Tipo de Arquivo
nada	arquivo normal
/	diretório
*	executável
@	link
=	socket
	named pipe

Uma descrição completa dos recursos e funções do comando **ls** pode ser encontrada em **info coreutils ls**.

### Mais ferramentas

Para saber mais sobre os tipos de dados que estamos lidando, usamos o comando **file**. Aplicando certos testes que verificam as propriedades de um arquivo no sistema de arquivos, números mágicos e alguns testes de linguagem, o **file** tenta realizar uma previsão fundamentada sobre o formato de um arquivo. Alguns exemplos:

```
mike:~> file Documents/
Documents/: diretório
```

```
mike:~> file high-tech-stats.pdf
high-tech-stats.pdf: documento PDF, versão 1.2
```

```
mike:~> file Nari-288.rm
Nari-288.rm: arquivo RealMedia
```

```
mike:~> file bijlage10.sdw
bijlage10.sdw: documento Microsoft Office
```

```
mike:~> file logo.xcf
logo.xcf: dados de imagem GIMP XCF, versão 0, 150 x 38, RGB Color
```

```
mike:~> file cv.txt
cv.txt: ISO-8859 texto
```

```
mike:~> file image.png
image.png: dados de imagem PNG , 616 x 862, 8-bit escala de cinza, não interlaçado
```

```
mike:~> file figure
figure: texto ASCII
```

```
mike:~> file me+tux.jpg
me+tux.jpg: dados de imagem JPEG, padrão JFIF 1.01, resolução (DPI),
"28 Jun 1999", 144 x 144
```

```
mike:~> file 42.zip.gz
42.zip.gz: dados comprimidos gzip, deflated, nome de arquivo original,
`42.zip', última modificação: Qui Nov 1 23:45:39 2001, os: Unix
```

```
mike:~> file vi.gif
vi.gif: dados de imagem GIF, versão 89a, 88 x 31
```

```
mike:~> file slide1
slide1: documento texto HTML
```

```
mike:~> file template.xls
template.xls: documento Microsoft Office
```

```
mike:~> file abook.ps
abook.ps: documento texto PostScript em nível 2.0
```

```
mike:~> file /dev/log
/dev/log: socket
```

```
mike:~> file /dev/hda
/dev/hda: bloco especial (3/0)
```

O comando **file** possui diversas opções, dentre elas a opção **-z** para investigar arquivos compactos. Veja a **info file** para uma descrição detalhada. Tenha em mente que os resultados do **file** não são perfeitos, mas apenas uma tentativa de adivinhar o que um arquivo é. Em outras palavras o **file** pode ser enganado.

## Por que todo esse escarcéu sobre tipos e formatos de arquivos?

Em poucas palavras, discutiremos algumas ferramentas da linha de comando para olhar *arquivos texto*. Estas ferramentas não irão funcionar quando usadas com o tipo errado de arquivo. No pior cenário, isso levará o seu terminal a um travamento ou a emitir o som de vários bipes. Se isto acontecer com você, apenas encerre a sessão do terminal e inicie uma nova. Tente, contudo, evitar que isto ocorra, vez que é muito irritante para outras pessoas.

## Criando e deletando arquivos e diretórios

### Fazer uma bagunça...

... Não é uma coisa difícil. Hoje em dia, quase todos os sistemas estão em rede, de forma que arquivos são rotineiramente copiados de uma máquina para outra. Especialmente quando se trabalha em um ambiente gráfico, a criação de novos arquivos é uma tarefa trivial, feita sem a aprovação do usuário. Para ilustrar o problema, segue o conteúdo completo do diretório de um novo usuário, criado em um sistema RedHat padrão:

```
[novousuário@blob usuário]$ ls -al
total 32
drwx----- 3 user      user      4096 Jan 16 13:32 .
drwxr-xr-x  6 root      root      4096 Jan 16 13:32 ..
-rw-r--r--  1 user      user       24 Jan 16 13:32 .bash_logout
-rw-r--r--  1 user      user      191 Jan 16 13:32 .bash_profile
-rw-r--r--  1 user      user      124 Jan 16 13:32 .bashrc
drwxr-xr-x  3 user      user      4096 Jan 16 13:32 .kde
-rw-r--r--  1 user      user     3511 Jan 16 13:32 .screenrc
-rw-----  1 user      user       61 Jan 16 13:32 .xauthDqztLr
```

À primeira vista, o conteúdo de uma pasta home “usada” não parece tão ruim

```
antigousuário:~> ls
app-defaults/ crossover/ Fvwm@      mp3/      OpenOffice.org638/
articles/     Desktop/   GNUstep/   Nautilus/ staroffice6.0/
bin/          Desktop1/ images/    nqc/      training/
bro/          desktoptest/ Machines@  ns_imap/  webstart/
C/            Documents/ mail/      nsmail/   xml/
closed/       Emacs@    Mail/     office52/ Xrootenv.0
```

Porém, quando todos os diretórios e arquivos iniciados com um ponto são incluídos, 185 itens aparecem neste diretório. Isto porque quase todos os aplicativos possuem seus próprios diretórios e/ou arquivos, contendo configurações específicas do usuário, na home deste usuário. Normalmente, estes arquivos são criados quando da primeira execução do aplicativo. Em algumas casos você será avisado se for preciso criar um diretório inexistente, porém, na maior parte do tempo, tudo é feito automaticamente.

Mais ainda, novos arquivos são criados quase que continuamente, vez que usuários querem salvar arquivos, manter diferentes versões de seus trabalhos, usar aplicativos de Internet e baixar arquivos e anexos para sua máquina local. Isto nunca pára. Parece evidente que é necessário ter um esquema para manter as coisas em ordem.

Na próxima seção, discutiremos a nossa forma de manter tudo organizado. Discutiremos somente as ferramentas texto disponíveis no shell, já que as ferramentas gráficas são muito intuitivas e possuem a mesma aparência do conhecido gerente de arquivos apontar-e-clicar do MS Windows, incluindo funções de ajuda gráficas e outros recursos que se espera neste tipo de aplicativo. A lista a seguir é um panorama dos gerentes de arquivos mais populares para o GNU/Linux. A maior parte dos gerentes de arquivos pode ser iniciado do menu de seu gerente de desktop, ou simplesmente clicando no ícone do seu diretório home, ou ainda a partir de linha de comando, informando estes comandos:

- **nautilus**: Gerente de arquivos padrão do Gnome, o desktop do GNU. Uma excelente documentação sobre esta ferramenta pode ser encontrada no <http://www.gnome.org>.
- **konqueror**: O gerente de arquivos usado no desktop KDE. O manual pode ser encontrado no <http://docs.kde.org>.
- **mc**: Midnight Commander, o gerente de arquivos do UNIX, seguindo a moda do Norton Commander. Toda a documentação pode ser encontrada no <http://gnu.org/directory/> ou em um espelho, como o <http://www.ibiblio.org>.

Estes aplicativos certamente merecem uma chance e normalmente impressionam os novatos no Linux, também porque existe uma extensa gama deles: estes são apenas os mais populares gerentes de diretórios e arquivos, muitos outros projetos estão em desenvolvimento. Passemos agora a analisar o interior do sistema e descobrir como estas ferramentas gráficas fazem uso dos comandos UNIX.

### As ferramentas

## Criando diretórios

Um modo de manter tudo em seu lugar é dar locais específicos a certos arquivos criando diretórios e subdiretórios (ou pastas e subpastas, se você preferir). Isto é feito com o comando **mkdir**:

```
richard:~> mkdir archive
```

```
richard:~> ls -ld archive
drwxrwxrwx  2 richard richard          4096 Jan 13 14:09 archive/
```

A criação de diretórios e subdiretórios em um único passo é realizada com a opção **-p**:

```
richard:~> cd archive
```

```
richard:~/archive> mkdir 1999 2000 2001
```

```
richard:~/archive> ls
1999/ 2000/ 2001/
```

```
richard:~/archive> mkdir 2001/reports/Restaurants-Michelin/
mkdir: não é possível criar o diretório `2001/reports/Restaurants-Michelin/':
Arquivo ou diretório inexistente
```

```
richard:~/archive> mkdir -p 2001/reports/Restaurants-Michelin/
```

```
richard:~/archive> ls 2001/reports/
Restaurants-Michelin/
```

Se o arquivo precisa de outras permissões que não a padrão na criação de arquivos, as novas permissões de acesso podem ser configuradas em um único passo, usando o comando **mkdir**, veja as páginas Info para saber mais. Discutiremos os modos de acesso na próxima seção sobre a segurança dos arquivos.

O nome de um diretório tem que se adequar às mesmas regras válidas para os nomes de arquivos regulares. Uma das restrições mais importantes é que não se pode ter dois arquivos com o mesmo nome dentro de um diretório (tenha em mente, contudo, que o Linux, como o UNIX, é um sistema operacional sensível a maiusculização). Praticamente não existem limites à extensão de um nome de arquivo, porém, em geral, eles possuem menos de 80 caracteres, de modo que possam caber em uma linha de terminal. É possível usar quaisquer caracteres no nome de um arquivo, embora seja recomendado excluir caracteres que possuam um significado especial para o shell. Quando estiver em dúvida, verifique no [Appendix C, Recursos do Shell](#).

## Movendo arquivos

Agora que nós estruturamos apropriadamente o seu diretório home, é hora de limpar os arquivos não classificados usando o comando **mv**:

```
richard:~/archive> mv ../report[1-4].doc reports/Restaurants-Michelin/
```

Este comando também pode ser usado para renomear arquivos:

```
richard:~> ls To_Do
-rw-rw-r--  1 richard richard    2534 Jan 15 12:39 To_Do
```

```
richard:~> mv To_Do done
```

```
richard:~> ls -l done
-rw-rw-r--  1 richard richard    2534 Jan 15 12:39 done
```

É claro que apenas o nome do arquivo muda. Todas as demais propriedades permanecem inalteradas.

Informações detalhadas sobre a sintaxe e os recursos do comando **mv** podem ser encontradas nas páginas man e Info. O uso desta documentação deve ser sempre o seu primeiro reflexo quando confrontado por um problema. A resposta ao seu problema provavelmente se encontra na documentação do sistema. Mesmo usuários experientes lêem páginas man todos os dias, de modo que usuários iniciantes deveriam lê-las o tempo todo. Após algum tempo, você passará a saber quais são as opções mais comuns dos comandos mais usuais, mas ainda precisará recorrer à documentação como primeira fonte de informação. Perceba que as informações contidas nos HOWTOs, FAQs, páginas man e outras fontes vêm lentamente sendo fundidas com as páginas Info, que são hoje a fonte mais atualizada de documentação online (bem como na documentação disponível no sistema).

## Copiando arquivos

A cópia de arquivos e diretórios é feita com o comando **cp**. Uma opção útil é a cópia recursiva (copia todos os arquivos e subdiretórios contidos em um diretório), feita por meio da opção **-R** ao comando **cp**. A sintaxe geral é:

```
cp [-R] fromfile tofile
```

Um exemplo é o caso do usuário *caranovo*, que quer as mesmas configurações de ambiente que o usuário *caravelho* usa. Uma forma de resolver este problema é copiando as configurações do *caravelho*

para a pasta home do *caranovo*:

```
victor:~> cp -R ../oldguy/.gnome/ .
```

Isto gera alguns erros relativos às permissões de arquivos, mas todos eles têm a ver com arquivos pessoais que, de qualquer forma, o *novocara* não precisará. A seguir discutiremos como mudar estas permissões, caso elas se tornem de fato um problema.

## Removendo arquivos

Use o comando **rm** para remover arquivos individuais, **rmdir** para remover diretórios vazios. (Use o **ls -a** para saber se um diretório está ou não vazio). O comando **rm** também possui opções para remover diretórios não-vazios com todos os seus subdiretórios, leia as páginas Info para saber sobre estas perigosas opções.

## O quão vazio um diretório pode estar?

É normal que os diretórios `.` (ponto) e `..` (ponto-ponto) não possam ser removidos, já que eles são também necessários em um diretório vazio para determinar a sua posição na hierarquia do sistema de arquivos.

No Linux, assim como no UNIX, não há lata de lixo - ao menos não para o shell, embora existam diversas soluções para o ambiente gráfico. Portanto, uma vez apagado, um arquivo é realmente perdido, sem chance de recuperá-lo, a não ser que você possua backups ou seja muito rápido e tenha um bom administrador de sistema. Para proteger o usuário iniciante deste perigo, o comportamento interativo dos comandos **rm**, **cp** e **mv** pode ser ativado usando a opção `-i`. Neste caso, o sistema não irá agir imediatamente após a ordem. Ao invés, ele irá perguntar por uma confirmação da ordem, de forma que é necessário apertar o **Enter** mais uma vez antes de realizar o dano:

```
mary:~> rm -ri archive/  
rm: descend into directory `archive'? y  
rm: descend into directory `archive/reports'? y  
rm: remove directory `archive/reports'? y  
rm: descend into directory `archive/backup'? y  
rm: remove `archive/backup/sysbup200112.tar'? y  
rm: remove directory `archive/backup'? y  
rm: remove directory `archive'? y
```

Discutiremos como tornar esta opção padrão [Chapter 7, Lar doce /lar \(/home\)](#), que discute a personalização do ambiente shell.

## Encontrando arquivos

### Usando os recursos do shell

No exemplo sobre movimentação de arquivos nós já vimos como o shell pode manipular múltiplos arquivos de uma só vez. Naquele exemplo, o shell descobre automaticamente o que o usuário quer dizer com os quesitos entre colchetes “[” e “]”. O shell pode substituir seqüências numéricas e também letras maiúsculas e minúsculas. Ele também substitui tantos caracteres quanto você quiser com um asterisco e apenas um caractere com um ponto de interrogação.

Todos os tipos de substituições podem ser usadas simultaneamente; quanto a isto o shell é muito lógico. O shell Bash, por exemplo, não tem problema algum com expressões como **ls *nomedodir*/\*/\*/\*[2-3]**.

Em outros shells, o asterisco é normalmente usado para minimizar o esforço de digitação: usa-se **cd *dir*\*** ao invés de **cd *diretorio***. No Bash, todavia, isto não é necessário porque o shell GNU possui um recurso chamado complemento de nome. Isto significa que você pode digitar os primeiros caracteres de um comando (em qualquer lugar) ou um arquivo (no diretório atual) e, se nenhuma confusão for possível, o shell irá descobrir o que você quer dizer. Por exemplo, em um diretório contendo muitos arquivos, você pode verificar se existem quaisquer arquivos cujos nomes se iniciam pela letra A apenas digitando **ls A** e pressionando duas vezes a tecla **Tab**, ao invés de **Enter**. Se houver apenas um arquivo com nome iniciado pela letra “A”,

## Which

Um modo muito simples de buscar arquivos é usar o comando **which**, para pesquisar pelo arquivo nos diretórios listados no PATH do usuário. É claro que, uma vez que o PATH somente contém os caminhos para os diretórios que contém programas executáveis, o **which** não funciona para procurar por arquivos comuns. O **which** é útil para resolver problemas de “Comando não Encontrado”. No exemplo abaixo, a usuária *tina* não consegue usar o programa **acroread**, ao passo em que seus colegas utilizam-no no mesmo sistema sem quaisquer problemas. O problema é semelhante ao do PATH na parte anterior: um colega da Tina diz a ela que consegue encontrar o programa no `/opt/acroread/bin`, porém este diretório não aparece no PATH dela:

```
tina:~> which acroread  
/usr/bin/which: no acroread in (/bin:/usr/bin:/usr/bin/X11)
```

O problema pode ser resolvido informando-se o caminho completo do comando a ser executado ou reenviando o conteúdo da variável PATH:

```
tina:~> export PATH=$PATH:/opt/acroread/bin
```

```
tina:~> echo $PATH  
/bin:/usr/bin:/usr/bin/X11:/opt/acroread/bin
```

Ao usar o comando **which** também é possível verificar se um comando é um alias para outro:

```
gerrit:~> which -a ls  
ls is aliased to `ls -F --color=auto`  
ls is /bin/ls
```

Se isto não funcionar no seu sistema, use o comando **alias**:

```
tille@www:~/mail$ alias ls  
alias ls='ls --color'
```

## Find e locate

Estas são as ferramentas verdadeiras para realizar pesquisas por diretórios que não aparecem no PATH. A ferramenta **find**, conhecida no UNIX, é muito poderosa, o que pode implicar em uma sintaxe um pouco mais complexa. O GNU **find**, todavia, lida com os problemas de sintaxe. Este comando não apenas permite que você procure por nomes de arquivos, mas aceita também seus tamanhos, data da última alteração e outras propriedades de arquivos como critérios para pesquisa. O uso mais comum é para encontrar arquivos por seus nomes:

```
find <path> -name <searchstring>
```

Isto pode ser interpretado como "Procure em todos os diretórios contidos no caminho fornecido e exiba os nomes dos arquivos que contém a seqüência informada em seus nomes" (não em seus conteúdos).

Outro uso do **find** é para pesquisar por arquivos de um dado tamanho, como no exemplo abaixo em que o usuário *peter* quer encontrar todos os arquivos no diretório atual ou em seus subdiretórios que sejam maiores que 5Mb:

```
peter:~> find . -size +5000k  
psychotic_chaos.mp3
```

Se você fuçar nas páginas man, você verá que o **find** também consegue realizar operações com os arquivos encontrados. O melhor é testar primeiro sem a opção **-exec**, para saber se os arquivos corretos estão selecionados, então o comando pode ser reexecutado para deletar os arquivos indicados. Abaixo nós procuramos por arquivos terminados em **.tmp**:

```
peter:~> find . -name "*.tmp" -exec rm {} \;
```

```
peter:~>
```

## Otimize!

Este comando pode invocar o **rm** tantas vezes quantos forem os arquivos que cumpram os critérios da busca. No pior caso, isto pode significar milhares ou mesmo milhões de casos. Isto será uma bela carga para o seu sistema.

Um modo mais plausível de realizar esta tarefa seria por meio de um pipe (|) e a ferramenta **xargs** com o **rm** de argumento. Desta forma, o comando **rm** somente será invocado quando a linha de comando estiver cheia, ao invés de fazê-lo para cada arquivo individualmente. Veja o [Chapter 5, Redirecionamento de E/S](#) para saber mais sobre redirecionamento de E/S para facilitar as tarefas quotidianas.

Mais adiante (em 1999 de acordo com as páginas man, depois de 20 anos de **find**), o **locate**) foi desenvolvido. Este programa é mais fácil de usar, porém também mais restrito que o **find**, uma vez que sua saída se baseia em um índice de arquivos que é atualizado somente uma vez por dia. Por outro lado, uma pesquisa feita com a base de dados do **locate** demanda menos recursos do sistema quando comparado com o **find** e, assim, exibe os resultados quase que instantaneamente.

A maior parte das distribuições Linux atualmente usa o **slocate**, locate seguro, uma versão moderna do **locate** que impede que os usuários obtenham informações sobre arquivos que se encontram em locais para os quais eles não possuem permissão de acesso. Um exemplo são os arquivos na pasta home do *root*, que normalmente não são acessíveis ao público. Um usuário que deseje encontrar outro que tenha conhecimento do shell C pode informar o comando **locate** **.cshrc**, para exibir os nomes de todos os usuários que possuam um arquivo de configuração adaptado para o shell C. Supondo que os usuários *root* e *jenny* estejam usando o C shell, apenas o arquivo */home/jenny/.cshrc* será exibido, permanecendo oculto o que se encontra na pasta home do *root*. Na maior parte dos sistemas, o **locate** é um link simbólico para o **slocate**:

```
billy:~> ls -l /usr/bin/locate  
lrwxrwxrwx 1 root slocate 7 Oct 28 14:18 /usr/bin/locate -> slocate*
```

A usuária *tina* poderia usar o **locate** para encontrar um aplicativo que estivesse procurando:

```
tina:~> locate acroread  
/usr/share/icons/hicolor/16x16/apps/acroread.png  
/usr/share/icons/hicolor/32x32/apps/acroread.png
```

```
/usr/share/icons/loicolor/16x16/apps/acroread.png
/usr/share/icons/loicolor/32x32/apps/acroread.png
/usr/local/bin/acroread
/usr/local/Acrobat4/Reader/intellinux/bin/acroread
/usr/local/Acrobat4/bin/acroread
```

Diretórios que não contém o nome `bin` não podem conter o programa, eles não guardam arquivos executáveis. Sobram três opções. O arquivo em `/usr/local/bin` é aquele pelo qual a *tina* procura: é um link para um script do shell que inicia o programa:

```
tina:~> file /usr/local/bin/acroread
/usr/local/bin/acroread: symbolic link to ../Acrobat4/bin/acroread

tina:~> file /usr/local/Acrobat4/bin/acroread
/usr/local/Acrobat4/bin/acroread: Bourne shell script text executable

tina:~> file /usr/local/Acrobat4/Reader/intellinux/bin/acroread
/usr/local/Acrobat4/Reader/intellinux/bin/acroread: ELF 32-bit LSB
executable, Intel 80386, version 1, dynamically linked (uses
shared libs), not stripped
```

Para manter o caminho tão curto quanto possível, de forma que o sistema não precise realizar uma extensa busca a cada vez que um usuário quiser executar um comando, nós adicionamos o diretório `/usr/local/bin` ao `path`, deixando fora outros diretórios que contém apenas arquivos binários de um programa específico, enquanto o `/usr/local/bin` guarda também outros programas.

Novamente, uma descrição de todos os recursos do **find** e do **locate** pode ser encontrada nas páginas Info.

## O comando grep

### Filtro de linha genérico

Um programa simples porém muito útil, o **grep** é usado para filtrar linhas de entrada e retornar certos padrões para a saída. Existem literalmente milhares de aplicações para o programagrep. No exemplo abaixo, o *jerry* usa o **grep** para ver como ele fez aquela coisa com o comando **find**:

```
jerry:~> grep -a find .bash_history
find . -name userinfo
man find
find ../ -name common.cfg
```

### Histórico de busca

Também é útil nestes casos a função de busca do **bash**, acionada pressionando-se simultaneamente **Ctrl+R**, como no exemplo onde queremos verificar de novo o que fizemos com o comando **find**:

```
thomas ~> ^R
(reverse-i-search)`find': find `/home/thomas` -name *.xml
```

Digite o seu critério de busca no prompt. Quanto maior a quantidade de caracteres digitada, mais restrita se torna a busca. Realiza-se uma leitura do histórico de comandos para esta sessão do shell (que é escrita no `.bash_history` na sua pasta home quando você finaliza a sessão). A ocorrência mais recente do seu critério de busca é então exibida. Caso você deseje ver comandos anteriores que contém a mesma seqüência, pressione novamente **Ctrl+R**.

Veja as páginas Info do **bash** para saber mais.

Todos os UNIXes com um mínimo de decência também possuem um dicionário online. O Linux é um deles. O dicionário é uma lista de palavras conhecidas gravadas em um arquivo chamado `words`, armazenado na pasta `/usr/share/dict`. Para uma verificação rápida da grafia correta de um vocábulo, não é necessário recorrer a qualquer aplicativo gráfico.

```
william:~> grep penguin /usr/share/dict/words
```

```
william:~> grep penguin /usr/share/dict/words
penguin
penguins
```

### Dicionário vs. lista de palavras

Algumas distribuições oferecem o comando **dict**, , que possui outros recursos além de uma mera lista de palavras.

Quem é o proprietário daquela pasta home perto da minha? Ei, ali está o seu número de telefone

```
lisa:~> grep gdbruyne /etc/passwd
gdbruyne:x:981:981:Guy Debruyne, tel 203234:/home/gdbruyne:/bin/bash
```

E qual era mesmo o email do Arno?

```
serge:~/mail> grep -i arno *  
sent-mail: To: <Arno.Hintjens@celeb.com>  
sent-mail: On Mon, 24 Dec 2001, Arno.Hintjens@celeb.com wrote:
```

O **find** e o **locate** muitas vezes são usados em conjunto com o **grep** para refinar as pesquisas. Para saber mais, veja o [Chapter 5, Redirecionamento de E/S](#) sobre redirecionamento de E/S.

## Caracteres especiais

Caracteres que possuem um significado especial para o shell precisam ser *demarcados*. O caracter de demarcação no Bash é a barra invertida, como o é na maior parte dos shells; isto retira o significado especial do caracter seguinte. O shell reconhece diversos caracteres especiais, entre os mais comuns /, ., ? e \*. Uma lista completa pode ser encontrada nas páginas Info e na documentação do seu shell.

Por exemplo, caso você queira exibir o arquivo "\*" ao invés de todos os arquivos em um diretório, você precisaria usar

```
less \*
```

O mesmo vale para nomes de arquivos que contém um espaço:

```
cat This\ File
```

## Outros modos de averiguar o conteúdo de arquivos

### Geral

Além do **cat**, que na verdade não faz muita coisa além de mandar os arquivos para a saída padrão, existem outras ferramentas para verificar o conteúdo de arquivos.

O modo mais fácil, óbvio, é fazer uso de ferramentas gráficas em lugar da linha de comando. Na introdução nós já demos uma pincelada em um aplicativo de escritório, o OpenOffice.org. Outros exemplos são o GIMP (iniciado com **gimp** na linha de comando), o GNU Image Manipulation Program; **xpdf** para ver arquivos em Portable Document Format (PDF); GhostView (**gv**) para ver arquivos PostScript; Mozilla/FireFox, **links** (um navegador modo texto), Konqueror, Opera e muitos outros para visualizar conteúdo web; XMMS, CDplay e outros para conteúdo multimídia; AbiWord, Gnumeric, KOffice e etc para diversos tipos de aplicativos de escritório e assim por diante. Existem milhares de aplicativos Linux, listá-los levaria dias.

Ao invés disso, nós nos concentraremos no shell, em aplicativos modo texto, que são a base para todos os outros aplicativos. Estes comandos funcionam melhor em um ambiente texto e em arquivos que contém texto. Em caso de dúvida, verifique com o comando **file**.

Então vejamos quais ferramentas texto nós temos que são úteis para ver o interior dos arquivos.

### Problemas com fontes

Ferramentas de texto puro, como as que iremos discutir agora, muitas vezes apresentam problemas com textos "puros" por conta da codificação usada nestes arquivos. Caracteres especiais, como aqueles acentuados, ideogramas chineses e caracteres de idiomas que utilizam conjuntos distintos do padrão *en\_US* e etc são exibidos de maneira errônea ou substituídos por símbolos ilegíveis. Estes problemas são discutidos na [the section called "Configurações de Região"](#).

### "less is more"

Sem dúvida, cedo ou tarde você ouvirá alguém dizer esta frase quando estiver trabalhando em um ambiente UNIX. Um pouco da história do UNIX torna tudo claro:

- No início havia o **cat**. A saída era exibida de forma incontrolável.
- Então veio o **pg**, que talvez ainda possa ser encontrado em UNIXes mais antigos. Este comando exibe texto na saída uma página por vez.
- O **more** foi uma versão revisada do **pg**. Este comando ainda está disponível para todos os sistemas Linux.
- O **less** é a versão GNU do more e possui mais recursos, como a demarcação de critérios de busca, o retorno a páginas anteriores e etc. A sintaxe é muito simples:

```
less name_of_file
```

Mais informações nas páginas Info

Na verdade você já conhecia os paginadores, eis que são eles que cuidam da exibição das páginas man.

### Os comandos head e tail

Estes dois comandos exibem, respectivamente, as primeiras/últimas n linhas de um arquivos. Para saber

quais foram os últimos dez comandos informados:

```
tony:~> tail -10 .bash_history
locate configure | grep bin
man bash
cd
xawtv &
grep usable /usr/share/dict/words
grep advisable /usr/share/dict/words
info quota
man quota
echo $PATH
frm
```

O **head** funciona de modo semelhante. O comando **tail** possui um recurso, muito útil, que exhibe continuamente as últimas linhas de um arquivo que é alterado constantemente. Esta opção `-f` normalmente é utilizada por administradores de sistemas para verificar arquivos de log. Mais informações podem ser obtidas nos arquivos de documentação do sistema.

## Interligando arquivos

### Tipos de links

Agora que sabemos mais sobre arquivos e sua representação no sistema de arquivos, entender os links (ou atalhos) é moleza. Um link nada mais é do que um meio de ligar dois ou mais diferentes nomes de arquivo ao mesmo conjunto de dados. Existem duas formas de alcançar este objetivo:

- Link rígido (ou hard link): Associa dois ou mais nomes de arquivo com o mesmo inode. Links rígidos compartilham os mesmos blocos de dados no disco, embora continuem a se comportar como arquivos independentes.

Existe uma desvantagem imediata: links rígidos não podem se espalhar em diferentes partições, já que os números de inode são únicos somente no interior de uma dada partição.

- Links maleáveis ou simbólicos (ou soft links / symlinks): um arquivos pequeno que é um ponteiro para outro arquivo. Um link simbólico contém o caminho para o arquivo alvo, ao invés de uma localização física no disco rígido. Uma vez que os inodes não são utilizados neste sistema, os links simbólicos podem se dispersar por diferentes partições.

Os dois tipos de links se comportam de forma semelhante, mas não são a mesma coisa, como ilustrado no esquema abaixo:

### Figure 3.2. Mecanismos dos links rígidos e simbólicos



Observe que ao remover o arquivo alvo de um link simbólico faz com que este se torne inútil.

Cada arquivo regular é, a princípio, um link rígido. Links rígidos não podem se espalhar por diferentes partições, vez que eles referenciam inodes, os quais são únicos dentro de uma dada partição.

Pode-se argumentar que existe um terceiro tipo de link, o link *do desktop*, que é semelhante a um atalho no MS Windows. Existem arquivos que contém metadados que somente podem ser interpretados por um gerenciador de arquivos gráfico. Para o kernel e para o shell estes links são apenas arquivos comuns. Eles podem ter o sufixo *.desktop* ou *.lnk*; um exemplo pode ser encontrado em `~/ .gnome-desktop`:

```
[dupont@boulot .gnome-desktop]$ cat La\ Maison\ Dupont
[Desktop Entry]
Encoding=Legacy-Mixed
Name=La Maison Dupont
Type=X-nautilus-home
X-Nautilus-Icon=temp-home
URL=file:///home/dupont
```

Este exemplo é do desktop do KDE

```
[lena@venus Desktop]$ cat camera
[Desktop Entry]
Dev=/dev/sda1
FSType=auto
Icon=memory
MountPoint=/mnt/camera
Type=FSDevice
X-KDE-Dynamic-Device=true
```

Criar estes links é muito simples através dos recursos do seu ambiente gráfico, Caso precise de ajuda, a documentação do seu sistema deve ser a sua primeira fonte de recursos.

Na próxima seção, nós estudaremos a criação dos links simbólicos no estilo UNIX utilizando a linha de comando.



## Criando links simbólicos

O link simbólico é particularmente interessante para usuários iniciantes: eles são fáceis de visualizar e não é preciso se preocupar com partições.

O comando para criar links é **ln**. Para criar symlinks é preciso informar a opção **-s**:

```
ln -s targetfile linkname
```

No exemplo abaixo, o usuário *freddy* cria um link em um subdiretório do seu diretório home para um diretório em outra parte do sistema:

```
freddy:~/music> ln -s /opt/mp3/Queen/ Queen
freddy:~/music> ls -l
lrwxrwxrwx 1 freddy freddy 17 Jan 22 11:07 Queen -> /opt/mp3/Queen
```

Links simbólicos são sempre arquivos muito pequenos, enquanto links rígidos possuem o mesmo tamanho do arquivo original.

O uso de links simbólicos é muito difundido. Eles são muito usados para economizar espaço em disco, para criar cópias de um arquivo para satisfazer os requisitos de instalação de um novo programa que espera que um dado arquivo se encontre em um outro local, são usados para consertar scripts que de repente precisam ser executados em um novo ambiente e podem geralmente evitar muito trabalho. Um administrador de sistemas pode decidir mover os diretórios home de todos os usuários para um novo lugar, o *disco2*, por exemplo. Porém, se ele quiser que tudo funcione como antes, como o arquivo */etc/passwd*, com um esforço mínimo ele irá criar um symlink na */home* para o novo local */disk2/home*.

## Segurança de arquivos

### Direitos de acesso: A primeira linha de defesa do Linux

O modelo de segurança no Linux é baseado naquele usado nos sistemas UNIX e é tão rígido quanto este (e as vezes até mais), que é muito robusto. Em um sistema Linux, cada arquivo é de propriedade de um usuário e de um grupo de usuários. Existe ainda uma terceira categoria de usuários, aqueles que não são o proprietário, nem fazem parte do grupo de usuários com direitos a ele. Para cada uma dessas categorias de usuários, permissões de leitura, escrita e execução podem ser garantidas ou negadas.

Nós já usamos a opção *long* para listar arquivos usando o comando **ls -l**, embora por razões diversas. Este comando também exibe permissões de arquivo para estas três categorias de usuários; eles são indicados pelos nove caracteres que se seguem ao inicial, que é o indicador do tipo de arquivo no início da linha de propriedades. As como visto nos exemplos abaixo, os três primeiros caracteres nesta seqüência de nove exibem os direitos de acesso para o usuário proprietário do arquivo. As três seguintes são referentes ao grupo de usuários com direitos sobre o arquivo e as três últimas para os outros usuários. As permissões são apresentadas sempre na mesma ordem: leitura, escrita e execução para o usuário, grupo e outros. Alguns exemplos:

```
marise:~> ls -l To_Do
-rw-rw-r-- 1 marise users      5 Jan 15 12:39 To_Do
marise:~> ls -l /bin/ls
-rwxr-xr-x 1 root   root   45948 Aug  9 15:01 /bin/ls*
```

O primeiro arquivo é um arquivo comum (primeiro traço). Usuários com o nome *marise* ou pertencentes ao grupo *usuários* podem ler e escrever (mudar/mover/apagar) o arquivo, mas eles não podem executá-lo (segundo e terceiro traços). Todos os demais usuários podem também ler o arquivo, porém não podem neles escrever ou executá-los (quarto e quinto traços);

O segundo exemplo é um arquivo executável, a diferença: todos podem executar este programa, mas é preciso ser o *root* para alterá-lo.

As páginas Info explicam como o comando **ls** lida em detalhes com a exibição de direitos de acesso, veja a seção *Quais informações são listadas*.

Para simplificar o uso dos comandos, ambos, direitos ou modos de acesso e grupos de usuários têm um código. Veja a tabela abaixo.

**Table 3.7. Códigos de modo de acesso**

Código	Significado
0 ou -	O direito de acesso que supostamente deveria estar neste local não é garantido.
4 ou r	o direito de leitura é garantido ao usuário ou categoria definido neste local
2 ou w	permissão de escrita é garantida ao usuário ou categoria definida neste local
1 ou x	permissão para execução é garantida à categoria de usuário definida neste local

**Table 3.8. Códigos de grupos de usuários**

Código	Significado
--------	-------------

u	permissões de usuário
g	permissões de grupos
o	permissões dos outros

Este esquema intuitivo é aplicado de modo muito estrito, o que permite um alto nível de segurança, mesmo sem uma rede segura. Dentre outras funções, o esquema de segurança toma conta das permissões de acesso dos usuários aos programas, pode fornecer arquivos de acordo com a demanda e proteger dados sensíveis, como pastas home e arquivos de configuração do sistema.

Você deve saber qual é o seu nome de usuário. Se não souber, ele pode ser exibido usando o comando **id**, que também informa a qual seu grupo padrão, bem como outros dos quais eventualmente seja membro:

```
tilly:~> id
uid=504(tilly) gid=504(tilly) groups=504(tilly),100(users),2051(org)
```

O seu nome de usuário também fica armazenado na variável de ambiente **USER**:

```
tilly:~> echo $USER
tilly
```

## As ferramentas

### O comando **chmod**

Uma consequência normal de aplicar um conjunto estrito de permissões, às vezes um incômodo, é que os direitos de acesso precisarão ser alterados por um sem fim de razões. Usamos o comando **chmod** para fazê-lo e *chmodar* se tornou quase que um verbo aceitável no Português, que significa alterar os modos de acesso de um arquivo. O comando **chmod** pode ser usado com opções alfanuméricas ou numéricas, qualquer que seja a sua preferência.

O exemplo abaixo usa opções alfanuméricas de forma a solucionar um problema que muitas vezes acontece com usuários novatos:

```
asim:~> ./hello
bash: ./hello: bad interpreter: Permission denied

asim:~> cat hello
#!/bin/bash
echo "Hello, World"

asim:~> ls -l hello
-rw-rw-r-- 1 asim asim 32 Jan 15 16:29 hello

asim:~> chmod u+x hello

asim:~> ./hello
Hello, World

asim:~> ls -l hello
-rwxrwx-r-- 1 asim asim 32 Jan 15 16:29 hello*
```

O operadores **+** e **-** são usados para garantir ou negar um dado direito a um dado grupo. Combinações separadas por vírgulas são permitidas. As páginas Info e man possuem exemplos úteis. Aqui vai outro, que faz o arquivo do exemplo anterior se tornar um arquivo particular do usuário *asim*:

```
asim:~> chmod u+rwx,go-rwx hello

asim:~> ls -l hello
-rwx----- 1 asim asim 32 Jan 15 16:29 hello*
```

O tipo de problema que resulta em uma mensagem de erro dizendo que foi negada permissão em algum lugar é normalmente um problema de direitos de acesso. Mais ainda, comentários como “Funcionava ontem,” e “Quando eu executo como root ele funciona,” são normalmente indícios de um problema de permissão.

Quando usar o **chmod** com argumentos numéricos, os valores para cada direito de acesso garantido tem de ser contado em conjunto para um grupo. Assim obtemos um número de três dígitos, que é um valor simbólico para as configurações que o **chmod** tem que fazer. A tabela a seguir lista as combinações mais comuns:

**Table 3.9. Proteção de arquivos com o **chmod****

Comando	Significado
<b>chmod 400 file</b>	Para proteger um arquivo contra sobrescrição acidental.
<b>chmod 500 directory</b>	Para proteger-se de si mesmo contra remoções, alterações de nome ou mudança de diretório.
<b>chmod 600</b>	

<b>chmod 600 file</b>	Um arquivo privado, alterável somente pelo usuário que informou este comando.
<b>chmod 644 file</b>	Um arquivo que pode ser lido por qualquer um, mas somente pode ser alterado pelo usuário que informou o comando.
<b>chmod 660 file</b>	Usuários que pertençam aos seu grupo podem mudar este arquivo, outros não possuem quaisquer permissões de acesso a ele.
<b>chmod 700 file</b>	Protege um arquivo contra qualquer acesso de outros usuários, ao mesmo tempo em que dá acesso irrestrito ao usuário.
<b>chmod 755 directory</b>	Para arquivos que devem ser legíveis e executáveis por outros, mas alteráveis apenas pelo usuário que informou o comando.
<b>chmod 775 file</b>	Modo padrão de compartilhamento de um arquivo por um grupo.
<b>chmod 777 file</b>	Qualquer um pode fazer o que quiser com o arquivo.

Se você informar um número com menos de três dígitos como argumento ao **chmod**, os caracteres omitidos serão substituídos por zeros, a partir da esquerda. Na verdade, existe um quarto caractere em sistemas Linux, que precede os outros três e estabelece modos especiais de acesso. Tudo a respeito deles e muito mais pode ser descoberto nas páginas Info.

### Efetuando o login em outro grupo

Quando você digita **id** na linha de comando, é-lhe apresentada uma lista de todos os grupos aos quais você poderia pertencer, precedidos por seu nome de usuário, seu ID, o nome do grupo e o ID que você está usando no momento. Entretanto, em muitos sistemas Linux, você só pode estar ativo em um grupo por vez. Por padrão, este grupo ativo ou *primário* é aquele que lhe é atribuído no arquivo `/etc/passwd`. O quarto campo deste arquivo guarda o ID do grupo primário do usuário, que está armazenado no arquivo `/etc/group`. Um exemplo:

```
asim:~> id
uid=501(asim) gid=501(asim) groups=100(users),501(asim),3400(web)
```

```
asim:~> grep asim /etc/passwd
asim:x:501:501:Asim El Baraka:/home/asim:/bin/bash
```

```
asim:~> grep 501 /etc/group
asim:x:501:
```

O quarto campo na linha do `/etc/passwd` contém o valor “501”, que representa o grupo *asim* no exemplo acima. Do `/etc/group` nós podemos obter o nome a que se refere este ID de grupo. Quando se conectar ao sistema, este será o grupo a que *asim* pertencerá.

### Esquema de grupo privado do usuário

De modo a ensinar mais flexibilidade, a maior parte dos sistemas Linux segue o chamado *esquema de grupo privado do usuário*, que atribui a cada usuário um grupo. Este grupo é aquele que contém somente um usuário, e daí o nome “grupo privado”. Normalmente este grupo é homônimo ao login do usuário, o que pode ser um pouco confuso.

Além de seu próprio grupo privado, o usuário *asim* pode também estar nos grupos *users* e *web*. Uma vez que estes grupos são secundários para este usuário, ele precisará usar o comando **newgrp** para efetuar o login em quaisquer destes grupos (use **gpasswd** para configurar a senha do grupo). No exemplo, *asim* precisa criar arquivos que sejam de propriedade do grupo *web*.

```
asim:/var/www/html> newgrp web
```

```
asim:/var/www/html> id
uid=501(asim) gid=3400(web) groups=100(users),501(asim),3400(web)
```

Agora, quando *asim* criar novos arquivos, eles serão de propriedade do grupo *web* ao invés de serem do grupo *asim*

```
asim:/var/www/html> touch test
```

```
asim:/var/www/html> ls -l test
-rw-rw-r-- 1 asim web 0 Jun 10 15:38 test
```

Efetuar o login em um novo grupo torna desnecessário usar o **chown** (veja [the section called “Alterando as titularidades de usuário e grupo”](#)) ou chamar o seu administrador para alterar as permissões para você.

Veja a página man do **newgrp** para saber mais.

### A máscara de arquivo

Quando um novo arquivo é salvo em algum local, ele primeiro é submetido aos procedimentos de segurança do Linux. Arquivos sem permissões não existem no Linux. O padrão de permissões de arquivos é determinado pela *máscara* de criação de novos arquivos. O valor desta máscara pode ser

exibido usando o comando **umask**:

```
bert:~> umask
0002
```

Ao invés de adicionar valores simbólicos a cada um, como no **chmod**, para calcular as permissões em um novo arquivo é necessário subtrair o total de permissões possíveis. No exemplo acima, entretanto, vemos 4 valores sendo exibidos, embora existam apenas 3 categorias de permissões: *usuário*, *grupo* e *outros*. O primeiro zero é parte das configurações de atributos de arquivos especiais, que discutiremos na [the section called "Alterando as titularidades de usuário e grupo"](#) e [the section called "SUID and SGID"](#). É possível também que este zero não seja exibido ao informar o comando **umask** no seu sistema, exibindo-se somente os três algarismos que representam a máscara de criação de arquivos padrão.

Cada sistema de base UNIX possui uma função de sistema para a criação de novos arquivos, que é chamada a cada vez que o usuário usa um programa que cria novos arquivos, como, por exemplo, quando estiver baixando um arquivo da internet, quando salvar um novo documento texto e assim por diante. Esta função cria tanto novos diretórios, como novos arquivos. Permissões completas para leitura, escrita e execução são garantidas a todos quando se criam novos diretórios. Quando se criam novos arquivos, esta função irá garantir permissões de leitura e escrita para todos, mas negar permissão de execução para todas as categorias de usuários. Assim, antes da máscara ser usada, um diretório possui permissões *777* ou *rw-rw-rw-*, um arquivo simples *666* ou *rw-rw-rw-*.

O valor da *umask* é subtraído destas permissões padrão após a criação de um novo diretório ou arquivo pela função. Assim, um diretório terá, por padrão, permissões *775* e um arquivo *664*, se o valor da máscara for *(0)002*. Isto é mostrado no exemplo abaixo:

```
bert:~> mkdir newdir

bert:~> ls -ld newdir
drwxrwxr-x  2 bert  bert          4096 Feb 28 13:45 newdir/

bert:~> touch newfile

bert:~> ls -l newfile
-rw-rw-r--  1 bert  bert           0 Feb 28 13:52 newfile
```

## Arquivos versus diretórios

Por padrão, um diretório obtém mais permissões: ele sempre possui a permissão de *execução*. Sem ela, o diretório seria inacessível. Faça um teste e dê um **chmod 644** em um diretório.

Se você efetuar o login em outro grupo usando o comando **newgrp**, a máscara continuará inalterada. Assim, se ela estiver configurada como *002*, os arquivos e diretórios que você criar enquanto estiver no novo grupo também serão acessíveis a outros membros daquele grupo; não é necessário usar o **chmod**.

O *root* geralmente possui permissões de criação de arquivos mais estritas.

```
[root@estoban root]# umask
022
```

Estes padrões são configurados para todo o sistema nos arquivos de configuração dos recursos do shell, como, por exemplo, */etc/bashrc* ou */etc/profile*. Você pode alterá-los no seu arquivo pessoal de configuração do shell, veja o [Chapter 7, Lar doce /lar \(/home\)](#) para saber sobre a configuração do seu ambiente shell.

## Alterando as titularidades de usuário e grupo

Quando um arquivo é de propriedade do usuário ou grupo incorreto, a falha pode ser corrigida com os comandos **chown** e **chgrp**. Alterar a titularidade de um arquivo é uma tarefa freqüente na administração de um sistema em ambientes em que os arquivos são compartilhados por um grupo de usuários. Ambos os comandos são muito flexíveis, como é possível perceber através da opção `--help`.

O comando **chown** pode ser usado para alterar tanto o usuário, como o grupo titular de um arquivo, ao passo em que o **chgrp** só altera o grupo. É claro que o sistema irá verificar se o usuário informando estes comandos possui permissões suficientes para o arquivo que deseja alterar.

Para alterar apenas o usuário que é titular do arquivo, use esta sintaxe:

```
chown newuser file
```

Se você usar o sinal de dois pontos após o nome do usuário (olha as páginas Info), a titularidade do grupo também será alterada para o grupo primário do usuário que informou o comando. Em um sistema Linux, cada usuário possui seu próprio grupo, então este comando serve para tornar os arquivos privados:

```
jacky:~> id
uid=1304(jacky) gid=(1304) groups=1304(jacky),2034(pproject)

jacky:~> ls -l my_report
-rw-rw-r--  1 jacky  project          29387 Jan 15 09:34 my_report

jacky:~> chown jacky: my_report
```

```
jacky:~> chmod o-r my_report
```

```
jacky:~> ls -l my_report
-rw-rw---- 1 jacky jacky      29387 Jan 15 09:34 my_report
```

Se a *jacky* quiser compartilhar este arquivo, sem precisar dar permissão de escrita a todos, ela pode usar o comando **chgrp**:

```
jacky:~> ls -l report-20020115.xls
-rw-rw---- 1 jacky jacky  45635 Jan 15 09:35 report-20020115.xls
```

```
jacky:~> chgrp project report-20020115.xls
```

```
jacky:~> chmod o= report-20020115.xls
```

```
jacky:~> ls -l report-20020115.xls
-rw-rw---- 1 jacky project 45635 Jan 15 09:35 report-20020115.xls
```

Desta forma, usuários do grupo *projeto* poderão trabalhar neste arquivo. Usuários que não participam do grupo não terão permissão para nada.

Ambos **chown** e **chgrp** podem ser usados para alterar a titularidade recursivamente, usando a opção **-R**. Neste caso, todos os arquivos subjacentes e subdiretórios de um dado diretório pertencerão a dado usuário ou grupo.

## Restrições

Na maioria dos sistemas, o uso dos comandos **chown** e **chgrp** é restrito aos usuários privilegiados. Se você não for o administrador do sistema, não conseguirá alterar as permissões de usuários e grupos por razões de segurança. Se o uso destes comandos não fosse restrito, usuários de má-fé poderiam atribuir a titularidade de arquivos a outros usuários e/ou grupos e alterar o comportamento dos ambientes destes usuários ou mesmo causar danos aos arquivos de outros usuários.

## Modos especiais

Para evitar que o administrador do sistema tenha que ficar resolvendo problemas de permissão o tempo inteiro, direitos especiais de acesso podem ser dados a diretórios inteiros ou programas separados. Existem três modos especiais:

- Modo sticky bit: Após a execução de uma tarefa, o comando é mantido na memória do sistema. Originalmente, este recurso era muito usado para economizar memória: grandes tarefas são carregadas na memória uma única vez. Mas hoje em dia a memória tornou-se barata e existem técnicas melhores de gerenciá-la, então este recurso deixou de ser utilizado para otimizar a manipulação de arquivos individuais. Quando aplicada a um diretório inteiro, contudo, o sticky bit possui um significado diferente. Neste caso, um usuário somente poderá alterar os arquivos no diretório quando for o titular do arquivo ou tiver as permissões necessárias. Este recurso é utilizado em diretórios como o `/var/tmp`, que precisa ser acessível a todos, mas no qual não convém que os usuários possam deletar os arquivos uns dos outros. O sticky bit é indicado por um `t` ao final do campo de permissões de acesso:

```
mark:~> ls -ld /var/tmp
drwxrwxrwt 19 root root      8192 Jan 16 10:37 /var/tmp/
```

O sticky bit é configurado usando o comando **chmod o+t** *directory*. A origem histórica do “t” está ligada ao recurso *save Text access* do UNIX

- SUID (configurar ID de usuário) e SGID (configurar ID de grupo): são representados pelo caracter `s` no campo de permissão do usuário ou grupo. Quando este modo está configurado em um arquivo executável, ele será executado com as permissões do arquivo ao invés daquelas do usuário que informou o comando, permitindo, assim, acesso aos recursos do sistema. Discutiremos isto mais adiante no [Chapter 4, Processos](#).
- SGID (set group ID) em um diretório: neste caso específico, todos os arquivos criados no diretório irão ser do mesmo grupo que o diretório (ao passo em que o comportamento normal é que tenha o mesmo grupo do usuário que os criou). Desta forma, os usuários não precisam se preocupar com as permissões de acesso ao compartilharem um diretório:

```
mimi:~> ls -ld /opt/docs
drwxrws--- 4 root users      4096 Jul 25 2001 docs/
```

```
mimi:~> ls -l /opt/docs
-rw-rw---- 1 mimi users      345672 Aug 30 2001-Council.doc
```

Este é o modo padrão de compartilhar arquivos no UNIX.

## Arquivos pré-existentes permanecem inalterados!

Arquivos que estão apenas sendo movidos para um diretório SGID, mas que foram criados em outro local, permanecem com suas propriedades originais de usuário e

grupo. Isto pode ser um pouco confuso.

## Resumo

No UNIX, assim como no Linux, todos os entes são de uma maneira ou de outra apresentados ao sistema como arquivos com suas devidas propriedades. O uso de caminhos (pré-definidos) permite aos usuários e ao administrador do sistema encontrar, ler e manipular os arquivos.

Realizamos os nossos primeiros passos para nos tornarmos peritos: discutimos as estruturas real e fictícia do sistema de arquivos e aprendemos sobre o modelo de segurança de arquivos do Linux, bem como diversas outras medidas de segurança que são tomadas em todos os sistemas por padrão.

O shell é a ferramenta mais importante para a interação com o sistema. Nós aprendemos vários comandos neste capítulo, os quais encontram-se listados na tabela abaixo.

**Table 3.10. Novos comandos no capítulo 3: Arquivos e o sistema de arquivos**

Comando	Significado
<b>bash</b>	Shell do GNU
<b>cat</b> arquivo(s)	Envia o conteúdo do(s) arquivo(s) para a saída padrão.
<b>cd</b> diretório	Entra no diretório. <b>cd</b> é um comando interno do <b>bash</b> .
<b>chgrp</b> novogrup arquivo(s)	Muda o grupo do arquivo arquivo(s) para <i>novogrupa</i>
<b>chmod</b> modo arquivo(s)	Muda as permissões de acesso do arquivo(s)
<b>chown</b> novoproprietario[:[novogrupa]] file(s)	Altera o usuário e grupo titulares.
<b>cp</b> arquivofonte arquivualvo	Copia arquivofonte para arquivualvo.
<b>df</b> arquivo	Relata o espaço em uso na partição que contém o arquivo.
<b>echo</b> string	Exibe uma linha do texto
<b>export</b>	Parte do <b>bash</b> que anuncia variáveis e seus valores ao sistema.
<b>file</b> nomedoarquivo	Determina o tipo de arquivo de nomedoarquivo.
<b>find</b> caminho expressão	Encontra arquivos no sistema de arquivos
<b>grep</b> PADRÃO arquivo	Imprime linhas no arquivo que contém o critério de pesquisa.
<b>head</b> arquivo	Envia o começo do arquivo para a saída padrão
<b>id</b>	Imprime os nomes do usuário e grupo real e efetivo.
<b>info</b> comando	Lê a documentação sobre o <b>comando</b> .
<b>less</b> arquivo	Visualiza o arquivo com um visualizador poderoso.
<b>ln</b> arquivualvo nomedolink	Cria um link de nome nomedolink para o arquivualvo.
<b>locate</b> critériodebusca	Exibe todos os arquivos acessíveis que se adequam ao critério buscado.
<b>ls</b> arquivo(s)	Exibe o conteúdo de um diretório.
<b>man</b> comando	Formata e exibe páginas de manual online (no sistema) para o <b>comando</b> .
<b>mkdir</b> novodir	Cria um novo diretório vazio.
<b>mv</b> antigoarquivo novoarquivo	Renomeia ou moveantigoarquivo.
<b>newgrp</b> nomedogrupa	Efetua o login em um novo grupo.
<b>pwd</b>	Exibe o diretório atual.
<b>quota</b>	Exibe o espaço de disco em uso e seus limites.
<b>rm</b> arquivo	Apaga arquivos e diretórios.
<b>rmdir</b> diretório	Apaga diretórios.
<b>tail</b> arquivo	Imprime o final do arquivo.
<b>umask</b> [value]	Exibe ou altera o modo de criação de novos arquivos.
<b>wc</b> file	Counts lines, words and characters in file.
<b>which</b> comando	Mostra o caminho absoluto para o <b>comando</b> .

Também enfatizamos o fato de que você deve LER AS PÁGINAS MAN. Esta documentação é o seu kit de primeiros socorros e contém as respostas para muitas perguntas. A lista acima contém os comandos básicos que você irá usar diariamente, mas eles podem fazer muito mais do que aquilo que foi dito aqui. A leitura da documentação lhe dará o controle necessário.

Por último, mas não menos importante, um panorama das permissões de arquivo:

**Table 3.11. Permissões de arquivo**

Quem\Oque	r(ead)	w(rite)	(e)x(ecutar)
u(suário)	4	2	1

g(rupo)	4	2	1
o(utros)	4	2	1

## Exercícios

Efetue o login com sua ID de usuário padrão

### Partições

- Em qual partição está a sua pasta home?
- Quantas partições existem no seu sistema?
- Qual o tamanho total de sua instalação do Linux?

### Caminhos

- Exibe o seu caminho de busca
- Exporte um caminho sem sentido informando, por exemplo, **export PATH=blah** e tente listar o conteúdo do diretório.
- Qual é o caminho do seu diretório home? Como outro usuário conseguiria acessar a sua pasta home partindo da dele por meio de um caminho relativo?
- Vá para o diretório tmp no /var.
- Agora vá para share no /usr usando apenas um comando. Altere para doc. Qual o seu diretório atual?

### Passeio pelo sistema

- Vá para o diretório /proc.
- Em qual CPU(s) o sistema está rodando?
- Quanta RAM está sendo usada?
- Quando espaço de swap você possui?
- Quais drivers foram carregados?
- Há quantas horas o sistema está rodando?
- Quais sistemas de arquivo são conhecidos pelo seu sistema?
- Mude para /etc/rc.d | /etc/init.d | /etc/runlevels e escolha o diretório apropriado para o seu nível de execução.
- Quais serviços deveriam estar sendo executados neste nível?
- Quais serviços são executados no modo gráfico, mas não no texto?
- Mude para /etc
- Por quanto tempo o sistema mantém o arquivo de log no qual os logins de usuários são monitorados?
- Qual distribuição você está rodando?
- Existem problemas ou mensagens do dia?
- Quantos usuários existem no seu sistema? Não os conte, deixe que o computador faça isso por você!
- Quantos grupos?
- Onde são armazenadas as informações de zona?
- Os COMOFAZER estão instalados no seu sistema?
- Mude para /usr/share/doc.
- Nomeie três programas que são fornecidos pelo pacote GNU *coreutils*.
- Qual versão do **bash** está instalada no seu sistema?

### Manipulando arquivos

- Crie um novo diretório na sua home.

- Você pode mover este diretório para o mesmo nível da sua home?
- Copie todos os arquivos XPM do `/usr/share/pixmaps` para o diretório novo. O que significa XPM?
- Liste os arquivos em ordem alfabética de trás para frente.
- Mude para a sua home. Crie um novo diretório e copie todos os arquivos do diretório `/etc`. Certifique-se de copiar também todos os arquivos e diretórios que são subdiretórios do `/etc`! (cópia recursiva)
- Mude para o novo diretório e faça um diretório para arquivos iniciados por uma letra maiúsculo e outro para arquivos iniciados por uma letra minúscula. Mova todos os arquivos para os diretórios apropriados. Use o menor número de comandos possível.
- Remova os arquivos restantes.
- Apague o diretório e todo o seu conteúdo usando um único comando.
- Use o **grep** para saber qual script inicia o Servidor de Fontes no nível de execução gráfico.
- Onde está o programa *sendmail*?
- Crie um link simbólico na sua pasta home `/var/tmp`. Verifique se realmente funciona.
- Crie outro link simbólico em sua pasta home para este link. Verifique se funciona. Apague o primeiro link e liste o conteúdo do diretório. O que aconteceu com o segundo link?

## Permissões de arquivo

- Você consegue alterar as permissões de arquivo na `/home`?
- Qual o padrão para a criação de arquivos?
- Altere a titularidade do `/etc` para o seu próprio usuário e grupo.
- Altere as permissões do arquivo `~/.bashrc` de forma que apenas você e seu grupo primário possam lê-lo.
- Informe o comando **locate root**. Você repara em algo diferente?
- Crie um link simbólico para o `root`. Ele pode ser usado?

## Chapter 4. Processos

### Table of Contents

#### [Processos virados do avesso](#)

- [Multiusuário e multitarefa](#)
- [Tipos de processos](#)
- [Atributos de processos](#)
- [Exibindo informações de processos](#)
- [Vida e morte de um processo](#)
- [SUID and SGID](#)

#### [Processo de inicialização, iniciar e desligar](#)

- [Introdução](#)
- [O processo de inicialização](#)
- [Recursos do GRUB](#)
- [Init](#)
- [Níveis de execução do init](#)
- [Desligamento](#)

#### [Gerenciando processos](#)

- [Trabalho para o administrador do sistema](#)
- [Quanto tempo leva?](#)
- [Performance](#)
- [Carga](#)
- [Eu posso fazer alguma coisa como um usuário?](#)

#### [Agendando processos](#)

- [Use aquele tempo ocioso!](#)
- [O comando sleep](#)
- [O comando at](#)
- [Cron e crontab](#)

#### [Resumo](#)

#### [Exercícios](#)



## Abstract

Depois de arquivos, processos são as coisas mais importantes em um sistema UNIX/Linux. Nesse capítulo iremos dar uma olhada em processos mais de perto. Iremos aprender mais sobre:

- Processamento multiusuário e multitarefa
- Tipos de processos
- Controlando processos com sinais diferentes
- Atributos de processos
- O ciclo de vida de um processo
- Inicialização e desligamento do sistema
- SUID e SGID
- Velocidade e responsividade do sistema
- Agendamento de processos
- O sistema Vixie cron
- Como extrair o máximo do seu sistema

## Processos virados do avesso

### Multiusuário e multitarefa

Agora que estamos mais familiarizados com nosso ambiente e somos capazes de nos comunicar um pouco com nosso sistema, é hora de estudar com mais detalhes os processos que podemos iniciar. Nem todo comando inicia apenas um processo. Alguns comandos iniciam uma série de processos, como o **mozilla**; outros, como **ls**, são executados como um único comando.

Além do mais, Linux é baseado em UNIX, onde tem sido comum ter múltiplos usuários executando múltiplos comandos ao mesmo tempo e no mesmo sistema. É claro que algumas medidas devem ser tomadas para que a CPU gerencie todos esses processos, e que alguma funcionalidade deve ser fornecida para que os usuários possam alternar entre os processos. Em alguns casos, um processo deve continuar mesmo que o usuário que o iniciou desconecte do sistema. E os usuários devem ter meios de reativar processos interrompidos.

Iremos explicar a estrutura dos processos em Linux nas próximas seções.

### Tipos de processos

#### Processos interativos

Processos interativos são iniciados e controlados através de uma sessão de terminal. Em outras palavras, deve haver alguém conectado ao sistema para que inicie estes processos; eles não iniciam automaticamente como parte das funções do sistema. Esses processos podem executar em primeiro plano, ocupando o terminal que iniciou o programa, e você não pode iniciar outros aplicativos enquanto esse processo estiver executando em primeiro plano. Como alternativa, eles podem executar em plano de fundo, de forma que o terminal que você iniciou o programa possa aceitar novos comandos enquanto o programa executa. Até agora, nós tratamos principalmente de programas executando em primeiro plano – o tempo para executá-los era muito curto para perceber – mas visualizar um arquivo com o comando **less** é um bom exemplo de um comando ocupando uma sessão de terminal. Nesse caso, o programa ativado está esperando você fazer alguma coisa. O programa ainda está conectado ao terminal de onde ele iniciou, e o terminal só serve para inserir comandos entendidos por esse programa. Outros comandos apenas irão resultar em erros ou em falta de resposta do sistema.

Enquanto o processo executa em plano de fundo, diferentemente, o usuário não é impedido de fazer outras coisas no terminal o qual iniciou o programa, durante sua execução.

O shell oferece um recurso chamado *controle de trabalho* que permite fácil manipulação de múltiplos processos. Esse mecanismo alterna processos entre o primeiro plano e o de fundo. Por esse sistema os programas também podem iniciar em plano de fundo imediatamente.

Executar em plano de fundo é útil apenas para programas que não precisam de interação com o usuário (via shell). Um trabalho é colocado em plano de fundo geralmente quando é esperado que demore bastante para terminar de executar. Para liberar o terminal em questão depois de inserir um comando a letra 'E' comercial é adicionada no final. No exemplo, no modo gráfico, abrimos uma janela terminal extra a partir de uma existente:

```
billy:-> xterm &
```

```

billy:~> jobs
[1]+  Running                  xterm &

```

Todos os recursos de controle de trabalhos são explicados em detalhe nas páginas Info do **bash**, então apenas os usos mais frequentes do controle de trabalhos são listados aqui :

**Table 4.1. Controlando processos**

(parte do) comando	Significado
<b>comando_qualquer</b>	Executa o comando em primeiro plano.
<b>comando_qualquer &amp;</b>	Executa este comando em plano de fundo (libera o terminal)
<b>jobs</b>	(trabalhos) Mostra os comandos que estão executando em plano de fundo.
<b>Ctrl+Z</b>	Suspende (pausa, mas não fecha) um processo executando em primeiro plano.
<b>Ctrl+C</b>	Interrompe(elimina e fecha) um processo executando em primeiro plano.
<b>%n</b>	Cada processo executando no plano de fundo recebe um número atribuído a ele. Usando a expressão %, um trabalho pode ser referido pelo seu número, por exemplo <b>fg %2</b> .
<b>bg</b>	Reativa um programa suspenso no plano de fundo ("bg" vem de "background", traduzido do inglês para "plano de fundo").
<b>fg</b>	Coloca o trabalho de volta em primeiro plano ("fg" vem de "foreground", traduzido do inglês para "primeiro plano").
<b>kill</b>	(matar) Finaliza um processo (veja também os comandos internos das páginas Info do <b>bash</b> )

Exemplos mais práticos podem ser encontrados nos exercícios.

Provavelmente a maioria dos sistemas UNIX são capazes de executar o comando **screen**, o qual é útil quando você realmente quer outro shell para executar comandos. Ao chamar **screen**, uma nova sessão é criada acompanhada de um shell, e/ou comandos especificados, e você pode então movê-la da sua frente. Nessa nova sessão você pode fazer o que quiser. Todos os programas e operações vão executar independentemente do shell em questão. Você pode então desconectar esta sessão, enquanto os programas que você iniciou continuarão executando, e voltar para seu *screen* quando quiser.

Este programa vem de um tempo quando os consoles virtuais ainda não haviam sido inventados, e tudo tinha que ser feito com apenas um terminal de texto. Para os devotos, ainda significa muito em Linux, apesar de já termos consoles virtuais por quase dez anos.

### Processos automáticos

Processos automáticos, ou em lotes, não estão conectados a terminais. Ao invés, são tarefas que podem ser enfileiradas numa área de spooler, onde podem esperar para serem executadas numa política FIFO ("first-in, first-out", traduzido do inglês para "o primeiro a entrar é o primeiro a sair"). Tais tarefas podem ser executadas por um desses dois critérios:

- Em uma data e hora determinada: pelo comando **at**, o qual iremos discutir na segunda parte desse capítulo.
- Quando a carga total do sistema é baixa o bastante para aceitar trabalhos extras: pelo comando **batch**. Por padrão, as tarefas são colocadas em filas onde esperam para serem executadas quando a carga do sistema for inferior a 0.8. Em ambientes grandes, o administrador do sistema pode preferir processamento em lotes quando o processamento envolver grandes quantidades de dados ou quando requerer muitos recursos de um sistema já atarefado. Processamento em lotes também é usado para otimizar a performance do sistema.

### Daemons

Daemons são processos de servidor que executam continuamente. É comum que sejam iniciados junto com o computador, e então esperam em plano de fundo até que o serviço seja requisitado. Um exemplo típico é o daemon de rede *xinetd*, que faz parte da maioria dos processos de inicialização. Depois que o sistema é inicializado, o daemon de rede apenas senta e espera até que um programa cliente, como um cliente de FTP, queira conectar.

### Atributos de processos

Um processo tem uma série de características , que podem ser visualizadas com o comando **ps**:

- O ID de processo, ou PID,: um número de identidade usado para identificar o processo como único.
- O ID do processo pai, ou PPID: o número do processo (PID) que originou este processo.
- Número nice (generoso): o grau de generosidade desse processo diante dos outros processos (não

confundir com prioridade do processo, que é calculado baseado nesse número nice e o quanto o processo usou a CPU recentemente).

- Terminal, ou TTY: terminal ao qual o processo está conectado.
- Usuário real e usuário efetivo (RUID e EUID): o dono do processo. O real é o usuário que executou o comando, o usuário efetivo determina o acesso aos recursos do sistema. O RUID e o EUID geralmente são iguais, e o processo tem os mesmos direitos de acesso que o usuário que executou o comando. Um exemplo para deixar isso claro: o navegador **mozilla** em `/usr/bin` pertence ao usuário `root`:

```
theo:~> ls -l /usr/bin/mozilla
-rwxr-xr-x 1 root root 4996 Nov 20 18:28 /usr/bin/mozilla*
```

```
theo:~> mozilla &
[1] 26595
```

```
theo:~> ps -af
UID      PID  PPID  C  STIME TTY          TIME CMD
theo  26601 26599  0  15:04 pts/5 00:00:00 /usr/lib/mozilla/mozilla-bin
theo  26613 26569  0  15:04 pts/5 00:00:00 ps -af
```

Quando o usuário `theo` inicia este programa, o próprio processo e todos os processos iniciados por este processo pertencerão ao usuário `theo`, e não pelo administrador do sistema. Quando o **mozilla** solicita o acesso a certos arquivos, esse acesso vai ser determinado pelas permissões de `theo` e não de `root`.

- Grupo real e efetivo do dono (RGID e EGID): O grupo real do dono de um processo é o grupo primário do usuário que iniciou o processo. O grupo efetivo geralmente é o mesmo, exceto quando o modo de acesso SGID é aplicado ao arquivo executável do programa.

## Exibindo informações de processos

O comando **ps** é uma das ferramentas para visualizar processos. Esse comando tem várias opções que podem ser combinadas para exibir diferentes atributos de processos.

Se nenhuma opção for especificada, o **ps** apenas informa sobre o shell atual e eventuais processos:

```
theo:~> ps
  PID TTY          TIME CMD
 4245 pts/7        00:00:00 bash
 5314 pts/7        00:00:00 ps
```

Já que isso não fornece informação o suficiente - geralmente, pelo menos centenas de processos estão executando no seu sistema - normalmente iremos selecionar processos em particular na lista de todos os processos com o comando **grep** em um *pipe*, veja [the section called "Redirecionamento de saída com > e |"](#), como nessa linha, que irá exibir todos os processos pertencentes a um usuário particular:

```
ps -ef | grep username
```

Este exemplo exibe todos os processos com um nome de processo **bash**, o shell de login mais comum em sistemas Linux:

```
theo:> ps auxw | grep bash
brenda  31970  0.0  0.3  6080 1556 tty2    S   Feb23   0:00  -bash
root    32043  0.0  0.3  6112 1600 tty4    S   Feb23   0:00  -bash
theo    32581  0.0  0.3  6384 1864 pts/1   S   Feb23   0:00  bash
theo    32616  0.0  0.3  6396 1896 pts/2   S   Feb23   0:00  bash
theo    32629  0.0  0.3  6380 1856 pts/3   S   Feb23   0:00  bash
theo    2214   0.0  0.3  6412 1944 pts/5   S   16:18   0:02  bash
theo    4245   0.0  0.3  6392 1888 pts/7   S   17:26   0:00  bash
theo    5427   0.0  0.1  3720  548 pts/7   S   19:22   0:00  grep bash
```

Nesses casos, o comando **grep** que está procurando as linhas com `bash` também é exibido em sistemas com muito tempo ocioso. Se você não quer que isso aconteça, use o comando **pgrep**.

Shells `bash` são um caso especial: essa lista de processos também mostra quais são shells com identificação de usuário na entrada (onde você tem que fornecer nome de usuário e senha, como quando você inicia em modo texto ou numa conexão remota, o contrário de shells que, por exemplo, são iniciados ao clicar no ícone de uma janela de terminal). Tais shells com identificação na entrada do usuário são precedidos com um traço (-).

|?

Iremos tratar do operador `|` no próximo capítulo, veja [Chapter 5, Redirecionamento de E/S](#).

Mais informações podem ser obtidas da maneira usual: **ps --help** ou **man ps**. O GNU **ps** oferece suporte a diferentes tipos de formatos de opções; os exemplos acima não contêm erros.

Note que o comando **ps** apenas fornece o estado momentâneo dos processos, é uma lembrança única. O programa **top** exibe uma visão mais precisa ao atualizar os resultados fornecidos pelo comando **ps** (com um monte de opções) uma vez a cada cinco segundos, gerando periodicamente uma lista dos

processos que estão usando mais recursos.

```
12:40pm up 9 days, 6:00, 4 users, load average: 0.21, 0.11, 0.03
89 processes: 86 sleeping, 3 running, 0 zombie, 0 stopped
CPU states: 2.5% user, 1.7% system, 0.0% nice, 95.6% idle
Mem: 255120K av, 239412K used, 15708K free, 756K shrd, 22620K buff
Swap: 1050176K av, 76428K used, 973748K free, 82756K cached
```

PID	USER	PRI	NI	SIZE	RSS	SHARE	STAT	%CPU	%MEM	TIME	COMMAND
5005	root	14	0	91572	15M	11580	R	1.9	6.0	7:53	X
19599	jeff	14	0	1024	1024	796	R	1.1	0.4	0:01	top
19100	jeff	9	0	5288	4948	3888	R	0.5	1.9	0:24	gnome-terminal
19328	jeff	9	0	37884	36M	14724	S	0.5	14.8	1:30	mozilla-bin
1	root	8	0	516	472	464	S	0.0	0.1	0:06	init
2	root	9	0	0	0	0	SW	0.0	0.0	0:02	keventd
3	root	9	0	0	0	0	SW	0.0	0.0	0:00	kapm-idled
4	root	19	19	0	0	0	SWN	0.0	0.0	0:00	ksoftirqd_CPU0
5	root	9	0	0	0	0	SW	0.0	0.0	0:33	kswapd
6	root	9	0	0	0	0	SW	0.0	0.0	0:00	kreclaimd
7	root	9	0	0	0	0	SW	0.0	0.0	0:00	bdfush
8	root	9	0	0	0	0	SW	0.0	0.0	0:05	kupdated
9	root	-1	-20	0	0	0	SW<	0.0	0.0	0:00	mdrecoveryd
13	root	9	0	0	0	0	SW	0.0	0.0	0:01	kjournald
89	root	9	0	0	0	0	SW	0.0	0.0	0:00	khubd
219	root	9	0	0	0	0	SW	0.0	0.0	0:00	kjournald
220	root	9	0	0	0	0	SW	0.0	0.0	0:00	kjournald

A primeira linha do comando **top** contém a mesma informação exibida pelo comando **uptime**:

```
jeff:~> uptime
3:30pm, up 12 days, 23:29, 6 users, load average: 0.01, 0.02, 0.00
```

Os dados para estes programas são armazenados, entre outros, em `/var/run/utmp` (informações sobre usuários conectados no momento) e no sistema de arquivos virtual `/proc` - por exemplo `/proc/loadavg` (informação sobre carga média do sistema). Há todo o tipo de aplicativos gráficos para visualizar estes dados, tal como o Monitor de Sistema do Gnome e *lavaps*. No [FreshMeat](#) e no [SourceForge](#) você vai encontrar dezenas de aplicativos que centralizam esta informação juntamente com outros dados de servidores e logs de vários serviços em um único servidor (web), possibilitando o monitoramento de infraestrutura de TI inteira a partir de uma única estação de trabalho.

As relações entre os processos pode ser visualizada com o comando **pstree**:

```
sophie:~> pstree
init--+-amd
      |-apmd
      |-2*[artsd]
      |-atd
      |-crond
      |-deskguide_apple
      |-eth0
      |-gdm--+-gdm--+-X
            `--gnome-session--+-Gnome
                               |
                               |--ssh-agent
                               `--true
      |-geyes_applet
      |-gkb_applet
      |-gnome-name-serv
      |-gnome-smproxy
      |-gnome-terminal--+-bash---vim
                        |
                        |--bash
                        |--bash---pstree
                        |--bash---ssh
                        |--bash---mozilla-bin---mozilla-bin---3*[mozilla-bin]
                        `--gnome-pty-helper
      |-gpm
      |-gweather
      |-kapm-idled
      |-3*[kdeinit]
      |-keventd
      |-khubd
      |-5*[kjournald]
      |-klogd
      |-lockd---rpciod
      |-lpd
      |-mdrecoveryd
      |-6*[mingetty]
      |-8*[nfsd]
      |-nscd---nscd---5*[nscd]
      |-ntpd
      |-3*[oafd]
      |-panel
      |-portmap
      |-rhnsd
      |-rpc.mountd
      |-rpc.rquotad
      |-rpc.statd
```

```

|-sawfish
|-screenshoter_a
|-sendmail
|-sshd---sshd---bash---su---bash
|-syslogd
|-tasklist_applet
|-vmnet-bridge
|-xfs
`-xinetd-ipv6

```

As opções `-u` e `-a` dão informações adicionais. Para mais opções e o que elas fazem, consulte as páginas Info.

Na próxima seção, veremos como um processo pode criar um outro.

## Vida e morte de um processo

### Criação de processos

Um processo novo é criado quando um processo existente cria uma cópia de si mesmo. Este processo filho tem o mesmo ambiente que seu pai, apenas o número ID do processo é diferente. Este procedimento é chamado de *fork* (bifurcar).

Depois da bifurcação do processo, o espaço de endereçamento do processo filho é sobrescrito com os dados do novo processo. Isso é feito através da chamada de sistema `exec` (executar).

Assim o mecanismo *bifurcar-e-executar* troca um comando antigo por um novo, enquanto o ambiente no qual o programa estava executando continua o mesmo, incluindo configuração de dispositivos de entrada e saída, variáveis de ambiente e prioridade. Em UNIX esse mecanismo é usado para criar todos os processos, logo isto se aplica também ao sistema operacional Linux. Até mesmo o primeiro processo, com ID de processo 1, é bifurcado durante a inicialização do sistema, no famoso procedimento chamado *bootstrapping*.

Este esquema ilustra o mecanismo bifurcar-e-executar. O ID do processo muda depois do procedimento `fork`:

**Figure 4.1. Mecanismo bifurcar-e-executar**



Existem uma ou duas situações onde o **init** se torna o pai direto de um processo, sendo que o processo não foi iniciado diretamente pelo processo **init**, como já vimos no exemplo do comando **pstree**. Muitos programas, por exemplo, transformam seus filhos em *daemons*, de forma que eles possam continuar a executar quando o seu pai finaliza ou é finalizado. Um gerenciador de janelas é um exemplo típico; ele inicia um processo **xterm** que cria um shell que aceita comandos. Então o gerenciador de janelas nega qualquer responsabilidade futura e passa a guarda do processo filho para o **init**. Com esse mecanismo, é possível trocar de gerenciador de janelas sem interromper os aplicativos que foram iniciados por eles.

Veja ou outra as coisas vão mal, até nas melhores famílias. Num caso à parte, um processo pode finalizar enquanto o pai não esperou pelo término desse processo. Um processo sem enterro assim é chamado de processo *zumbi*.

### Finalizando processos

Quando um processo finaliza normalmente (não é morto e nem interrompido de maneira inesperada de alguma outra forma), o programa retorna seu *status de finalização* para seu processo pai. O status de finalização é um número que provê os resultados da execução do programa. O esquema de retornar informação sempre que executar uma tarefa tem sua origem na linguagem de programação C na a qual o UNIX tem sido escrito.

Os códigos de retorno podem então serem interpretados pelo pai, ou em scripts. Os valores dos códigos de retorno são específicos a cada programa. Geralmente essa informação pode ser encontrada nos manuais do programa em questão, por exemplo, o comando **grep** retorna `-1` se nenhum padrão coincidente for encontrado, baseada na qual a mensagem "No files found" pode ser impressa. Outro exemplo é o comando interno **true** do Bash, que não faz nada além de retornar `0`, que significa sucesso.

### Sinais

Processos finalizam porque eles recebem um sinal. São vários os sinais que podem ser enviados a um processo. Uso o comando **kill** (matar) para enviar um sinal para um processo. O comando **kill -l** exhibe uma lista de sinais. A maioria dos sinais são para uso interno do sistema, ou para programadores quando escrevem programas. Enquanto usuário, você precisará apenas da seguinte visão geral:

**Table 4.2. Sinais comuns**

Nome do sinal	Número do sinal	Significado
SIGTERM	15	Termina o processo de forma organizada.
SIGINT	2	Interrompe o processo. Um processo pode ignorar esse sinal.

SIGKILL	9	Interrompe o processo. Um processo não pode ignorar esse sinal.
SIGHUP	1	Para daemons: que releiam suas configurações.

Você pode ler mais sobre as ações padrões quando um sinal é enviado a um processo em **man 7signal**.

## SUID and SGID

Como prometido no capítulo anterior, iremos agora discutir os modos especiais SUID e SGID em mais detalhes. Estes modos existem para permitir que usuários normais executem tarefas que normalmente não poderiam devido ao rigoroso esquema de permissões usado em sistemas baseados em UNIX. O ideal é que os modos especiais sejam usados o mínimo possível, já que acrescentam riscos à segurança. Desenvolvedores Linux geralmente têm tentado evitá-los o quanto podem. A versão Linux do comando **ps**, por exemplo, usa a informação armazenada nos arquivos `/proc`, que são acessíveis a todos, mas evita expor recursos e informações sensíveis do sistema para o público geral. Antes disso, e ainda nos sistemas UNIX mais velhos, o programa **ps** precisava do acesso a arquivos como `/dev/mem` e `/dev/kmem`, o qual tinha desvantagens por causa das permissões e posses desses arquivos:

```
rita:~> ls -l /dev/*mem
crw-r----- 1 root kmem      1,  2 Aug 30 22:30 /dev/kmem
crw-r----- 1 root kmem      1,  1 Aug 30 22:30 /dev/mem
```

Com as versões mais antigas do comando **ps**, não é possível iniciar o programa como um usuário comum, a não ser que os modos especiais tenham sido aplicados.

Apesar de geralmente tentamos evitar aplicar qualquer modo especial, algumas vezes é necessário usar um SUID. Um exemplo é o mecanismo para torçar de senhas. É claro que os usuários vão querer fazer isso eles mesmos ao invés de ter suas senhas definidas pelo administrador do sistema. Como nós sabemos, nomes de usuários e senhas são listados no arquivo `/etc/passwd`, o qual tem estes donos e permissões:

```
bea:~> ls -l /etc/passwd
-rw-r--r-- 1 root root      1267 Jan 16 14:43 /etc/passwd
```

Os usuários precisam poder alterar suas próprias informações nesse arquivo. Isto é alcançado ao dar permissões especiais ao comando **passwd**:

```
mia:~> which passwd
passwd is /usr/bin/passwd
```

```
mia:~> ls -l /usr/bin/passwd
-r-s--x--x 1 root root      13476 Aug  7 06:03 /usr/bin/passwd*
```

Quando chamado, o comando **passwd** irá executar usando as permissões de acesso de `root`, e assim permitindo um usuário comum editar o arquivo de senhas, o qual pertence ao administrador do sistema.

Modos SGID não são usados com tanta frequência quanto o SUID, porque SGID geralmente envolve a criação de grupos extras. Em alguns casos, no entanto, nós temos que encarar esse problema no intuito de construir uma solução elegante (não se preocupe tanto com isso - os grupos necessários geralmente são criados durante a instalação). Esse é o caso dos comandos **write** e **wall**, que são usados para enviar mensagens aos terminais de outros usuários (ttys). O comando **write** escreve uma mensagem para um usuário individual, enquanto o **wall** escreve para todos os usuários conectados.

Enviar texto para o terminal ou dispositivo gráfico de outro usuário não é permitido. Para contornar esse problema, um grupo que possui todos os dispositivos de terminal foi criado. Quando são dadas permissões SGID aos comandos **write** e **wall**, os comandos executarão com os direitos de acessos aplicáveis para este grupo, `tty` no caso do exemplo. Já que o grupo tem direito de escrita para o terminal de destino, um usuário sem permissões para usar o terminal também pode enviar mensagens a ele.

No exemplo abaixo, o usuário `joe` primeiro descobre qual terminal a sua destinatária está conectada, usando o comando **who**. Então ele a envia uma mensagem usando o comando **write**. Estão ilustrados também os direitos de acesso do programa **write** e dos terminais ocupados pelo usuário receptor: está claro que outros além do usuário não tem acesso ao dispositivo, exceto para o grupo, que pode escrever nele.

```
joe:~> which write
write is /usr/bin/write
```

```
joe:~> ls -l /usr/bin/write
-rwxr-sr-x 1 root tty      8744 Dec  5 00:55 /usr/bin/write*
```

```
joe:~> who
jenny  tty1    Jan 23 11:41
jenny  pts/1    Jan 23 12:21 (:0)
jenny  pts/2    Jan 23 12:22 (:0)
jenny  pts/3    Jan 23 12:22 (:0)
joe    pts/0    Jan 20 10:13 (lo.callhost.org)
```

```
joe:~> ls -l /dev/tty1
crw--w---- 1 jenny tty  4,    1 Jan 23 11:41 /dev/tty1
```

```
joe:~> write jenny tty1
hey Jenny, que tal almoçarmos juntos?
^C
```

Usuária *jenny* recebe em sua tela:

```
Mensagem de joe@lo.callhost.org on ptys/1 at 12:36 ...
hey Jenny, que tal almoçarmos juntos?
EOF
```

Após receber a mensagem, o terminal pode ser limpo com a combinação de teclas **Ctrl+L**. Para não receber nenhuma mensagem (exceto do administrador do sistema), use o comando **mesg**. Para ver quais usuários conectados aceitam mensagens de outros, use o comando **who -w**. Todos os recursos são explicados por completo nas páginas Info de cada comando.

## Nomes de grupos podem variar

O esquema de grupos é próprio da distribuição sendo usada. Outras distribuições podem usar outros nomes ou outras soluções.

# Processo de inicialização, iniciar e desligar

## Introdução

Um dos aspectos mais poderosos do Linux é a respeito dos seus métodos abertos de inicialização e desligamento do sistema operacional, onde são carregados programas específicos com suas configurações particulares, que permitem mudar estas configurações para controlar o processo de inicialização, e também desligar de uma forma organizada e graciosa.

Além da questão de controlar o processo de inicialização ou desligamento, a natureza aberta do Linux torna bem mais fácil determinar a fonte exata de muitos problemas associados com a inicialização ou desligamento do seu sistema. Um entendimento básico desse processo é bastante benéfico para todos que usam um sistema Linux.

Muitos sistemas Linux usam o **lilo**, o LIinux LOader para inicializar sistemas operacionais. Porém, iremos discutir apenas o GRUB, que é mais flexível e fácil de usar. Caso necessite de informações sobre o **lilo**, consulte as páginas de manuais e guias. Ambos sistemas suportam instalações duplas de sistemas operacionais na mesma máquina. Nesse assunto, confiamos os exemplos práticos e informações complementares à outros guias.

## O processo de inicialização

Quando um computador x86 é inicializado, o processador busca pelo BIOS no final da memória do sistema (Basic Input/Output System - Sistema Básico de Entrada e Saída) e o executa. O programa BIOS é escrito em memória permanente de apenas leitura e está sempre disponível para uso. O BIOS provê a interface de menor nível para os dispositivos periféricos e controla o primeiro passo no processo de inicialização.

O BIOS testa o sistema, procura e verifica periféricos, e então procura por um drive para usar na inicialização do sistema. Geralmente verifica drive de disquete (ou drive de CD-ROM em muitos sistemas mais novos) em busca de mídia inicializável, quando presentes, e então verifica o disco rígido. A ordem dos drives usados para inicialização é geralmente controlada por uma configuração particular no BIOS do sistema. Uma vez que o Linux estiver instalado no disco rígido de um sistema, o BIOS procura um MBR (Master Boot Record - registro mestre de inicialização) iniciando no primeiro setor do primeiro disco rígido, carrega seu conteúdo na memória, e então passa o controle para ele.

Esse MBR contém instruções sobre como iniciar o carregador de inicialização GRUB (ou LILO), com um sistema operacional pré-selecionado. O MBR inicia o carregador de inicialização, o qual assume o processo (se o carregador de inicialização está instalado no MBR). Em uma configuração Linux Red Hat padrão, o GRUB utiliza as configurações contidas na MBR para exibir opções de inicialização em um menu. Uma vez que o GRUB tenha recebido instruções corretas para iniciar o sistema operacional, seja por linha de comando ou arquivo de configuração, ele encontra o arquivo de inicialização necessário e entrega o controle da máquina para o sistema operacional.

## Recursos do GRUB

Esse método de inicialização é chamado *carregamento direto* porque as instruções são usadas para carregar o sistema operacional diretamente, sem nenhum código intermediário entre os carregadores de inicialização e os arquivos principais do sistema operacional (como o kernel). Porém, o processo de inicialização de outros sistemas operacionais podem diferir ligeiramente do descrito acima. Por exemplo, os sistemas operacionais DOS e Windows, da Microsoft, sobrescrevem completamente qualquer coisa na MBR quando são instalados, sem considerar a configuração presente na MBR no momento. Isso destrói qualquer outra informação armazenada na MBR por outros sistemas operacionais, como Linux. Os sistemas operacionais da Microsoft, assim como vários outros sistemas operacionais proprietários, são carregados usando um método de carregamento de inicialização em corrente. Com esse método, o MBR aponta para o primeiro setor da partição que contém o sistema operacional, onde encontra os arquivos especiais necessários para realmente inicializar aquele sistema operacional

O GRUB funciona com os dois métodos, permitindo você usá-lo com quase qualquer sistema operacional, com grande parte dos sistemas de arquivos mais importantes, e com quase qualquer disco rígido que seu BIOS possa reconhecer.

O GRUB tem vários outros recursos; os mais importantes incluem:

- O GRUB oferece um verdadeiro ambiente de comando pré sistema operacional em máquinas x86 que permite máxima flexibilidade ao carregar sistemas operacionais com certas opções, ou reunir informações sobre o sistema.
- GRUB aceita modo LBA (Logical Block Addressing - endereçamento lógico de blocos) (LBA), necessário para acessar alguns discos IDE e quaisquer discos SCSI. Antes do LBA, discos rígidos tinham um limite de 1024 cilindros, no qual o BIOS não seria capaz de encontrar um arquivo após esse limite.
- O arquivo de configuração do GRUB é lido do disco toda vez que o sistema inicializa, fazendo com que você não precise escrever no MBR toda vez que você muda as opções de BOOT.

Uma descrição completa do GRUB pode ser acessada pelo comando **info grub** ou no [site do GRUB](#). O Projeto de Documentação do Linux tem um [mini-guia de inicialização com GRUB em máquinas com mais de um sistema operacional instalado](#).

## Init

O kernel, uma vez carregado, encontra o programa **init** no diretório `sbin` e o executa.

Quando o **init** inicia, ele se torna o pai ou avô de todos os processos que iniciam automaticamente no seu sistema Linux. A primeira coisa que o **init** faz é ler seu arquivo de inicialização, `/etc/inittab`. Isso instrui o **init** a ler um script de configuração inicial do ambiente, que configura os caminhos de busca, inicia a memória virtual, verifica os sistemas de arquivos, e por aí vai. Basicamente, esse passo cuida de tudo que seu sistema precisa que seja feito durante a sua inicialização: configura o relógio, inicializa portas seriais e adiante.

Quando o **init** segue lendo o arquivo `/etc/inittab`, que descreve como o sistema deve ser configurado em cada nível de execução qual o *nível de execução* padrão. Um nível de execução é uma configuração de processo. Todos os sistemas do tipo UNIX podem operar em diferentes configurações de processo, tais como o modo com um único usuário, o qual é chamado nível de execução 1 ou nível de execução S (ou s). Nesse modo, apenas o administrador do sistema pode acessar o sistema. É usado para realizar tarefas de manutenção, sem o risco de danificar dados do sistema ou de usuários. Naturalmente, nessa configuração nós não precisamos oferecer serviços aos usuários, então, eles são todos desabilitados. Outro nível de execução é o de reinicialização, ou nível de execução 6, que desliga todos os serviços em execução de acordo com os procedimentos apropriados e então reinicia o sistema.

Use o comando **who** para verificar qual o seu nível de execução atual:

```
willy@ubuntu:~$ who -r
run-level 2 2006-10-17 23:22          last=S
```

Mais a respeito de níveis de execução na próxima seção, veja [the section called "Níveis de execução do init"](#).

Depois de descobrir o nível de execução padrão para o seu sistema, o **init** inicia todos os processos de plano de fundo que são necessário para que o sistema execute, que estão listados no diretório `rc` daquele nível de execução. O **init** executa cada um dos scripts de kill (cujos nomes são iniciados por um K) com um parâmetro de parada. Então, ele executa todos os scripts de start (cujos nomes são iniciados por S) no diretório de nível de execução apropriado, de forma que todos os serviços e aplicativos são carregados corretamente. Na verdade, é possível executar estes scripts manualmente depois que o sistema terminar a inicialização com um comando como o `/etc/init.d/httpd stop` ou o **service httpdstop**, se estiver usando o `root`, neste caso parando o servidor web.

## Caso especial

Perceba que na inicialização do sistema, os scripts no `rc2.d` e no `rc3.d` são normalmente executados. Neste caso, nenhum serviço é parado (ao menos não permanentemente). Há apenas serviços que são iniciados.

Nenhum dos scripts que de fato são inicializados e param os serviços estão localizados no `/etc/rc<x>.d`. Na verdade, todos os arquivos que estão no `/etc/rc<x>.d` são links simbólicos para os scripts de fato que estão armazenados no `/etc/init.d`. Um link simbólico nada mais é do que um arquivo que aponta para outro arquivo e que é usado neste caso porque ele pode ser criado e aparado sem afetar os scripts que matam ou iniciam serviços. Os links simbólicos para os vários scripts estão numerados de uma tal forma que eles são executados naquela ordem. Você pode alterar a ordem em que os serviços são inicializados ou mortos alterado o nome do link simbólico que remete ao script que de fato controla o serviço. Você pode usar o mesmo número várias vezes se você quiser que um serviço em particular seja iniciado ou parado logo antes ou logo depois de um outro, como no exemplo abaixo, listando o conteúdo do `/etc/rc5.d`, em que o **crond** e o **xfs**, ambos iniciados a partir de um link cuja denominação é iniciada por "S90". Neste caso, os scripts são inicializados em ordem alfabética.

```
[jean@blub /etc/rc5.d] ls
K15httpd@      K45named@      S08ipchains@  S25netfs@      S85gpm@
```



K16rarpd@	K46radvd@	S08iptables@	S26apmd@	S90crond@
K20nfs@	K61ldap@	S09isdn@	S28autofs@	S90xfs@
K20rstatd@	K65identd@	S10network@	S30nscd@	S95anacron@
K20rusersd@	K74ntpd@	S12syslog@	S55sshd@	S95atd@
K20rwall@	K74ypserv@	S13portmap@	S56rawdevices@	S97rhnsd@
K20rwhod@	K74ypxfrd@	S14nfslock@	S56xinetd@	S99local@
K25squid@	K89bcm5820@	S17keytable@	S60lpd@	
K34yppasswdd@	S05kudzu@	S20random@	S80sendmail@	

Depois que o **init** tiver percorrido todos os níveis de execução até o padrão, o script `/etc/inittab` cria um subprocesso **getty** para cada console virtual (prompt de login no modo texto). O **getty** abre linhas tty, configura seus modos, imprime um prompt de login, obtém o nome do usuário e então inicia o processo de login para aquele usuário. Isto permite que os usuários se autenticuem no sistema e usem-no. Por padrão, a maior parte dos sistemas oferece 6 consoles virtuais, mas como você pode perceber pelo arquivo `inittab`, isto é configurável.

O `/etc/inittab` também pode dizer ao **init** como ele deveria reagir ao usuário que pressiona **Ctrl+Alt+Delete** no console. Se o sistema deveria ser devidamente desligado e reiniciado ao invés de apenas sofrer um ciclo de energia, o **init** é configurado para executar o comando `/sbin/shutdown -t3 -rnow`, por exemplo, quando um usuário pressionar tais teclas. Além disso, o `/etc/inittab` é quem diz o que o **init** deve fazer em caso de quedas de energia, se o seu sistema tiver uma unidade de UPS ligada a ele.

Na maior parte dos sistemas baseados em RPM, a tela de login gráfico é iniciada no nível de execução 5, na qual o `/etc/inittab` executa um script chamado `/etc/X11/prefdm`. O script `prefdm` executa o gerenciador de tela X, baseado no conteúdo do diretório `/etc/sysconfig/desktop`. Normalmente o **gdm** se você usa o GNOME ou o **kdm** se usa o KDE, mas eles podem ser misturados. Existe ainda o **xdm** que vem com a instalação X padrão.

Mas existem outras possibilidades também. No Debian, por exemplo, existe um initscript para cada um dos gerenciadores de tela e o conteúdo do `/etc/X11/default-display-manager` é usado para determinar qual deve ser iniciado. Pode-se aprender mais sobre a interface gráfica na seção [the section called "O ambiente gráfico"](#). Por fim, a documentação do seu sistema irá explicar os detalhes sobre os aspectos de alto nível do **init**.

O diretório `/etc/default` e/ou o `/etc/sysconfig` contém informações sobre uma série de funções e serviços, as quais são lidas quando da inicialização do sistema. A localização do diretório que contém os padrões do sistema pode ser um pouco diferente dependendo da sua distribuição Linux.

Além do ambiente de usuário gráfico, muitos outros serviços talvez sejam inicializados. Porém, se tudo estiver certo, ser-lhe-á apresentado o prompt de login ou a tela de login quando o processo de inicialização tiver terminado.

## Outros procedimentos

Nós explicamos como o SysV**init** funciona em máquinas x86. Os processos de inicialização podem variar em outras arquiteturas e distribuições. Outras distribuições podem usar o **init** no estilo BSD, no qual os arquivos de inicialização não estão espalhados em diferentes `/etc/rc<LEVEL>.d` diretórios. É possível também que o seu sistema use o `/etc/rc.d/init.d` ao invés do `/etc/init.d`.

## Níveis de execução do init

A idéia por trás da operação de diferentes serviços em diferentes níveis de execução gira em torno do fato de que sistemas diferentes podem ser usados de formas diferentes. Alguns serviços podem não ser utilizados até que o sistema esteja em um estado, ou *modo*, em particular, como, por exemplo, estar pronto para mais de um usuário ou possuir mais de uma rede disponível.

Pode ser que em algum momento você queira operar o sistema em um nível inferior. Para corrigir problemas de corrupção de disco no nível 1, por exemplo, de modo que nenhum outro usuário possa estar no sistema, ou então deixar um sistema executando no nível 3 sem uma sessão X. Nestes casos, os serviços em execução que dependam de um modo de sistema de mais alto nível não fazem sentido, vez que, de qualquer maneira, não irão funcionar corretamente. Se cada serviço for iniciado quando seu nível de execução for atingido, garante-se um processo de inicialização ordenado e permite-se a mudança rápida de modo, sem a necessidade de se preocupar com quais serviços inicializar ou parar.

Os níveis de execução disponíveis estão descritos de maneira genérica no `/etc/inittab`, que se exhibe abaixo:

```
# inittab Este arquivo descreve como o processo INIT deve configurar
# o sistema em um dado nível de execução.

# Nível de execução padrão. Os níveis são:
# 0 - desligado (NÃO configure o initdefault para este nível)
# 1 - Modo usuário único
# 2 - Multiusuários, sem NFS
# (O mesmo que o 3, se não houver rede)
# 3 - Modo multiusuário completo
# 4 - sem uso
# 5 - X11
# 6 - reinicialização (NÃO configure o initdefault para este nível)
#
```

```
id:5:initdefault:
<--cut-->
```

Sinta-se livre para configurar níveis de execução não utilizados (normalmente o nível 4), se entender necessário. Muitos usuários configuram estes níveis de execução do modo que mais lhes apraz, deixando os níveis padrão intocados. Isto permite que mudem rapidamente sua configuração personalizada, sem impor quaisquer alterações ao conjunto de recursos dos níveis de execução padrão.

Se a sua máquina ficar em um estado em que ela não inicia por conta de um `/etc/inittab` ruim ou não permite o login porque o seu arquivo `/etc/passwd` foi corrompido (ou se você tiver simplesmente esquecido a senha), efetue o login no modo de usuário único.

## Sem gráficos?

Quando você estiver trabalhando no modo texto porque não lhe foi apresentada a tela de login gráfico no console de sua máquina, é possível mudar para o console 7 ou acima para obter um login gráfico. Se este não for o caso, verifique o nível atual usando o comando **who -r**. Se estiver configurado para algo diferente do que o padrão original do `/etc/inittab`, é possível que o sistema não inicialize no modo gráfico por padrão. Neste caso, entre em contato com o seu administrador ou leia **man init**. Perceba que a alteração dos níveis de execução é feita preferencialmente pelo comando **telinit**; mudar de um console texto para um gráfico ou vice e versa não envolve uma mudança de nível de execução.

A discussão dos níveis de execução, scripts e configurações neste guia tenta ser o mais genérica possível. Muitas variações existem. Por exemplo, o Gentoo Linux armazena scripts no `/etc/run levels`. Outros sistemas podem inicialmente ser executados através de um nível de execução inferior e rodar todos os scripts lá contidos antes de chegar no nível de execução final e rodar aqueles scripts. Procure na documentação do seu sistema para saber mais. Você pode também ler os scripts que são mencionados no `/etc/inittab` para ter uma compreensão melhor do que acontece no seu sistema.

## Ferramentas

Os utilitários **chkconfig** ou **update-rc.d**, quando instalados no seu sistema, proporcionam uma ferramenta de linha de comando simples para manter a hierarquia de diretórios `/etc/init.d`. Estes evita que os administradores de sistema tenham que manipular diretamente diversos links simbólicos nos diretórios `/etc/rc[x].d`.

Mais ainda, alguns sistemas oferecem a ferramenta **ntsysv**, que fornece uma interface modo texto; você pode achá-la mais fácil de usar do que a interface de linha de comando do **chkconfig**. No SuSE Linux, você encontra o **yast** e o **insserv**. Para configurar Mandrake, você pode querer tentar o DrakConf, que permite, dentre outros recursos, alternar entre os níveis 3 e 5. No Mandriva, ele se tornou o Mandriva Linux Control Center.

A maior parte das distribuições fornece uma interface gráfica para a configuração de processos, verifique a documentação de seu sistema.

Todos esses utilitários precisam ser executados como o root. O administrador do sistema também pode criar manualmente os links apropriados em casa diretório de nível de execução de forma a iniciar ou parar um serviço em dado nível.

## Desligamento

O UNIX não foi feito para ser desligado, mas se você de fato precisar, use o comando **shutdown**. Após completar o procedimento de desligamento, a opção `-h` irá parar o sistema, enquanto a opção `-r` irá reiniciá-lo.

Os comandos **reboot** e **halt** podem agora invocar o **shutdown**, se forem executados quando o sistema estiver nos níveis de execução 1-5 e, assim, garantir o desligamento seguro do sistema. Porém é um hábito ruim e nem todas as versões do UNIX/Linux possuem este recurso.

Se o seu computador não se desligar, você não deve desligá-lo até que veja uma mensagem indicando que o sistema parou ou terminou de ser desligado, para permitir que o sistema desmonte todas as partições. A impaciência pode implicar na perda de dados.

## Gerenciando processos

### Trabalho para o administrador do sistema

Embora gerenciar os recursos do sistema, incluindo os processos, seja uma tarefa para o administrador do sistema, não dói aprender um pouco sobre isso, em especial quando está em jogo a execução ótima de seus processos.

Nós explicaremos um pouco em um nível teórico sobre a performance do sistema, não tão profundo a ponto de discutir otimização de hardware e outros procedimentos avançados. Ao invés disso, nós estudaremos problemas quotidianos na vida de um usuário, bem como as medidas que podem ser adotadas para usar os recursos disponíveis de maneira ótima. Como aprenderemos na seção seguinte, isto é principalmente uma questão de pensar antes de agir.

## Figure 4.2. Você não pode ir mais rápido?

### Quanto tempo leva?

O bash possui o comando interno **time** que exibe o tempo que um comando leva para ser executado. A contagem é muito precisa e pode ser usada em qualquer comando. No exemplo abaixo, este livro leva cerca de um minuto e meio para ser feito:

```
tilly:~/xml/src> time make
Output written on abook.pdf (222 pages, 1619861 bytes).
Transcript written on abook.log.

real    1m41.056s
user    1m31.190s
sys     0m1.880s
```

O GNU **time** NO `/usr/bin` (distinto do comando interno) exibe mais informações que podem ser formatadas de diferentes maneiras. Ele também exibe o status de saída do comando e o total do tempo transcorrido. O mesmo comando acima usando o **time** independente fornece a seguinte saída:

```
tilly:~/xml/src> /usr/bin/time make
Output written on abook.pdf (222 pages, 1595027 bytes).
Transcript written on abook.log.

Command exited with non-zero status 2
88.87user 1.74system 1:36.21elapsed 94%CPU
      (0avgtext+0avgdata 0maxresident)k
0inputs+0outputs (2192major+30002minor)pagefaults 0swaps
```

Consulte novamente as páginas Info para acessar toda a informação.

### Performance

Para um usuário, performance é sinônimo de execução rápida de comandos. Para um gerente de sistemas, por outro lado, significa muito mais: o administrador deve otimizar a performance do sistema para todo ele, incluindo os usuários, programas e daemons. A performance do sistema pode depender de milhares de pequenas coisas que não são levadas em conta pelo comando **time**:

- o programa em execução foi mal escrito ou não usa o computador de maneira apropriada
- acesso a discos, controles, tela e todos os tipos de interfaces, etc.
- capacidade de alcançar sistemas remotos (performance de rede)
- quantidade de usuários no sistema, quantidade de usuários trabalhando simultaneamente
- hora do dia
- ...

### Carga

Basicamente: a carga depende do que é normal para o seu sistema. Meu velho P133 executando um firewall, um servidor SSH, um servidor de arquivos, um daemon de rotas, um servidor de emails, um servidor de proxy e alguns outros serviços não reclama com 7 usuários conectados; a carga permanece em média 0. Alguns sistemas (multi-CPU) que encontrei ficavam bastante felizes com uma carga de 67. Existe apenas uma forma de descobrir: verifique a carga rotineiramente se você quiser descobrir o que é normal. Se você não o fizer, somente será capaz de medir a carga do seu sistema pelo tempo de resposta da linha de comando, o que é uma medida muito rústica, vez que esta medida é influenciada por uma centena de outros fatores.

Tenha em mente que sistemas diferentes se comportam de maneiras diferentes com a mesma média de carga. Por exemplo, um sistemas com um cartão de gráfico que suporta aceleração de hardware não terá quaisquer dificuldades em lidar com imagens 3D, enquanto o mesmo sistema com um cartão VGA barato ficara muito lento enquanto estiver lidando com tais imagens. Meu velho P133 fica muito desconfortável com o servidor X, mas em um sistema moderno quase não se nota a diferença na carga do sistema.

### Eu posso fazer alguma coisa como um usuário?

Um ambiente grande pode reduzir sua velocidade. Se você tiver várias variáveis de ambiente configuradas (ao invés de variáveis do shell), caminhos de busca longos que não são otimizados (erros na configuração da variável de ambiente PATH) e mais dessas configurações do que é feito “durante a execução”, o sistema irá precisar de mais tempo para pesquisar e ler dados.

No X, gerentes de janelas e ambientes de desktop podem ser verdadeiros devoradores de CPU. Um desktop cheio de estilo tem seu preço, mesmo quando baixado gratuitamente, vez que a maior parte dos desktops fornece add-ons ao infinito. A modéstia é uma virtude se você não compra um pc novo a cada ano.

## Prioridade

A prioridade ou importância de um trabalho é definida pelo seu número *nice*. Um programa com um número *nice* é amigável com outros programas, outros usuários e com o sistema; não é um trabalho importante. Quanto menor o número de *nice*, mais importante o trabalho e mais recursos ele irá tomar sem compartilhá-los.

Tornar um trabalho mais amigável, aumentando o seu número *nice* somente é útil para processos que usam muito tempo da CPU (compiladores, aplicativos de matemática e processos do gênero). Processos que sempre usam muito tempo de E/S são automaticamente premiados pelo sistema e recebem uma prioridade maior (um número *nice* menor), por exemplo, a entrada do teclado sempre tem a maior prioridade em um sistema.

A definição da prioridade de um programa é feita com o comando **nice**.

A maior parte dos sistemas também fornece o comando **renice** do BSD, que permite que você altere o o *nice* de um comando em execução. Novamente, leia as páginas man de seu sistema para saber mais.

## Programas interativos

NÃO é uma boa idéia dar um **nice** ou **renice** em um programa interativo ou em uma tarefa em execução no primeiro plano.

O uso destes comandos é normalmente uma tarefa para administradores de sistemas. Leia as páginas man para saber mais sobre as funções disponíveis para o administrador.

## Recursos de CPU

Em cada sistema Linux, muitos programas querem usar o CPU ao mesmo tempo, mesmo quando você é o único usuário no sistema. Todos os programas precisam de uma dada quantidade de ciclos do CPU para serem executados. Às vezes pode não haver ciclos o suficiente porque o CPU está muito ocupado. O comando **uptime** é pouquíssimo preciso (exibe apenas médias, é preciso saber o que é o normal), mas está longe de ser inútil. Existem algumas ações que você pode tomar se acha que é culpa do seu CPU a falta de resposta do seu sistema:

- Execute programas pesados quando a carga do sistema estiver baixa. Este pode ser o caso do seu sistema ao longo da noite. Veja a seção seguinte sobre agendamento.
- Evite que o sistema realize trabalho desnecessário: pare os daemons e programas que você não usa, use o comando **locate** ao invés do pesado **find**, ...
- Execute grandes tarefas com baixa prioridade

Se nenhuma destas soluções for uma opção em uma situação em particular, talvez seja hora de comprar um CPU melhor. Em uma máquina UNIX este é um trabalho do administrador de sistemas.

## Recursos de memória

Quando os processos em execução esperam mais do que aquilo que o sistema possui fisicamente disponível, um sistema Linux não irá travar; ele irá começar a paginar, ou *swapping*, ou seja, os processos usam a memória em disco ou no espaço de swap, movendo o conteúdo da memória física (pedaços de programas em execução ou programas inteiros no caso de *swapping*) para o disco, recuperando, assim, a memória física para a manipulação de mais processos. Isto desacelera muito o sistema, vez que o acesso ao disco é muito mais lento do que o acesso à memória. O comando **top** pode ser usado para exibir a memória e a swap em uso. Sistemas usando glibc oferecem os comandos **memusage** e **memusagestat** para a visualização do uso de memória.

Se você descobrir que muita memória e swap estão em uso, você pode tentar:

- Matar, parar ou mudar o *nice* dos programas que usam uma parcela muito grande da memória.
- Adicionar mais memória (e, em alguns casos, mais espaço de swap) ao seu sistema.
- Aperfeiçoar a performance do sistema, o que está além do escopo deste documento. Veja a lista de leitura no [Appendix A, Para aonde ir a partir daqui?](#) para saber mais.

## Recursos de Entrada/Saída

Embora as limitações de E/S sejam uma das principais causas de stress para os administradores de sistemas, o sistema Linux oferece utilitários um tanto quanto pobres para mensurar a performance de E/S. As ferramentas **ps**, **vmstat** e **top** podem dar alguns indícios de quantos programas estão esperando por E/S; o **netstat** exibe as estatísticas de rede, mas praticamente inexitem ferramentas para mensurar a resposta de E/S para a carga no sistema e o comando **iostat** fornece um breve panorama do uso de E/S em geral. Diversas interfaces gráficas existem para colocar a saída destes comandos em um formato inteligível por um humano.

Cada dispositivo tem seus próprios problemas, mas a banda disponível para as interfaces de rede e a banda disponível para os discos são duas das causas primárias de afunilamento de performance de E/S.

Problemas de E/S em rede:

- Sobrecarga de rede:

A quantidade de dados transportada pela rede é maior que a sua capacidade, do resulta uma execução lenta de todas as tarefas relacionadas à rede, para todos os usuários. Eles podem ser resolvidos limpando a rede (o que basicamente envolve a desabilitação de protocolos ou serviços que você não utiliza) ou pela reconfiguração da rede (por exemplo, uso de subredes, a troca de concentradores por comutadores, a atualização de interfaces e equipamentos).

- Problemas de integridade de rede:

Ocorrem quando os dados são transmitidos de forma incorreta. A solução deste tipo de problema somente pode ser realizada isolando o elemento falho e trocando-o.

Problemas de Entrada/Saída de disco:

- taxa de transferência por processo baixa:

Velocidade de leitura ou escrita para um único processo não é suficiente.

- taxa de transferência agregada muito baixa:

O máximo de banda total que o sistema consegue fornecer para todos os programas em execução não é o suficiente.

Este tipo de problema é de detecção mais difícil e normalmente requer hardware adicional de forma a redividir os fluxos de dados pelos barramentos, controles e discos, se um hardware sobrecarregado for a causa do problema. Uma solução para isto é a configuração de array em RAID, otimizada para ações de entrada e saída. Desta forma, é possível manter o mesmo hardware. Uma atualização para barramentos, controles e discos mais rápidos é normalmente a outra opção.

Se sobrecarga não for a causa, talvez o seu hardware esteja gradualmente falhando ou mal conectado ao sistema. Antes de mais nada, verifique os contatos, conectores e plugs.

## Usuários

Usuários podem ser divididos em diversas categorias, dependendo do seu comportamento com e uso dos recursos:

- Usuários que executam um grande número de pequenas tarefas, você o usuário Linux iniciante, por exemplo.
- Usuários que executam relativamente poucos porém grandes trabalhos: usuários executando simulações, cálculos, emuladores ou outros programas que comem muita memória e normalmente estes usuários possuem também grandes arquivos de dados.
- Usuários que executam poucos trabalhos, mas usam muito tempo de CPU (desenvolvedores e seus congêneres).

Você pode ver que os requisitos do sistema podem variar para cada classe de usuários e que pode ser difícil satisfazer a todos. Se você estiver em um sistema multi-usuários, é útil (e divertido) descobrir os hábitos de outros usuários e o sistema, de forma a obter o máximo dele para os seus propósitos.

## Ferramentas gráficas

Para o ambiente gráfico, existe um punhado de ferramentas de monitoramento disponíveis. Abaixo, está uma captura de tela do Gnome System Monitor, que tem recursos para a exibição e busca de informação sobre processos e monitoramento dos recursos do sistema:

**Figure 4.3. Gnome System Monitor**



Existem também alguns ícones muito úteis que se pode instalar na barra de tarefas, como um monitor de disco, memória e carga. **xload** é outro pequeno aplicativo X para monitorar a carga no sistema. Descubra o seu favorito!

## Interrompendo processos

Como um usuário não privilegiado, você pode influenciar seus próprios processos. Nós já vimos como se pode exibir processos e filtrá-lo de outros que pertençam a um usuário em particular, bem com quais restrições podem ocorrer. Quando você perceber que um dos seus processos está comendo muito dos recursos, do sistema, há duas coisas que você pode fazer:

1. Fazer com que eles usem menos recursos sem interrompê-los;
2. Pará-los de uma vez.

Caso você queira que o processo continue em execução, mas também queira dar aos outros processos no sistema uma chance, você pode dar um **renice** nele. Além de usar os comandos **nice** ou **renice**, o **top** é uma ferramenta útil para captar processos problemáticos e de reduzir sua prioridade.

Identifique o processo na coluna "NI", ele provavelmente terá uma prioridade negativa. Digite **r** e informe a ID do processo do qual você quer mudar o nice. Então, informe um valor de nice, por exemplo, "20". Isto significa que, daqui em diante, este processo irá usar, no máximo, 1/5 dos ciclos da CPU.

Exemplos de processos que é desejável manter em execução são os emuladores, máquinas virtuais, compiladores e assim por diante.

Se você quer parar um processo porque ele congela ou fica completamente descontrolado no consumo de E/S, criação de arquivos ou uso de outros recursos do sistema, use o comando **kill**. Se você tiver a oportunidade, tente primeiro matar o processo gentilmente, enviando o sinal **SIGTERM**. Esta é uma instrução para terminar o que estiver sendo feito, de acordo com os processos descritos no código do programa:

```
joe:~> ps -ef | grep mozilla
joe  25822      1  0 Mar11 ?        00:34:04 /usr/lib/mozilla-1.4.1/mozilla-

joe:~> kill -15 25822
```

No exemplo acima, o usuário *joe* parou o seu navegador Mozilla porque ele congelou.

Alguns processos são um pouco mais difíceis de se livrar. Se você tiver tempo, pode talvez querer enviar a eles o sinal **SIGINT** para interrompê-los. Se isto não resolver, use o final mais forte, **SIGKILL**. No exemplo abaixo, o *joe* pára um Mozilla mozilla que está travado:

```
joe:~> ps -ef | grep mozilla
joe  25915      1  0 Mar11 ?        00:15:06 /usr/lib/mozilla-1.4.1/mozilla-

joe:~> kill -9 25915
```

```
joe:~> ps -ef | grep 25915
joe  2634 32273  0 18:09 pts/4    00:00:00 grep 25915
```

Nestes casos, você pode querer verificar se o processo de fato morreu, usando o filtro **grep** novamente no PID. Se isto retornar apenas o processo **grep**, você pode ter certeza de que obteve sucesso.

Dentre os processos que são difíceis de matar está o seu shell. E isto é uma boa coisa: Se ele fosse fácil de matar, você perderia o seu shell sempre que acidentalmente digitasse **Ctrl-C** na linha de comando, vez que isto equivale a enviar um **SIGINT**.

## UNIX sem pipes é quase impensável

O uso de pipes (|) para redirecionar a saída de um comando para a entrada de outro será explicada no próximo capítulo, [Chapter 5, Redirecionamento de E/S](#).

Em um ambiente gráfico, o programa **xkill** é muito simples de usar. Digite apenas o nome do comando, seguido de um **Enter** e selecione a janela do aplicativo que você deseja terminar. Ele é bem perigoso, eis que, por padrão envia um **SIGKILL**, então, use-o apenas quando um aplicativo congelar.

## Agendando processos

### Use aquele tempo ocioso!

Um sistema Linux pode ter muito do que sofrer, mas ele normalmente sofre apenas durante o horário comercial. Seja em um ambiente de escritório, em uma sala de servidores ou em casa, a maior parte dos sistemas Linux fica sem uso durante as manhãs, noites, madrugadas e finais de semana. Usar este tempo de folga pode ser muito mais econômico do que comprar uma daquelas máquinas que seriam indispensáveis caso você quisesse fazer tudo ao mesmo tempo.

Existem três tipos de execução postergada:

- Esperar um certo tempo e então resumir a execução de um trabalho usando o comando **sleep**. O tempo de execução depende do tempo do sistema no momento da submissão.
- Executar um comando em um tempo específico, usando o comando **at**. A execução de trabalhos depende do tempo do sistema, não do tempo da submissão.
- Regularmente executar um comando em uma base mensal, semanal, diária ou horária, usando os recursos do **cron**.

As seções a seguir discutem cada uma dessas possibilidades.

### O comando sleep

A página Info do **sleep** é provavelmente uma das mais curtas que existem. Tudo o que o **sleep** faz é esperar. Por padrão, o tempo de espera é anunciado em segundos.

Então por que ele existe? Alguns exemplos práticos :

Alguém liga para você no telefone, você diz "Sim, eu estarei com você em meia hora", mas você está afundado no trabalho e propenso a esquecer o seu almoço:

```
(sleep 1800; echo "Lunch time.") &
```

Quando você não pode usar o comando **at** por alguma razão, é cinco da tarde e você quer ir para casa, mas ainda há trabalho a ser feito e agora mesmo alguém está comendo os recursos do sistema:

```
(sleep 10000; myprogram) &
```

Certifique-se de que existe um logout automático no seu sistema e que você efetuou o logout ou travou o monitor do seu escritório quando submeter este tipo de trabalho ou executá-lo em uma sessão **screen**.

Quando você imprime uma série de arquivos grandes, mas você quer permitir que outros usuários imprimam no meio:

```
lp lotoftext; sleep 900; lp hugefile; sleep 900; lp anotherlargefile
```

A impressão de arquivos é discutida no [Chapter 8, Impressoras e impressão](#).

Programadores freqüentemente usam o comando `sleep` para parar um script ou programa em execução por um determinado tempo.

## O comando at

O comando **at** executa outros comandos em uma determinada hora, usando o seu shell padrão, a mesmo que você instrua o programa para proceder de modo distinto (veja a página `man`).

As opções do **at** são muito amigáveis, o que fica claro no exemplo abaixo:

```
steven@home:~> at tomorrow + 2 days
warning: commands will be executed using (in order) a) $SHELL
        b) login shell c) /bin/sh
at> cat reports | mail myboss@mycompany
at> <EOT>job 1 at 2001-06-16 12:36
```

Ao pressionar **Ctrl+D** encerra-se o utilitário **at** e é criada uma mensagem "EOT" .

O usuário *steven* faz uma coisa estranha aqui combinando dois comandos; nós estudaremos este tipo de prática no [Chapter 5, Redirecionamento de E/S](#), Redirecionamento de Entrada e Saída.

```
steven@home:~> at 0237
warning: commands will be executed using (in order) a) $SHELL
        b) login shell c) /bin/sh
at> cd new-programs
at> ./configure; make
at> <EOT>job 2 at 2001-06-14 02:00
```

A opção `-m` envia um email para o usuário quando o trabalho termina ou avisa se um trabalho não pode ser realizado. O comando **atq** lista os trabalhos; para evitar que os trabalhos se iniciem ao mesmo tempo, informe este comando antes de submeter trabalhos. Com o comando **atrm** você pode remover trabalhos agendados, caso tenha mudado de idéia.

É uma boa idéia pegar uma horários de execução inusitados, já que os trabalhos do sistema normalmente são executados em horas "cheias", como você pode ver na [the section called "Cron e crontab"](#) próxima seção. Por exemplo, trabalhos são muitas vezes executados exatamente à 1h da manhã (ex. indexação do sistema para atualizar a base de dados padrão do locate), de forma que, informar o horário 0100 pode facilmente sobrecarregar seu sistema ao invés de acelerá-lo. Para evitar que os trabalho sejam executados ao mesmo tempo, você também pode fazer uso do comando **batch**, que ordena os processos e alimenta o sistema de uma forma balanceada, evitando súbitas sobrecargas de uso dos recursos. Veja as páginas `man` para saber mais.

## Cron e crontab

O sistema cron é gerenciado pelo daemon **cron**. Ele obtém informações sobre quando e quais programas devem ser executados das entradas no `crontab` do sistema e dos usuários. Apenas o usuário `root` tem acesso aos `crontabs` do sistema, enquanto cada usuário deveria apenas ter acesso ao seu próprio `crontab`. Em alguns sistemas (alguns) usuários podem não ter acesso ao utilitário `cron`.

Quando da inicialização do sistema, o daemon `cron` procura no `/var/spool/cron/` pelas entradas no `crontab` que recebem seus nomes a partir das contas no `/etc/passwd`, pesquisa também o `/etc/cron.d/` e o `/etc/crontab`, e então usa estas informações a cada minuto para verificar se ainda há alguma coisa a se fazer. Ele executa os comandos como o usuário dono do arquivo `crontab` e envia um email com quaisquer saídas dos comando para o proprietário.

Em sistemas que usam o Vixie cron, os trabalhos que são executador a cada hora, dia, semana ou mês são armazenados em diferentes diretórios no `/etc` para que se possa ter uma visão do todo, ao contrário da função cron padrão no UNIX, em que todas as tarefas estão listadas em um grande arquivo.

Um exemplo de um arquivo `crontab` do Vixie:

```
[root@blob /etc]# more crontab
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/

# run-parts
# commands to execute every hour
01 * * * * root run-parts /etc/cron.hourly
# commands to execute every day
02 4 * * * root run-parts /etc/cron.daily
# commands to execute every week
22 4 * * 0 root run-parts /etc/cron.weekly
# commands to execute every month
42 4 1 * * root run-parts /etc/cron.monthly
```

## Alternativa

Você poderia também o comando **crontab -l** para exibir crontabs.

Algumas variáveis são configuradas e, após isto, ocorre o agendamento, uma linha por trabalho, começando com 5 campos de hora e data. O primeiro campo contém os minutos (de 0 a 59), o segundo define o horário da execução (0-23), o terceiro dia é um dia do mês (1-31), então o número do mês (1-12), o último é o dia da semana (0-7, sendo ambos, 0 e 7 o domingo). Um asterisco nestes campos representa a amplitude total no campo. São permitidas listas; para executar um trabalho de segunda à sexta, informe 1-5 no último campo, para executar um trabalho às segundas, quartas e sextas, informe 1,3,5.

Então aparece um usuário que deveria executar os processos que estão listados na última coluna. O exemplo acima é de uma configuração do Vixie cron na qual o root executa o programa **run-parts** em intervalos regulares, com os diretórios apropriados como opções. Nestes diretórios, os trabalhos que de fato serão executados na hora agendada são armazenados como scripts do shell, como este pequeno script que é executado diariamente para atualizar a base de dados usada pelo comando **locate**:

```
billy@ahost cron.daily]$ cat slocate.cron
#!/bin/sh
renice +19 -p $$ >/dev/null 2>&1
/usr/bin/updatedb -f "nfs,smbfs,ncpfs,proc,devpts" -e \
"/tmp,/var/tmp, /usr/tmp,/afs,/net"
```

Os usuários devem editar seus crontabs de uma maneira segura usando o comando **crontab -e**. Isto evitará que um usuário acidentalmente abra mais de uma cópia do seu arquivo crontab. O editor padrão é o **vi** (veja [Chapter 6, Editores de texto](#), mas é possível usar qualquer editor de texto, como o **gvim** ou o **gedit**, caso você se sinta mais confortável em um editor gráfico.

Quando você sair o sistema irá informá-lo que um novo crontab foi instalado.

Esta entrada crontab lembra o *billy* de ir ao seu clube toda quinta à noite:

```
billy:~> crontab -l
# DO NOT EDIT THIS FILE - edit the master and reinstall.
# (/tmp/crontab.20264 installed on Sun Jul 20 22:35:14 2003)
# (Cron version -- $Id: chap4.xml,v 1.28 2007-09-19 12:22:26 tille Exp $)
38 16 * * 3 mail -s "sports evening" billy
```

Após adicionar uma nova tarefa à agenda, o sistema irá informá-lo que um novo arquivo crontab foi instalado. Você não precisa reiniciar o daemon **cron** para que as mudanças tenham efeito. No exemplo, *billy* adicionou uma nova linha apontando para um script de backup:

```
billy:~> crontab -e
45 15 * * 3 mail -s "sports evening" billy
4 4 * * 4,7 /home/billy/bin/backup.sh
```

```
<--write and quit-->
```

```
crontab: installing new crontab
```

```
billy:~>
```

O scrip `backup.sh` é executado toda quinta e domingo. Veja [the section called "Scripts do shell"](#) para uma introdução ao scripting do shell. Tenha em mente que a saída de comandos, se houver, é encaminhada ao proprietário do arquivo crontab. Se nenhum serviço de email estiver configurado, você poderá encontrar a saída dos seus comandos na sua caixa de correio local, `/var/spool/mail/<your_username>`, em um arquivo texto.

## Quem executa meus comandos?

Não é preciso especificar o usuário que deverá executar os comandos. Eles são executados com as próprias permissões do usuários.

## Resumo



O Linux é um sistema operacional multi-usuário e multi-tarefa que possui um modo UNIX de manipular processos. A velocidade dos comandos pode depender de uma infinidade de pequenas coisas. Dentre outras coisas, aprendemos diversos novos comandos para visualizar e manipular processos. Segue uma lista:

**Table 4.3. Novos comandos no capítulo 4: Processos**

Comando	Significado
<b>at</b>	Põe tarefas em uma fila para executar mais tarde
<b>atq</b>	Lista as tarefas pendentes do usuário
<b>atrm</b>	Remove tarefas, determinadas pelo número da tarefa.
<b>batch</b>	Executa comandos quando o nível de carga do sistema permite.
<b>crontab</b>	Mantém arquivos do crontab para usuários individuais.
<b>halt</b>	Pára o sistema.
<b>init run level</b>	Inicialização do controle de processo
<b>jobs</b>	Lista tarefas atualmente em execução.
<b>kill</b>	Termina um processo.
<b>mesg</b>	Controla o acesso a escrita no seu terminal.
<b>netstat</b>	Exibe conexões de rede, tabelas de rota, estatísticas de dispositivo, conexões mascaradas e associações de multicast.
<b>nice</b>	Executa um programa com uma prioridade de agendamento modificada.
<b>pgrep</b>	Exibe processos.
<b>ps</b>	Relata status de processos.
<b>pstree</b>	Exibe uma árvore de processos.
<b>reboot</b>	Pára o sistema.
<b>renice</b>	Altera prioridade dos processos em execução.
<b>shutdown</b>	Desliga o sistema.
<b>sleep</b>	Suspende por um tempo específico.
<b>time</b>	Marca o tempo de um comando ou relata o consumo de recursos.
<b>top</b>	Exibe os processos da CPU que estão no topo.
<b>uptime</b>	Mostra quanto tempo o sistema esteve operando.
<b>vmstat</b>	Relata estatísticas de memória virtual.
<b>w</b>	Mostra quem está conectado e o que estão fazendo.
<b>wall</b>	Envia uma mensagem os terminais de todos.
<b>who</b>	Mostra quem está conectado.
<b>write</b>	Envia uma mensagem para outro usuário.

## Exercícios

Estes são alguns exercícios que irão ajudá-lo a sentir os processos em execução no seu sistema.

### Geral

- Execute o **top** em um terminal enquanto faz os exercícios em outro.
- Execute o comando **ps**.
- Leia as páginas man para descobrir como exibir todos os seus processos.
- Execute o comando **find /**. Qual efeito ele gera na carga do sistema? Pare este comando.
- No modo gráfico, inicie o programa **xclock** no primeiro plano. Então faça-o continuar sua execução no plano de fundo. Pare-o usando o comando **kill**.
- Execute o **xcalc** diretamente no plano de fundo, de forma que o prompt do terminal que o informou seja liberado.
- O que o comando **kill -9 -1** faz?
- Abra dois terminais ou duas janelas de terminal e use o **write** para enviar uma mensagem de um para o outro.
- Informe o comando **dmesg**. O que ele diz?
- Quanto demora para executar o comando **ls** no diretório atual?
- Baseado nas entradas de processos no `/proc`, de propriedade de sua UID, como você faria para descobrir a quais processos elas se referem?

- Há quanto tempo o seu sistema está rodando?
- Qual é a sua TTY atual?
- Diga 3 processos que não poderiam ser filhos do **init**.
- Diga 3 processos que usam o modo SUID. Explique o porquê.
- Diga quais comandos estão normalmente dando causa à maior carga no seu sistema.

## Inicialização, init etc.

- Você pode reiniciar o seu sistema como um usuário normal? Por que?
- De acordo com o seu nível de execução atual, diga três passos que são tomados durante o desligamento do sistema.
- Como você muda o nível de execução do sistema? Mude do nível de execução atual para o nível 1 e vice e versa.
- Faça uma lista de todos os serviços de daemons que são iniciados quando o seu sistema se inicializa.
- Qual kernel é carregado na inicialização?
- Suponha que você precisa iniciar algum servidor exótico quando da inicialização. Até agora, você efetuava o login no sistema e iniciava o servidor manualmente usando um script chamado `deliver_pizza` na sua pasta home. O que é preciso fazer para que o servidor se inicie automaticamente no nível de execução 4, que você especificamente definiu para este propósito?

## Agendamento

- Use o comando **sleep** para criar um lembrete de que o seu macarrão estará pronto em dez minutos.
- Crie um trabalho **at** que copia todos os arquivos em sua pasta home para o `/var/tmp` em meia hora. Você pode querer criar um subdiretório no `/var/tmp`.
- Crie um cronjob que realiza esta tarefa toda segunda e sexta durante o almoço.
- Verifique se funciona.
- Crie um erro em uma entrada crontab, como informar um comando inexistente, **copy** ao invés de **cp**. O que acontece quando da execução da tarefa?

## Chapter 5. Redirecionamento de E/S

### Table of Contents

#### [Redirecionamentos simples](#)

- [O que são entrada e saída padrão?](#)
- [Os operadores de redirecionamento](#)

#### [Recursos avançados de redireção](#)

- [Uso dos descritores de arquivo](#)
- [Exemplos](#)

#### [Filtros](#)

- [Mais sobre o grep](#)
- [Filtrando a saída](#)

#### [Resumo](#)

#### [Exercícios](#)

### Abstract

O presente capítulo descreve em detalhes o poderoso mecanismo UNIX de redirecionamento de entrada, saída e erros. Os tópicos incluem:

- Padrão de entrada, saída e erros
- Operadores de redirecionamento
- Como usar a saída de um comando como entrada para outro
- Como transcrever a saída de um comando em um arquivo para referência posterior
- Como apensar a saída de múltiplos comandos a um arquivo

- Redireção de entrada
- Manipulação das mensagens de erro padrão
- Como combinar o redirecionamento dos fluxos de entrada, saída e erros
- Filtros de saída

## Redirecionamentos simples

### O que são entrada e saída padrão?

A maior parte dos comandos Linux lêem entrada, como um arquivo ou outro atributo do comando e escrevem saída. O padrão é que a entrada seja informada por meio do teclado e a saída seja exibida na tela. O teclado é o dispositivo de *entrada padrão* (`stdin`) e a tela, ou uma janela de terminal, é o dispositivo de *saída padrão* (`stdout`).

Entretanto, como o Linux é um sistema flexível, estas configurações padronizadas não são necessariamente as usadas em dado sistema. A saída padrão, por exemplo, em um servidor monitorado dentro de um ambiente complexo poderia ser uma impressora.

### Os operadores de redirecionamento

#### Redirecionamento de saída com `>` e `|`

Acontecerá de você precisar colocar a saída de um comando em um arquivo ou então de querer informar um outro comando na saída do primeiro. Isto é conhecido como redirecionamento de saída. O redirecionamento é feito usando `>` (símbolo de maior que) ou então através do operador `|` (pipe), que envia a saída padrão de um comando para a entrada padrão de outro.

Como já visto, o comando **cat** concatena arquivos e os envia juntos para a saída padrão. Redirecionando esta saída para um arquivo, tal arquivo será criado ou substituído se já existente, então tome cuidado.

```
nancy:~> cat test1
some words
```

```
nancy:~> cat test2
some other words
```

```
nancy:~> cat test1 test2 > test3
```

```
nancy:~> cat test3
some words
some other words
```

### Não sobrescreva!

Tome cuidado para não sobrescrever arquivos pré-existentes (importantes) ao redirecionar a saída de um comando. Muitas shells, incluindo aqui a Bash, têm um recurso interno para protegê-lo deste risco: o **noclobber**. Veja as páginas Info para obter mais informações. No Bash, você poderia adicionar o comando **set -onoclobber** ao seu arquivo de configuração `.bashrc`, de modo a prevenir, substituições acidentais de arquivos.

Redirecionar “nada” a um arquivo existente equivale a esvaziar o arquivo:

```
nancy:~> ls -l list
-rw-rw-r-- 1 nancy nancy 117 Apr 2 18:09 list
```

```
nancy:~> > list
```

```
nancy:~> ls -l list
-rw-rw-r-- 1 nancy nancy 0 Apr 4 12:01 list
```

Este procedimento é chamado de *truncamento*.

O mesmo redirecionamento quando feito para um arquivo inexistente irá criar um novo arquivo vazio com o nome dado:

```
nancy:~> ls -l newList
ls: newList: No such file or directory
```

```
nancy:~> > newList
```

```
nancy:~> ls -l newList
-rw-rw-r-- 1 nancy nancy 0 Apr 4 12:05 newList
```

[Chapter 7, Lar doce /lar \(/home\)](#) dá outros exemplos sobre o use deste tipo de redirecionamento.

Alguns exemplos usando piping de comandos:

Para encontrar uma palavra em algum texto, exibir todas as linhas em que ocorre “padrão1” e excluir

aquelas em que ocorre o “padrão2”:

```
grep pattern1 file | grep -v pattern2
```

Para exibir o conteúdo de um diretório listando uma página por vez:

```
ls -la | less
```

Para encontrar um arquivo em um diretório:

```
ls -l | grep part_of_file_name
```

## Redirecionamento de entrada

Em outro caso, pode ser que se deseje que um arquivo seja a entrada que, em regra, não aceita um arquivo como opção. Este redirecionamento de entrada é feito usando o “<” (símbolo de menor que).

Abaixo se encontra um exemplo do envio de um arquivo usando a redireção de entrada.

```
andy:~> mail mike@somewhere.org < to_do
```

Se o usuário *cadu* existe no sistema, não é necessário digitar o endereço completo. Se você deseja alcançar alguém na Internet, insira o endereço completo como um argumento para o **mail**.

A leitura é mais difícil do que o iniciante **cat file | mail someone**, mas é certamente um modo muito mais elegante de se usar as ferramentas disponíveis.

## Combinando redirecionamentos

O exemplo seguinte combina redirecionamentos de entrada e saída. O arquivo `texto.txt` é primeiramente conferido, em busca de erros de tipografia, redirecionando-se o arquivo com o log dos erros :

```
spell < text.txt > error.log
```

O comando a seguir lista todos os comandos que se pode informar para examinar outro arquivo se quando estiver usando o **less**:

```
mike:~> less --help | grep -i examine
:e [file]      Examine a new file.
:n             * Examine the (N-th) next file from the command line.
:p             * Examine the (N-th) previous file from the command line.
:x             * Examine the first (or N-th) file from the command line.
```

A opção `-i` é usada para realizar buscas sem diferenciação de maiúsculas e minúsculas - lembre que os sistemas UNIX são sensíveis à maiusculização.

Se você quer salvar a saída deste comando para referência futura, redirecione-a para um arquivo:

```
mike:~> less --help | grep -i examine > examine-files-in-less
```

```
mike:~> cat examine-files-in-less
:e [file]      Examine a new file.
:n             * Examine the (N-th) next file from the command line.
:p             * Examine the (N-th) previous file from the command line.
:x             * Examine the first (or N-th) file from the command line.
```

Em tese, a saída de um comando pode ser redirecionada para outro comando tantas vezes quantas for preciso, desde que estes comando aceitem normalmente ler a entrada da entrada padrão e escrever a saída para a saída padrão. Às vezes isto não é verdade, mas nestes casos podem existir opções especiais que instruem estes comando a agir de acordo com as definições padrão; então, na hipótese de ocorrer algum erro, leia a documentação (as páginas `man` e `Info` ) dos comandos usados.

Novamente, certifique-se de que você não está informando nomes de arquivos existentes que você ainda necessite. Redirecionar a saída para arquivos existentes irá substituir o conteúdo destes arquivos.

## O operador >>

Ao invés de sobrescrever o conteúdo do arquivo, também é possível adicionar texto para um arquivo existente usando dois símbolos de maior que subsequentes.

Exemplo:

```
mike:~> cat wishlist
more money
less work
```

```
mike:~> date >> wishlist
```

```
mike:~> cat wishlist
more money
less work
Thu Feb 28 20:23:07 CET 2002
```

O comando **date** normalmente colocaria a última linha na tela; agora é apensado ao arquivo `lista_de_desejos`.

## Recursos avançados de redireção

### Uso dos descritores de arquivo

Existem três tipos de E/S, os quais têm identificadores próprios, chamados de descritores de arquivo:

- entrada padrão: 0
- saída padrão: 1
- erro padrão: 2

Nas descrições a seguir, se o número do descritor de arquivo for omitido e o primeiro caractere do operador de redireção for `<`, a redireção se refere à entrada padrão (descritor de arquivo 0). Se o primeiro caractere do operador de redireção for `>`, a redireção se refere a saída padrão (descritor de arquivo 1).

Alguns exemplos práticos tornarão isto mais claro:

```
ls > dirlist 2>&1
```

redirecionará ambos, saída e erro padrão, para o arquivo `listadir`, enquanto o comando

```
ls 2>&1 > dirlist
```

direcionará apenas a saída padrão para `listadir`. Esta pode ser uma opção útil para programadores.

As coisas estão se tornando um tanto quanto complicadas agora, não confunda o uso da ampulheta aqui com aquele no [the section called "Processos interativos"](#), onde ela foi usada para executar um processo em segundo plano. Aqui, ela serve simplesmente como uma indicação de que o número que segue não é nome de arquivo, mas antes um local para o qual o fluxo de dados é direcionado. Note, ainda, que o símbolo de maior-que não deve ser separado por espaços do número do descritor de arquivo. Caso contrário, estar-se-ia a direcionar a saída para um arquivo novamente. Demonstra-se isto no exemplo a seguir:

```
[nancy@asus /var/tmp]$ ls 2> tmp
```

```
[nancy@asus /var/tmp]$ ls -l tmp
-rw-rw-r-- 1 nancy nancy 0 Sept  7 12:58 tmp
```

```
[nancy@asus /var/tmp]$ ls 2 > tmp
ls: 2: No such file or directory
```

O primeiro comando que a *nancy* executa está correto (ainda que não sejam gerados erros e, portanto, permaneça vazio o arquivo para o qual ele seria redirecionado). O segundo comando entende que 2 é um nome de arquivo, inexistente neste caso, então é exibido um erro.

Todos estes recursos são explicados em detalhes nas páginas Info do Bash.

## Exemplos

### Analisando erros

Se um processo gera muitos erros, este é um modo de examiná-los :

```
command 2>&1 | less
```

Isto é usado em geral quando se cria um novo software usando o comando **make**, como em:

```
andy:~/newsoft> make all 2>&1 | less
--output omitted--
```

### Separando a saída padrão do erro padrão

Construtos como estes são muito usados por programadores, de forma que a saída é exibida em uma janela do terminal e os erros em outra. Descubra qual pseudo terminal você está usando informando o comando **tty**:

```
andy:~/newsoft> make all 2> /dev/pts/7
```

### Escrevendo para a saída e arquivos simultaneamente

É possível usar o comando **tee** para copiar a entrada para a saída padrão e um ou mais arquivos de saída em um único passo. Usando a opção `-a` para o **tee** resulta em apensar a entrada para o arquivo(s). Isto é útil se você quer ao mesmo tempo ver e salvar a saída. Os operadores `>` e `>>` não permitem realizar as duas ações simultaneamente.

Esta ferramenta é normalmente usada por meio de um pipe (|), como demonstrado no exemplo abaixo:

```
mireille ~/test> date | tee file1 file2
Thu Jun 10 11:10:34 CEST 2004

mireille ~/test> cat file1
Thu Jun 10 11:10:34 CEST 2004

mireille ~/test> cat file2
Thu Jun 10 11:10:34 CEST 2004

mireille ~/test> uptime | tee -a file2
11:10:51 up 21 days, 21:21, 57 users,  load average: 0.04, 0.16, 0.26

mireille ~/test> cat file2
Thu Jun 10 11:10:34 CEST 2004
11:10:51 up 21 days, 21:21, 57 users,  load average: 0.04, 0.16, 0.26
```

## Filtros

Quando um programa realiza operações na entrada e escreve o resultado para a saída padrão, ele é denominado de filtro. Um dos usos mais comum dos filtros é para reestruturar a saída. Discutiremos alguns dos filtros mais importantes abaixo.

### Mais sobre o grep

Como visto no [the section called "O comando grep"](#), o **grep** vasculha a saída linha por linha, buscando por ocorrências. Todas as linhas em que constar o critério serão exibidas na saída padrão. Este comportamento pode ser revertido usando a opção `-v`.

Alguns exemplos: suponha que queremos saber quais arquivos em um determinado diretório foram modificados em Fevereiro:

```
jenny:~> ls -la | grep Feb
```

O comando **grep**, como a maior parte dos comando, diferencia maiúscula de minúsculas. Use a opção `-i` para inibir esta diferenciação. Muitas extensões GNU encontram-se disponíveis também, como `--colour`, que facilita muito realçando os termos da busca em longas linhas e o `--after-context`, que exibe o número de linhas após a última ocorrência. Você pode informar um **grep** recursivo que busca todos os subdiretórios do diretório fornecido usando a opção `-r`. Como de costume, as opções podem ser combinadas.

Expressões regulares podem ser usada para detalhar ainda mais as ocorrências de caracteres desejadas dentre todas as linhas de entrada. O melhor modo de começar com expressões regulares é lendo a documentação do **grep**. Um excelente capítulo encontra-se inclusive no na página Info do **grep**. Uma vez que nos afastaríamos muito se discutíssemos os detalhes das expressões regulares, é altamente recomendável começar agora se quiser saber mais a respeito delas.

Brinque um pouco com o **grep**, vale muito a pena gastar um tempo neste muito básico, porém poderosíssimo comando de filtragem. Os exercício ao final deste capítulo o ajudarão a começar, veja [the section called "Exercícios"](#).

### Filtrando a saída

O comando **sort** dispõe as linhas em ordem alfabética:

```
thomas:~> cat people-I-like | sort
Auntie Emmy
Boyfriend
Dad
Grandma
Mum
My boss
```

Mas existem muitas outras coisas que o **sort** pode fazer. Ver o tamanho do arquivo, por exemplo. Com este comando, o conteúdo de um diretório é disposto a partir do menor arquivo até o maior dentre eles:

```
ls -la | sort -nk 5
```

### Sintaxe antiga do sort

É possível obter o mesmo resultando com `ls -la | sort +4n`, mas esta é uma fomra antiga que não se adequa aos padrões atuais.

O comando **sort** também usado combinado com o programa **uniq** (ou **sort -u**) para dispôr a saída e filtrar as ocorrências em duplicidade:

```
thomas:~> cat itemlist
1
4
2
5
```

34  
567  
432  
567  
34  
555

```
thomas:~> sort itemlist | uniq  
1  
2  
34  
4  
432  
5  
555  
567
```

## Resumo

Neste capítulo aprendemos como os comandos podem ser vinculados uns aos outros e como a entrada de um comando pode ser usada como saída para outro.

Redirecionamento de entrada/saída é uma tarefa comum nas máquinas UNIX e Linux. Este poderoso mecanismo permite um uso flexível dos tijolos de que o UNIX é feito.

Os redirecionamentos mais comuns são `>` e `|`. Para um panorama dos comandos de redireção e construtos do shell [Appendix C, Recursos do Shell](#).

**Table 5.1. Novos comando no capítulo 5: Redirecionamento E/S**

Comando	Significado
<b>date</b>	Exibe informação sobre data e hora
<b>set</b>	Configura as opções do shell.
<b>sort</b>	Organiza linhas de texto
<b>uniq</b>	Remove linhas duplas de um arquivo organizado pelo sort.

## Exercícios

Estes exercícios fornecem outros exemplos de como combinar comandos. O objetivo principal é testar e usar a tecla **Enter** o mínimo possível.

Todos os exercício podem ser feitos usando uma conta de usuário normal e de forma a gerar alguns erros. Quando os estiver fazendo, não se esqueça de ler as páginas man.

- Use o comando **cut** na saída de uma longa lista de diretório para exibir apenas as permissões dos arquivos. Então, encaminhe esta saída para o **sort** e então filtre com o **uniq** retirando as linhas em duplicidade. Use então o **wc** para contar quantos tipos de permissões diferentes existem no diretório.
- Coloque a saída do **date** em um arquivo. Apense a saída do **ls** a este arquivo. Envie este arquivo para a sua caixa de emails local (não especifique nenhum `<@domínio>`, apenas o nome de usuário é suficiente). Quando estiver usando o Bash, você verá um aviso de novo email.
- Liste os dispositivos no `/dev` que estão sendo usado pelo seu UID. Encaminhe através do **less** para vê-los de maneira adequada.
- Informe os comandos a seguir como um usuário sem privilégios. Determine a entrada, saída e erros padrão para cada comando.

Cheque os seus resultados informando os comando novamente, agora redirecionando a saída padrão para o arquivo `/var/tmp/saida` e o erro padrão para o arquivo `/var/tmp/erros`.

- `cat nonexistentfile`
- `file /sbin/ifconfig`
- `grep root /etc/passwd /etc/nofiles > grepresults`
- `/etc/init.d/sshd start > /var/tmp/output`
- `/etc/init.d/crond start > /var/tmp/output 2>&1`
- Quantos processos a sua máquina está executando neste momento?
- Quantos arquivos invisíveis a existem no seu diretório home?
- Use o **locate** para encontrar a documentação sobre o kernel.

- Descubra qual arquivo contém a seguinte ocorrência:

```
root:x:0:0:root:/root:/bin/bash
```

E esta:

```
system:                root
```

- Veja o que acontece quando é informado este comando:

```
> time; date >> time; cat < time
```

- Qual comando você usaria para checar qual script no `/etc/init.d` começa um dado processo?

## Chapter 6. Editores de texto

### Table of Contents

#### [Editores de texto](#)

[Por que eu deveria usar um editor?](#)

[Qual editor devo usar?](#)

#### [Usando o editor Vim](#)

[Dois modos](#)

[Comandos básicos](#)

[O modo fácil](#)

#### [O Linux no escritório](#)

[História](#)

[Suítes e programas](#)

[Observações](#)

#### [Resumo](#)

#### [Exercícios](#)

### Abstract

Neste capítulo, discutiremos a importância de dominar um editor. Focaremos principalmente no editor **vi** aprimorado.

Após terminar este capítulo, você será capaz de:

- Abrir e fechar arquivos no modo texto
- Editar arquivos
- Procurar texto
- Desfazer erros
- Misturar arquivos
- Recuperar arquivos perdidos
- Encontrar um programa ou suíte para uso de escritório

## Editores de texto

### Por que eu deveria usar um editor?

É muito importante ser capaz de usar ao menos um editor no modo texto. Aprender a usar um editor no seu sistema é o primeiro passo para a independência.

Será preciso dominar um editor no próximo capítulo, já que precisaremos editar arquivos que influenciam o ambiente. Como um usuário avançado, você poderá desejar escrever script, livros, desenvolver websites ou novos programas. Dominar um editor irá aprimorar sua produtividade e capacidade imensamente.

### Qual editor devo usar?

Nosso foco são os editores de texto, que possam ser usados mesmo em sistemas sem um ambiente gráfico e em terminais. A vantagem adicional de dominar um editor de texto é usá-lo em máquinas remotas. Uma vez que você não precisa transferir todo o ambiente gráfico através da rede, trabalhar com editores no modo texto aprimora sua velocidade tremendamente.

Existem, como de costume, várias formas de enfrentar o problema. Vejamos quais os editores mais comuns:



## Ed

O editor **ed** é orientado para linhas e usado para criar, exibir, modificar e manipular arquivos texto tanto no modo interativo como por meio de scripts do shell.

O **ed** é o editor de texto original das máquinas UNIX e, portanto, está amplamente disponível. Todavia, ele em grande parte foi substituído por editores de texto de tela cheia como o **emacs** e o **vi**, veja abaixo.

## GNU Emacs

O Emacs é um editor extensível, personalizável, auto-documentado, com exibição em tempo real, conhecido em muitos sistemas UNIX. O texto editado é visível na tela e é automaticamente atualizado quando são digitados os comandos. É um editor em tempo real, pois a sua exibição é atualizada com frequência, via de regra logo após ser digitado um ou dois caracteres. Isto minimiza a quantidade de informações que você precisa manter em mente enquanto edita. O Emacs é chamado de avançado porquanto provê ferramentas que vão além da mera inserção e remoção: controla subprocessos; indentação automática de programas; exibição de dois ou mais arquivos de uma vez; edição de texto formatado; e manipulação de caracteres, palavras, linhas, sentenças, parágrafos e páginas, assim como de expressões e comentários em diversas linguagens de programação distintas.

*Auto-documentado* implica que a qualquer tempo é possível digitar um caractere especial, **Ctrl+H**, para descobrir quais são as opções disponíveis. Também é possível usá-lo para descobrir o que um comando faz ou todos os comandos relativos a um assunto. *Personalizável* quer dizer que é possível modificar as definições dos comandos do Emacs. Por exemplo, se estiver usando uma linguagem de programação na qual os comentários começam “<\*” e terminam com “\*>”, é possível dizer ao manipulador de comentários do Emacs usar estes caracteres. Outro modo de personalizá-lo é reorganizando os comandos. Por exemplo, se você prefere os quatro movimentos básicos de cursor (cima, baixo, esquerda e direita) no teclado em forma de diamante, é possível vinculá-las assim.

*Extensível* quer dizer que é possível ir além da mera personalização e escrever comandos inteiramente novos, programas na linguagem Lisp que são executados pelo interpretador Lisp do próprio Emacs. O Emacs é um sistema extensível *online*, o que significa que ele é dividido em muitas funções que chamam umas as outras, sendo que todas podem ser redefinidas ao longo de uma sessão. Quase qualquer parte do Emacs pode ser substituída sem a necessidade de realizar uma cópia de todo o Emacs. A maior parte dos comandos de edição do Emacs são escritos em Lisp; as poucas exceções poderiam ter sido escritas em Lisp, mas foram escritas em C por conta da eficiência. Embora apenas um programador consiga escrever uma extensão, qualquer um pode usá-la a partir de então.

Quando executado dentro do Sistema de Janelas X (iniciado como **xemacs**) o Emacs provê os seus próprios menus e vínculos convenientes com os botões do mouse. Mas o Emacs provê muitas das funcionalidades de um sistema de janelas em um terminal apenas de texto. Por exemplo, é possível olhar diferentes arquivos de uma só vez, movimentar-se pelos arquivos e editá-los enquanto são executados comandos do shell.

## Vi(m)

Vim é uma sigla para “Vi IMproved” (Vi aprimorado). Inicialmente era “Vi IMitação”, mas havia tantas melhorias que a mudança de nome foi apropriada. Vim é um editor de texto que inclui quase todos os comandos do programa **vi** do UNIX, além de muitos outros.

Os comandos no editor **vi** são informados usando apenas o teclado, o que trás a vantagem de poder manter os dedos no teclado e os olhos na tela, ao invés de ter que repetidas vezes recorrer ao mouse. Para aqueles que quiserem, entretanto, é possível ativar suporte para mouse e uma versão GUI com barra de rolagem.

Faremos referência ao **vi** ou **vim** ao longo deste livro para editar os arquivos, embora você esteja livre para usar o editor de sua preferência. Entretanto, recomendamos que você ao menos aprenda o básico sobre o **vi**, já que ele é o editor padrão nos sistemas UNIX, enquanto o **emacs** pode ser um pacote opcional. Podem existir pequenas diferenças entre diferentes computadores e terminais, mas o ponto principal é que se você for capaz de trabalhar com o **vi**, será capaz de sobreviver em qualquer sistema UNIX.

Além do comando **vim**, os pacotes do vim podem conter também o **gvim**, a versão Gnome do **vim**. Usuários iniciantes poderão achar este mais fácil, por conta da ajuda oferecida pelos menus quando se esqueceu ou não se sabe como realizar uma tarefa de edição específica usando os comandos padrões do **vim**.

## Usando o editor Vim

### Dois modos

O editor **vi** é uma ferramenta muito poderosa e possui um extenso manual interno, que pode ser acionado através do comando **:help** ao iniciar o programa (ao invés de usar o **man** ou o **info**, os quais nem de longe possuem a mesma quantidade de informações). Discutiremos aqui apenas o básico para que você possa começar a usá-lo.

Algo que torna o **vi** confuso para o iniciante é que ele opera em dois modos: modo comando e modo de

inserção. O editor sempre é iniciado no modo comando. Os comandos movem você ao longo do texto, busca, substituição, blocos de marcação, realizam outras tarefas de edição e alguns mudam o editor para o modo de inserção.

Isto significa que cada tecla tem não um, mas dois significados: ela pode representar ou um comando para o editor quando no modo de comando ou um caractere que se deseja inserir no texto, quando no modo texto.

## Pronúncia

No original, pronuncia-se “vi-ai”.

## Comandos básicos

### Movendo-se através do texto

A movimentação através do texto em regra é possível a partir das setas . Do contrário, tente:

- **h** para mover o cursor para a esquerda
- **l** para movê-lo para a direita
- **k** para movê-lo para cima
- **j** para movê-lo para baixo

SHIFT-G irá levar o prompt ao final do documento.

### Operações básicas

Estes são alguns comando populares do **vi**:

- **n dd** deleta n linhas a contar da posição atual do cursor.
- **n dw** deleta n palavras à direita do cursor.
- **x** deleta o caractere sobre o qual o cursor está.
- **:n** leva o cursor para a linha n.
- **:w** salva as modificações no arquivo.
- **:q** sai do editor.
- **:q!** força a finalização quando se quer sair de um arquivo que ainda contém mudanças não salvas.
- **:wq** salva as alterações e sai.
- **:w novoarquivo** salva o texto em novoarquivo.
- **:wq!** sobrepõe a permissão apenas para leitura (se você tiver permissão para sobrepôr permissões, por exemplo se estiver usando a conta *root*).
- **/umasequência** procura pela seqüência ao longo do arquivo e posiciona o cursor na primeira ocorrência abaixo de sua posição.
- **/** realiza novamente a mesma busca, movendo o cursor para a próxima ocorrência.
- **:1, \$s/palavra/outrapalavra/g** substitui palavra por outra palavra ao longo de todo o arquivo.
- **yy** copia um bloco de texto.
- **n p** cola o que foi copiado n vezes.
- **:recover** recupera um arquivo após uma interrupção inesperada.

### Comandos que alteram o editor para o modo de inserção.

- **a** apensa: muda o cursor uma posição à direita antes de alternar para o modo de inserção.
- **i** insere
- **o** insere uma linha vazia abaixo da posição atual do cursor e move-o para aquela linha.

Pressionando a tecla **Esc** retorna para o modo de comando. Se você não está certo de qual modo está porque que você está usando uma versão bem antiga do **vi** que não exibe uma mensagem “INSERT”, digite **Esc** de forma a se certificar de que está no modo comando. É possível que o sistema dê um pequeno alerta se você já estiver no modo comando, bipando ou acionando uma campainha visual (um flash na tela). Este comportamento é normal.

## O modo fácil

Ao invés de ler o texto, que é um tanto cansativo, você pode usar o vimtutor para aprender os seus primeiros comandos no Vim. Ele é um tutorial de trinta minutos que ensina as funções mais básicas do Vim em oito fáceis exercícios. Embora não se possa aprender tudo sobre o vim em apenas meia hora, o tutorial foi projetado para descrever os comandos suficientes para que você possa usar o Vim como um editor genérico.

No UNIX e no MS Windows, se o Vim tiver sido devidamente instalado, é possível iniciá-lo a partir do shell ou linha de comando, informando o comando **vimtutor**. Isto fará uma cópia do arquivo tutor, de forma que você possa editá-lo sem correr o risco de danificar o original. Existem algumas poucas traduções do tutorial. Para descobrir se a sua está disponível, use o código de duas letras do idioma. Para Português seria **vimtutor pt** (se estiver instalado no sistema).

## O Linux no escritório

### História

Ao longo da década passada o mercado dos escritórios foi dominado pelo MS Office e admitamos: os formatos Microsoft Word, Excel e PowerPoint são padrões na indústria com os quais teremos que lidar cedo ou tarde.

Este monopólio da Microsoft demonstrou ser uma grande desvantagem para que o Linux conseguisse novos usuários, então um grupo de desenvolvedores alemães iniciou o projeto StarOffice que era, e ainda é, uma tentativa de fazer um clone do MS Office. Sua empresa, Star Division, foi incorporada pela Sun Microsystems ao final dos anos 1990, um pouco antes do lançamento da versão 5.2. A Sun continuou o seu desenvolvimento, porém restringiu o acesso ao código-fonte. Entretanto, o desenvolvimento do código-fonte original continuou a ser realizado pela comunidade Open Source, que foi obrigada a rebatizar o projeto OpenOffice. O OpenOffice hoje se encontra disponível para uma variedade de plataformas, incluindo o MS Windows, Linux, MacOS e Solaris. Você pode ver uma captura de tela no [the section called "Dez anos de experiência a seu dispor"](#).

Quase que simultaneamente, alguns outros famosos projetos decolaram. Uma outra alternativa muito comum ao uso do MS Office é o KOffice, a suíte de escritório que se tornou popular entre os usuários do SuSE. Como o original, este cline incorpora aplicativos compatíveis com o MS Word e Excel e muito mais.

Projetos menores lidam com determinados aplicativos da suíte exemplo da MS, como Abiword e MS Wordview para compatibilidade com os documentos do MS Word e o Gnumeric para visualização e criação de planilhas compatíveis com o Excel.

### Suítes e programas

As distribuições atuais em geral vem com todas as ferramentas necessárias. Uma vez que estas provêm excelentes diretrizes e índices navegáveis no menu Ajuda, não os discutiremos em detalhes. Para mais referências, veja na documentação do seu sistema ou procure na web pelos sites dos projetos, como

- <http://www.openoffice.org>
- <http://www.koffice.org>
- [Freshmeat](#) e [SourceForge](#) para outros projetos.

### Observações

#### Uso geral dos documentos de escritório

Tente limitar o uso de documentos de escritório ao fim para o qual é destinado: escritório.

Um exemplo: é de deixar a um usuário Linux louco quando lhe é enviado um email que diz no corpo algo como: " Olá, eu quero lhe dizer uma coisa, veja em anexo", e então o anexo vem a ser um documento MS Word com o seguinte: "Olá amigo, como vai no novo trabalho? Vai ter tempo de almoçar comigo amanhã?" Outra má idéia é que a sua assinatura esteja anexa em tal documento. Se você precisar assinar mensagens ou arquivos, use o GPG, GNU Privacy Guard, compatível com os certificados PGP ou SSL (Secure Socket Layer).

Estes usuários não se incomodam porque eles não conseguem ler estes documentos ou porque preocupam-se com o fato de que eles em geral criam arquivos muito grandes, mas antes pela implicação de estar usando o MS Windows e, possivelmente, por conta do trabalho extra de iniciar programas adicionais.

#### Arquivos de configuração do sistema e de usuários

No próximo capítulo, começaremos a configurar nosso ambiente e isto pode incluir editar todos os tipos de arquivos que influenciam como um programa se comporta.

*Não edite qualquer destes arquivos com um componente de escritório!*

As especificações do formato de arquivo padrão fariam com que o programa adicionasse diversas linhas

de código, definindo o formato do arquivo e as fontes usadas. Estas linhas não serão devidamente interpretadas pelos programas que deles dependem, do que resultarão erros e uma pane do programa lendo o arquivo. Em alguns casos, é possível salvar o arquivo como texto puro, mas você incorrerá em problemas se fizer disto um hábito.

### **Mas eu quero um editor de textos gráfico!**

Se você realmente insiste, tente o **gedit**, **kedit**, **kwrite** ou **xedit**; estes programas fazem apenas arquivos texto, que é o que precisaremos. Se você pretende fazer qualquer coisa séria, porém, fique com um editor de texto real como o **vim** ou o **emacs**.

Uma alternativa aceitável é o **gvim**, a versão Gnome do vim. Ainda será necessário usar os comandos do vi, mas se você ficar preso, poderá buscar por eles nos menus.

## **Resumo**

Neste capítulo aprendemos a usar um editor. Embora a escolha de qual usar caiba a você, é preciso saber usar pelo menos um.

O **vi** encontra-se disponível em todos os sistemas UNIX.

A maior parte das distribuições do Linux incluem uma suíte de escritório e um editor de texto gráfico.

## **Exercícios**

Este capítulo tem apenas um exercício: inicie o tutorial do Vim informando **vimtutor** em uma sessão de terminal e inicie o aprendizado.

Alternativamente, você pode iniciar o **emacs** e digitar **Ctrl+H** e então **T** para invocar o tutorial do Emacs.

O único caminho é a prática!

## **Chapter 7. Lar doce /lar (/home)**

### **Table of Contents**

#### [Boa manutenção do lar](#)

- [Introdução](#)
- [Crie espaço](#)

#### [Seu ambiente texto](#)

- [Variáveis do ambiente](#)
- [Arquivos de configuração do shell](#)
- [Um conjunto típico de arquivos de configuração](#)
- [O prompt do Bash](#)
- [Scripts do shell](#)

#### [O ambiente gráfico](#)

- [Introdução](#)
- [O Sistema de Janelas X](#)
- [Configuração do servidor X](#)

#### [Configurações de Região](#)

- [Configuração de teclado](#)
- [Fontes](#)
- [Data e Fuso Horário](#)
- [Idioma](#)
- [Informações específicas do país](#)

#### [Instalando novos programas](#)

- [Geral](#)
- [Formatos de pacotes](#)
- [Automatizando o gerenciamento de pacotes e atualizações](#)
- [Atualizando o seu kernel](#)
- [Instalando pacotes extras a partir do CD de instalação](#)

#### [Resumo](#)

#### [Exercícios](#)

- [Ambiente shell](#)
- [Ambiente gráfico](#)

### **Abstract**

Este capítulo trata da configuração do seu ambiente. Agora que sabemos como usar um editor, podemos mudar todos os tipos de arquivos para nos sentirmos melhor em casa. Após completar este capítulo, você saberá mais sobre:

- Organizar o seu ambiente
- Arquivos de comuns de configuração do shell
- Configurações do shell
- Configurar o prompt
- Configurar o ambiente gráfico
- Aplicativos de áudio e vídeo
- Gerenciadores de exibição e janelas
- Como funciona o servidor X
- Configurações de linguagem e fonte
- Instalação de novos programas
- Atualizar os pacotes existentes

## Boa manutenção do lar

### Introdução

Como já mencionado, é fácil bagunçar o sistema. Não se pode enfatizar mais a importância de mantê-lo limpo. Quando se aprende isto desde o começo, se torna um bom hábito, que poupará muito tempo quando se estiver programando em um ambiente Linux ou UNIX ou aparecerem tarefas de gerenciamento do sistema. Aqui se encontram descritos alguns modos de tornar a vida mais fácil :

- Crie um diretório `bin` para os seus programas e scripts.
- Organize os arquivos não executáveis em diretórios apropriados e crie tantos deles quantos forem necessários. Como exemplo, pode-se criar diretórios distintos para imagens, documentos, projetos, arquivos recebidos, planilhas, arquivos pessoais e assim por diante.
- Torne os diretórios privados com o **chmod 700 dirname** command.
- Nomeie seus arquivos com nomes auto-explicativos como `Reclamação ao Presidente 050302 ao invés de carta1`.

### Crie espaço

Em alguns sistemas, o sistema **quota** pode forçá-lo a limpá-lo de tempos em tempos ou os limites físicos do seu disco rídico o forçarão a criar mais espaço sem recorrer a programas de monitoramento. Esta seção discute alguns modos, além de usar o comando **rm**, de reaver espaço em disco.

Execute o comando **quota -v** para descobrir quando de espaço ainda resta.

### Esvaziando arquivos

Algumas vezes o conteúdo de um arquivo não lhe interessa, mas é necessário o seu nome como um marcador (por exemplo, você precisa apenas do carimbo de hora, um lembrete de que o arquivo lá estava e lá deverá estar em algum momento futuro). Redirecionando a saída de um comando nulo é o modo de fazê-lo nos shells Bourne e Bash:

```
andy:~> cat wishlist > placeholder
```

```
andy:~> ls -la placeholder
-rw-rw-r-- 1 andy andy      200 Jun 12 13:34 placeholder
```

```
andy:~> > placeholder
```

```
andy:~> ls -la placeholder
-rw-rw-r-- 1 andy andy      0 Jun 12 13:35 placeholder
```

O processo de redução de um arquivo existente para um arquivo com o mesmo nome que tem 0 bytes de tamanho é chamado de *truncar*.

Para criar um novo arquivo vazio, o mesmo efeito pode ser obtido com o **touch**. Em um arquivo existente, o **touch** irá apenas atualizar o carimbo de hora. Veja as páginas **Info** do comando **touch** para mais detalhes.

Para “quase” esvaziar um arquivo, use o comando **tail**. Suponha que a lista de desejos do usuário *andy* se torne muito longa, porque ele sempre adiciona coisas ao seu final, mas nunca apaga as que já

conseguiu. Agora ele quer manter apenas os últimos cinco itens:

```
andy:~> tail -5 wishlist > newlist
```

```
andy:~> cat newlist > wishlist
```

```
andy:~> rm newlist
```

## Mais sobre os arquivos de log

Alguns programas do Linux insistem em escrever todo tipo de saída em um log. Geralmente existem opções para fazer log apenas dos erros ou apenas de uma quantidade mínima de informações, por exemplo configurando o nível de depuração de erros de um programa. Ainda assim, é possível que você não se preocupe com o arquivo log. Seguem alguns meios de se livrar deles ou ao menos estabelecer alguns limites para o seu tamanho:

- Tente remover o arquivo log quando o programa não está sendo executado, se você tiver certeza de que não necessitará dele novamente. Alguns programas podem até mesmo ver, quando reiniciados, que não existe um arquivo de log e portanto não informam log.
- Se você remover o arquivo de log e o programa o recriar, leia a documentação para este programa específico e descubra como evitar que ele crie arquivos de log.
- Tente criar arquivos log menores informando log apenas daquilo que lhe for relevante ou aquilo que for realmente significativo.
- Tente substituir o arquivo log por um link simbólico para `/dev/null`; se você tiver sorte o programa não se queixará. Não faça isso com arquivos log de programas que são executados durante a inicialização do sistema ou programas executador a partir do cron (veja [Chapter 4, Processos](#)). Estes programas podem substituir o link simbólico por um pequeno arquivo que crescerá novamente.

## Email

Limpe regularmente a sua caixa de email, crie subpastas e redirecionamentos automáticos usando o **procmail** (veja as páginas Info) ou os filtros do seu programa de emails favorito. Se você tiver uma lixeira, limpe-a com regularidade.

Para redirecionar emails, use o arquivo `.forward` da sua home. O serviço de email do Linux procura por este arquivo sempre que precisar enviar emails locais. O conteúdo do arquivo determina o que o sistema de emails deve fazer com seus emails. Ele pode conter uma linha única com o endereço de email completo. Neste caso, o sistema enviará todos os emails para tal endereço. Por exemplo, quando estiver alugando espaço para um website, você pode querer encaminhar os emails enviados ao webmaster para sua conta pessoal para não desperdiçar espaço em disco. O `.forward` do webmaster se parecerá com isto:

```
webmaster@www ~/> cat .forward
mike@pandora.be
```

Reencaminhar emails também pode ser útil ao evitar que você precise verificar várias caixas de email diferentes. Pode-se fazer com que todas as contas de email apontem para uma conta central e facilmente acessível.

Você pode pedir ao administrador do seu sistema para que ele defina um endereço para você no arquivo local de apelidos, quando, por exemplo, uma conta for fechada, mas o email permanecer ativo por algum tempo.

## Economize espaço com links

Quando diversos usuários precisam ter acesso ao mesmo arquivo ou programa, quando o nome do arquivo original é muito longo ou difícil de lembrar, use um link simbólico ao invés de criar uma cópia separada para cada usuário ou propósito.

Links simbólicos múltiplos podem ter nomes diferentes, p. ex., o link pode ser chamado `monfichier` no diretório de um usuário e `meulink` em outro. Múltiplos links (diferentes nomes) podem ocorrer inclusive no mesmo diretório. Isto acontece muito no diretório `/lib`: quando é informado o comando

```
ls -l /lib
```

você perceberá que este diretório encontra-se repleto de links que apontam para os mesmos arquivos. Eles são criados para que os programas que procuram por um nome não travem, sendo redirecionados para o nome correto ou atual da biblioteca que precisam.

## Limite o tamanho dos arquivos

O shell possui um comando interno para limitar o tamanho dos arquivos, o **ulimit**, que também pode ser usado para exibir as limitações nos recursos do sistema:

```
cindy:~> ulimit -a
core file size (blocks)      0
data seg size (kbytes)      unlimited
```

file size (blocks)	unlimited
max locked memory (kbytes)	unlimited
max memory size (kbytes)	unlimited
open files	1024
pipe size (512 bytes)	8
stack size (kbytes)	8192
cpu time (seconds)	unlimited
max user processes	512
virtual memory (kbytes)	unlimited

Cindy não é uma desenvolvedora e não se importa com descargas de memória, que contém informações sobre a depuração de um programa. Se você quer realizar descargas de memória, é possível estabelecer seu tamanho usando o comando **ulimit**. Leia as páginas Info do **bash** para uma explicação detalhada.

## Arquivo núcleo?

Um arquivo núcleo ou *descarga de memória* é criado quando algo de errado acontece com um programa ao longo de sua execução. Os arquivos núcleo contém uma cópia da memória do sistema, tal como se encontrava quando o erro aconteceu.

## Arquivos compactados

Arquivos compactos são úteis porque ocupam menos espaço no seu disco rígido. Outra vantagem é que os arquivos compactos consomem menos banda para serem transmitidos, carregando menos a sua rede. Muitos arquivos, como as páginas man, encontram-se arquivados em formatos compactos no seu sistema. Porém, ter que descompactá-los para conseguir uma pequena informação e então ter que compactá-lo novamente é algo que consome muito tempo. Você não vai querer, p. ex., descompactar uma página man, ler sobre uma opção para um comando e então compactá-la novamente. A maior parte das pessoas se esquece de apagar os arquivos descompactados após conseguirem a informação que precisavam.

Por este motivos existem ferramentas que trabalham os arquivos compactados, descompactando-os apenas em memória. O arquivo compacto permanece intocado no seu disco. A maior parte dos sistemas aceita o **zgrep**, o **zcat**, o **bzless** e muitos outros membros da família z, de forma a evitar ações de compressão e descompressão desnecessárias. Veja o diretório binário do seu sistema e as páginas Info.

Veja o [Chapter 9, Técnicas Fundamentais de Backup](#) para saber mais a respeito da compressão de arquivos e exemplos de como arquivá-los.

## Seu ambiente texto

### Variáveis do ambiente

#### Geral

Já mencionamos algumas variáveis do ambiente, como a PATH e a HOME. Até agora, vimos apenas exemplos em que elas servem para um propósito específico do shell. Existem, porém, muitos outros utilitários do Linux que precisam de informações a seu respeito para que realizem um bom trabalho.

Quais outras informações os programas necessitam além das paths e diretórios da home?

Muitos programas querem saber qual tipo de terminal você está usando; esta informação encontra-se registrada na variável TERM. No modo texto, esta será a emulação do terminal *linux*, no modo gráfico provavelmente você usará o *xterm*. Muitos programas querem saber qual o seu editor favorito, caso precisem iniciar um editor em um subprocesso. O shell que você está usando encontra-se gravado na variável SHELL, o tipo de sistema operacional no OS e assim em diante. Uma lista das variáveis atualmente definidas pode ser vista informando-se o comando **printenv**.

As variáveis do ambiente são gerenciadas pelo shell. Ao contrário do que ocorre com as variáveis do shell, as variáveis do ambiente são herdadas por quaisquer programas iniciados, inclusive outros shells. Novos processos recebem uma cópia destas variáveis, as quais podem ler, modificar e passar a diante para os processos filhos.

Não há nada de especial nos nomes das variáveis, a não ser o fato de que as mais comuns, por convenção, são dadas em maiúsculas. Você pode pensar em qualquer nome, embora existam variáveis padrão que são importantes o suficiente para aparecerem em todos os sistemas Linux, como, por exemplo, as variáveis PATH e HOME.

### Exportando variáveis

O conteúdo de uma variável em geral é exibido usando o comando **echo**, como nos exemplos a seguir:

```
debby:~> echo $PATH
/usr/bin:/usr/sbin:/bin:/sbin:/usr/X11R6/bin:/usr/local/bin
```

```
debby:~> echo $MANPATH
/usr/man:/usr/share/man:/usr/local/man:/usr/X11R6/man
```

Se você deseja mudar o conteúdo de uma variável de uma forma que seja útil a outros programas, é

preciso exportar o novo valor do seu ambiente para aquele em que estes programas são executados. Um exemplo comum é exportar a variável PATH. Você pode declará-la como a seguir, de forma a poder jogar o software de simulação de vôo que esta no /opt/FlightGear/bin:

```
debby:~> PATH=$PATH:/opt/FlightGear/bin
```

Isto instrui o shell a procurar estes programas não apenas no path atual, \$PATH, mas também no diretório /opt/FlightGear/bin.

Entretanto, enquanto o novo valor da variável PATH for desconhecido do seu ambiente, ele continuará a não funcionar:

```
debby:~> runfgfs
bash: runfgfs: command not found
```

A exportação de variáveis é feita usando o comando interno do shell **export**:

```
debby:~> export PATH
debby:~> runfgfs
--flight simulator starts--
```

No Bash, isto normalmente é feito em um passo elegante:

```
export VARIABLE=value
```

A mesma técnica é usada para a variável MANPATH, que informa o comando **man** onde procurar pelas páginas man. Se novos softwares forem adicionados ao sistema em diretórios novos ou não-usuais, a sua documentação provavelmente também será alocada nestes diretórios. Se você quiser ler as páginas man para o novo software, amplie a variável MANPATH:

```
debby:~> export MANPATH=$MANPATH:/opt/FlightGear/man
debby:~> echo $MANPATH
/usr/man:/usr/share/man:/usr/local/man:/usr/X11R6/man:/opt/FlightGear/man
```

Você pode evitar ter que redigitar este comando em todas as janelas que abrir adicionando-a a um dos arquivos de configuração do shell, veja o [the section called "Arquivos de configuração do shell"](#).

## Variáveis reservadas

A tabela a seguir fornece um panorama das variáveis pré-definidas mais comuns:

**Table 7.1. Variáveis de ambiente comuns**

Nome da variável	Informação armazenada
DISPLAY	é usada no sistema de janelas X para identificar o servidor de exibição
DOMAIN	nome de domínio
EDITOR	armazena o seu editor de linha favorito
HISTSIZE	tamanho, em número de linhas, do arquivo de histórico do shell
HOME	caminho para o seu diretório home
HOSTNAME	nome da máquina local
INPUTRC	local do arquivo de definição dos aparelhos de entrada, como o teclado
LANG	idioma de preferência
LD_LIBRARY_PATH	caminho para procurar por bibliotecas
LOGNAME	nome do login
MAIL	local da sua pasta de entrada de emails
MANPATH	caminho de busca por páginas man
OS	seqüência com a descrição do sistema operacional
OSTYPE	mais informações sobre a versão e etc.
PAGER	usada por programas como <b>man</b> que precisam saber o que fazer caso a saída seja maior do que uma janela de terminal.
PATH	caminho de busca por comandos
PS1	prompt primário
PS2	prompt secundário
PWD	diretório atual
SHELL	shell atual
TERM	tipo de terminal
UID	ID do usuário
USER(NAME)	nome de usuário
VISUAL	seu editor de tela cheia favorito



XENVIRONMENT	local das suas configurações pessoais para o comportamento do X
XFILESEARCHPATH	caminho para busca por bibliotecas gráficas

Muitas variáveis não são apenas pré-definidas, mas também pré-configuradas, usando arquivos de configuração. Nós os discutimos na próxima seção.

## Arquivos de configuração do shell

Quando é informado o comando **ls -al** para visualizar uma lista detalhada de todos os arquivos, incluindo aqueles que começam com um ponto, no seu diretório home, você verá um ou mais arquivos iniciados por um **.** e terminados por **rc**. No caso do **bash**, este arquivo é o **.bashrc**. Este é a contraparte do arquivo de configuração **/etc/bashrc**, válido para todo o sistema.

Quando é informado o comando **ls -al** para visualizar uma lista detalhada de todos os arquivos, incluindo aqueles que começam com um ponto, no seu diretório home, você verá um ou mais arquivos iniciados por um **.** e terminados por **rc**. No caso do **bash**, este arquivo é o **.bashrc**. Este é a contraparte do arquivo de configuração **/etc/bashrc**, válido para todo o sistema.

Quando uma shell de login é finalizada, o **bash** lê e executa os comandos do arquivo **~/.bash\_logout**, se existir.

Este procedimento é explicado em detalhes nas páginas man do **login** e do **bash**.

## Um conjunto típico de arquivos de configuração

### Exemplo **/etc/profile**

Vejamos alguns destes arquivos de configuração. Primeiro o **/etc/profile** é lido, no qual importantes variáveis como **PATH**, **USER** e **HOSTNAME** estão armazenadas:

```
debby:~> cat /etc/profile
# /etc/profile

# System wide environment and startup programs, for login setup
# Functions and aliases go in /etc/bashrc

# Path manipulation
if [ `id -u` = 0 ] && ! echo $PATH | /bin/grep -q "/sbin" ; then
    PATH=/sbin:$PATH
fi

if [ `id -u` = 0 ] && ! echo $PATH | /bin/grep -q "/usr/sbin" ; then
    PATH=/usr/sbin:$PATH
fi

if [ `id -u` = 0 ] && ! echo $PATH | /bin/grep -q "/usr/local/sbin"
    then
    PATH=/usr/local/sbin:$PATH
fi

if ! echo $PATH | /bin/grep -q "/usr/X11R6/bin" ; then
    PATH="$PATH:/usr/X11R6/bin"
fi
```

Estas linhas checam o caminho a ser configurado: se o **root** abre um shell (ID de usuário 0), confere-se que **/sbin**, **/usr/sbin** e **/usr/local/sbin** estão no caminho. Caso contrário, eles são adicionados. Verifica-se para todos que **/usr/X11R6/bin** está no caminho.

```
# No core files by default
ulimit -S -c 0 > /dev/null 2>&1
```

Todo lixo vai para **/dev/null** se o usuário não mudar esta configuração.

```
USER=`id -un`
LOGNAME=$USER
MAIL="/var/spool/mail/$USER"
```

```
HOSTNAME=`/bin/hostname`
HISTSIZE=1000
```

Aqui as variáveis comuns recebem seus valores normais.

```
if [ -z "$INPUTRC" -a ! -f "$HOME/.inputrc" ]; then
    INPUTRC=/etc/inputrc
fi
```

Se a variável **INPUTRC** não estiver definida e não existir o **.inputrc** no diretório home do usuário, então o arquivo de controle padrão é carregado.

```
export PATH USER LOGNAME MAIL HOSTNAME HISTSIZE INPUTRC
```

Todas as variáveis são exportadas, de forma a ficarem disponíveis para outros programas que peçam informações sobre o seu ambiente.

## O diretório profile.d

```
for i in /etc/profile.d/*.sh ; do
    if [ -r $i ]; then
        . $i
    fi
done
unset i
```

Todos os scrips legíveis do shell do diretório `/etc/profile.d` são lidos e executados. Eles realizam tarefas como habilitar o *color-ls*, vincular o **vi** ao **vim**, configurar locais e etc. A variável temporária `i` é desabilitada para prevenir que interfira no comportamento do shell.

## Exemplo de .bash\_profile

Então o **bash** procura por um `.bash_profile` no diretório home do usuário:

```
debby:~> cat .bash_profile
#####
#
# .bash_profile file
#
# Executed from the bash shell when you log in.
#
#####

source ~/.bashrc
source ~/.bash_login
```

Este arquivo, muito direto, instrui o seu shell a ler primeiramente `~/.bashrc` e então o `~/.bash_login`. Você encontrará o comando interno do shell **source** regularmente, quando estiver trabalhando em um ambiente shell: ele costumava aplicar as mudanças de configuração ao ambiente atual.

## Exemplo de .bash\_login

O arquivo `~/.bash_login` define a proteção de arquivo padrão, configurando o valor do **umask**, veja o [the section called "Efetuando o login em outro grupo"](#). O arquivo `~/.bashrc` é usado para definir diversos aliases e funções específicas de usuário, bem como variáveis de ambiente pessoal. Primeiramente ele lê `/etc/bashrc`, que descreve o prompt padrão (PS1) e o valor do `umask`. Em seguida, é possível adicionar suas configurações pessoais. Se não existir o `~/.bashrc`, por padrão, o `/etc/bashrc` é lido.

## Exemplo de /etc/bashrc

O seu arquivo `/etc/bashrc` deve parecer-se com:

```
debby:~> cat /etc/bashrc
# /etc/bashrc

# System wide functions and aliases
# Environment stuff goes in /etc/profile

# by default, we want this to get set.
# Even for non-interactive, non-login shells.
if [ `id -gn` = `id -un` -a `id -u` -gt 99 ]; then
    umask 002
else
    umask 022
fi
```

Estas linhas configuram o valor do **umask**. Então, dependendo do tipo de shell, o prompt é configurado:

```
# are we an interactive shell?
if [ "$PS1" ]; then
    if [ -x /usr/bin/tput ]; then
        if [ "x`tput kbs`" != "x" ]; then
# We can't do this with "dumb" terminal
            stty erase `tput kbs`
        elif [ -x /usr/bin/wc ]; then
            if [ "`tput kbs|wc -c`" -gt 0 ]; then
# We can't do this with "dumb" terminal
                stty erase `tput kbs`
            fi
        fi
    fi
case $TERM in
    xterm*)
        if [ -e /etc/sysconfig/bash-prompt-xterm ]; then
            PROMPT_COMMAND=/etc/sysconfig/bash-prompt-xterm
        else
            PROMPT_COMMAND='echo -ne "\033]0;${USER}@${HOSTNAME%.*}:\'
```

```

${PWD/$HOME/~}\007"
    fi
;;
*)
[ -e /etc/sysconfig/bash-prompt-default ] && PROMPT_COMMAND=\
/etc/sysconfig/bash-prompt-default
;;
esac
[ "$PS1" = "\\s-\\v\\\$ " ] && PS1="\u@\h \W\\\$ "

if [ "x$SHLVL" != "x1" ]; then # We're not a login shell
    for i in /etc/profile.d/*.sh; do
        if [ -x $i ]; then
            . $i
        fi
    done
fi
fi

```

## Exemplo de .bash\_logout

No logout, os comandos no `~/.bash_logout` são executados, que podem por exemplo limpar o terminal, de forma a se ter uma janela limpa quando se realiza o logout de uma sessão remota ou ao se deixar o console do sistema:

```

debby:~> cat .bash_logout
# ~/.bash_logout

```

```
clear
```

Na próxima seção, vamos dar uma olhada mais de perto para ver como estes scripts funcionam. Mantenha o **info** *bash* a mão.

## O prompt do Bash

### Introdução

O prompt do Bash pode fazer muito mais do que exibir informações simples como o seu nome de usuário, o nome da sua máquina e alguma indicação do diretório atual. Pode-se adicionar informações a respeito da data e hora atuais, número de usuários conectados e etc.

Antes de começarmos, entretanto, salvaremos o nosso prompt atual em outra variável de ambiente:

```

[jerry@nowhere jerry]$ MYPROMPT=$PS1

[jerry@nowhere jerry]$ echo $MYPROMPT
[\u@\h \W]\$

[jerry@nowhere jerry]$

```

Agora, quando mudarmos o prompt, por exemplo, informando o comando `PS1="->"`, poderemos a qualquer tempo ter nosso prompt de volta com o comando `PS1=$MYPROMPT`. Você irá, é claro, tê-lo de volta ao reconectar, desde que você mexa apenas com o prompt na linha de comando e evite colocar isto em um arquivo de configuração do shell.

### Alguns exemplos

Para entender estes prompts e as seqüências de escape usadas, procure nas páginas [man](#) ou [Info](#).

- **export PS1="[t j] "**  
Exibe a hora do dia e o número de processos em execução
- **export PS1="[d][u@h w] : "**  
Exibe a data, nome de usuário, nome da máquina e diretório atual. Perceba que `\W` exibe apenas o nome básico do diretório atual.
- **export PS1="{!} "**  
Exibe o número do histórico para cada comando.
- **export PS1="\[\033[1;35m\]\u@\h\[\033[0m\] "**  
Exibe usuário@máquina em rosa.
- **export PS1="\[\033[1;35m\]\u\[\033[0m\] \[\033[1;34m\]\w\[\033[0m\] "**  
Configura o nome do usuário em rosa e o diretório atual em azul.
- **export PS1="\[\033[1;44m\]\$USER is in \w\[\033[0m\] "**

Prompt para pessoas que têm dificuldade em ver a diferença entre o prompt e aquilo que eles digitam.

- **export PS1="\[\033[4;34m\]\u@\h \w \[\033[0m\]"**

Prompt sublinhado.

- **export PS1="\[\033[7;34m\]\u@\h \w \[\033[0m\] "**

Caracteres azuis em um fundo de tela azul.

- **export PS1="\[\033[3;35m\]\u@\h \w \[\033[0m\]\a"**

Prompt rosa em uma fonte mais leve que avisa quando os seus comando tiverem terminado.

- **export PS1=...**

Variáveis são exportadas de forma que comandos executados em seqüência também saibam sobre o seu ambiente. O ideal é que a linha de configuração do prompt que você fique no arquivo de configuração do seu shell, `~/.bashrc`.

Se você quiser, os prompts podem executar scripts do shell e se comportar de maneiras diferentes de acordo com a situação. Você pode fazer com que o prompt toque um som toda vez que você informar um comando, embora isto se torne um tanto quanto chato em pouquíssimo tempo. Mais informações podem ser encontradas no [Bash-Prompt HOWTO](#).

## Scripts do shell

### O que são scripts?

Um script do shell é, como vimos nos exemplos de configuração do shell, um arquivo texto contendo comandos do shell. Quando um arquivo destes é usado como o primeiro argumento não-opção ao se invocar o Bash, e quer a opção `-c` quer a `-s` são informadas, o Bash lê e executa as opções do arquivo e então sai. Este modo de operação cria um shell não-interativo. Quando o Bash executa um script do shell, ele configura o parâmetro especial `0` ao nome do arquivo, ao invés do nome do shell, e os parâmetros de posição (tudo aquilo que segue o nome do script) são configurados aos argumentos restantes, se quaisquer forem informados. Caso contrário, os parâmetros de posição não são configurados.

Pode-se tornar um script do shell executável informando o comando **chmod** para ligar o bit de execução. Quando o Bash encontra um destes arquivos enquanto procura no PATH por um comando, ele invoca um sub-shell para executá-lo. Em outras palavras, executar

```
filename ARGUMENTS
```

é equivalente a executar

```
bash filename ARGUMENTS
```

se "nomedearquivo" é um script executável do shell. Este sub-shell se reinicia, de tal forma que o efeito é o mesmo daquele que ocorreria caso fosse invocado um novo shell para interpretar o script, com a exceção de que a localização dos comandos lembrados pelo pai (veja páginas Info para o **hash**) é mantida pelo filho.

A maior parte das versões do UNIX tornam isto parte do mecanismo de execução de comandos do sistema. Se a primeira linha de um script começa com os caracteres "#!", a sua continuação da linha especifica um intérprete para o programa. Portanto, você pode especificar **bash**, **awk**, **perl** ou algum outro intérprete ou shell e escrever o restante do arquivo script naquela língua.

Os argumentos para o intérprete consistem em um único argumento opcional imediatamente após o nome do intérprete na primeira linha do arquivo de script, seguido do nome do arquivo e, após, pelo restante dos argumentos. O Bash realiza esta operação nos sistemas operacionais que não conseguem fazê-lo por conta própria.

Os scripts do Bash em geral começam com

```
#!/bin/bash
```

(assumindo-se que o Bash foi instalado no `/bin`), vez que isto assegura que o Bash será usado para interpretar o script, mesmo que tenha que ser executado em outro shell.

### Alguns exemplos simples

Um script muito simples, de apenas um comando, que diz olá para o usuário que o executar:

```
[jerry@nowhere ~] cat hello.sh
#!/bin/bash
echo "Hello $USER"
```

O script consiste, na verdade, de apenas um comando, **echo**, que usa o valor de *valor de* (\$) a variável USER do ambiente para imprimir uma seqüência personalizada pelo usuário que informa o comando.

Outra de apenas uma linha, usada para exibir os usuário conectados:

```
#!/bin/bash
who | cut -d " " -f 1 | sort -u
```

Aqui está um script com mais algumas linhas, que eu uso para fazer cópias de backup de todos os arquivos em um diretório. O script primeiro cria uma lista de todos os arquivos no diretório atual e coloca na variável `LISTA`. Em seguida ele configura o nome da cópia para cada arquivo e, então, copia o arquivo. Para cada arquivo, uma mensagem é exibida:

```
tille:~> cat bin/makebackupfiles.sh
#!/bin/bash
# make copies of all files in a directory
LIST=`ls`
for i in $LIST; do
    ORIG=$i
    DEST=$i.old
    cp $ORIG $DEST
    echo "copied $i"
done
```

Apenas informar uma linha como `mv *.old` não funciona, como você perceberá quando tentar fazê-lo em um conjunto de arquivos teste. Um comando `echo` foi adicionado para exibir alguma atividade. Os `echo` em geral são úteis quando um comando não funciona: insira um após cada passo dúbio que você logo encontrará o erro.

O diretório `/etc/rc.d/init.d` contém muitos exemplos. Olhemos este script que controla o servidor fictício `EuPossoVeLo`:

```
#!/bin/sh
# descrição: EuPossoVeLo permite visualizar as pessoas na rede

# process name: EuPossoVeLo
# pidfile: /var/run/EuPossoVeLo/EuPossoVeLo.pid
# config: /etc/EuPossoVeLo.cfg

# Biblioteca fonte das funções.
. /etc/rc.d/init.d/functions

# Veja como (com quais argumentos) nós fomos chamados.
case \"$1\" in
    start)
        echo -n "Iniciando EuPossoVeLo: "
        daemon EuPossoVeLo
        echo
        touch /var/lock/subsys/EuPossoVeLo
        ;;
    stop)
        echo -n "Finalizando EuPossoVeLo: "
        killproc EuPossoVeLo
        echo
        rm -f /var/lock/subsys/EuPossoVeLo
        rm -f /var/run/EuPossoVeLo/EuPossoVeLo.pid
        ;;
    status)
        status EuPossoVeLo
        ;;
    restart)
        $0 stop
        $0 start
        ;;
    *)\n"
        echo "Usage: $0 {start|stop|restart|status}"
        exit 1
esac

exit 0
```

Primeiramente, com o comando `.` (ponto) um conjunto de funções do shell, usados por quase todos os scripts do shell no `/etc/rc.d/init.d`, é carregado. Então, um comando `case` é informado, definindo 4 diferentes modos de execução do script. Um exemplo pode ser `EuPossoVeLo start`. A decisão de qual caso aplicar é feita através da leitura do (primeiro) argumento ao script, com a expressão `$1`.

Quando nenhuma entrada de conformação for informada, o caso padrão, marcado com um asterisco, para o qual o script apresenta uma mensagem de erro. A lista do `case` termina com o enunciado `esac`. No caso do `start` o programa servidor é iniciado como um daemon, recebendo uma ID de processo e cadeado. No caso do `stop`, o processo do servidor é rastreado e finalizado, removendo-se seus bloqueio e ID do processo. Opções, como `daemon` e funções como `killproc` são definidas no arquivo `/etc/rc.d/init.d/functions`. Este arquivo de configuração é específico da distribuição usada neste exemplo. Os initscripts do seu sistema podem usar outras funções, definidas em outros arquivos ou mesmo nenhuma.

Em caso de sucesso, o script retorna um código de saída de zero ao seu pai.

Este script é um ótimo exemplo de uso de funções, o que torna o script fácil de ler e agiliza o trabalho. Perceba que eles usam o **sh** ao invés do **bash**, de forma a torná-los úteis em uma maior gama de sistemas. Em um sistema Linux, chamar o **bash** como **sh** resulta em um shell executado em conformidade com o POSIX.

As páginas man do **bash** contém mais informações sobre a combinação de comandos, os loops e expressões regulares do **for**- e do **while**-, bem como alguns exemplos. Um curso completo de Bash para administradores de sistemas e usuários pró, com exercícios, da mesma autora deste Guia de Introdução ao Linux, pode ser encontrado no <http://tille.garrels.be/training/bash/>. Uma explicação detalhada dos recursos do Bash pode ser encontrada no guia de referência [Advanced Bash Scripting](#).

## O ambiente gráfico

### Introdução

O usuário médio pode não se importar muito com as suas configurações de login, mas o Linux oferece uma imensa variedade de gerenciadores de desktop e janelas para serem usados sob o X, o ambiente gráfico. O uso e a configurações dos gerenciadores de janelas e desktops é intuitivo e pode até mesmo guardar semelhanças como os padrões MS Windows, Apple ou UNIX CDE, embora muitos usuários Linux prefiram desktop mais brilhantes e gerenciadores de janelas mais extravagantes. Não discutiremos aqui as configurações específicas de usuário. Experimente. Leia as funções internas de Ajuda destes gerenciadores que você conseguirá se virar sozinho.

Nós iremos, contudo, dar uma olhada mais de perto no sistema por trás.

### O Sistema de Janelas X

O Sistema de Janelas X é um sistema de janelas em redes transparentes que funciona em uma larga gama de máquinas gráficas e de computação. O sistema X funciona em servidores com capacidade de exibir em bitmap. O servidor X aceita entrada do usuário e saídas de diversos programas clientes, por meio de uma variedade grande de canais de comunicação. Embora o caso mais comum seja que os programas cliente rodem na mesma máquina que o servidor, eles podem também ser executados em outras máquinas (incluindo máquinas cuja arquitetura e sistema operacional seja distintos do servidor) de forma transparente. Você aprenderá como fazê-lo no [Chapter 10, Redes](#) que trata de redes e aplicativos remotos.

O X é capaz de sobrepor janelas em hierarquia, bem como texto e operações gráficas, tanto em displays coloridos quanto em monocromáticos. O número de clientes X que usam o servidor X é muito grande. Alguns dos programas que são fornecidos no núcleo do Consórcio X são:

- **xterm**: um emulador de terminal
- **twm**: um gerenciador de janelas minimalista
- **xdm**: um gerenciador de display
- **xconsole**: um programa de redirecionamento de console
- **bitmap**: um editor bitmap
- **xauth**, **xhost** e **iceauth**: programas de controle de acesso
- **xset**, **xmodmap** e outros: programas de configuração de preferências de usuário
- **xclock**: um relógio
- **xlsfonts** e outros: um exibidor de fontes, utilitários para listar informações sobre fontes, janelas e display.
- **xfst**: um servidor de fontes
- ...

Indicamos, novamente, a leitura das páginas man destes comandos para obter mais detalhes. Mais explicações sobre as funções disponíveis podem ser encontradas no manual *Xlib - C language X Interface* que vem com a sua distribuição do X, as especificações do *Protocolo do Sistema de Janelas X* e os vários manuais e documentos sobre as ferramentas do X. O diretório `/usr/share/doc` contém referência a estes e muitos outros documentos.

Muitos outros utilitários, gerenciadores de janelas, jogos, ferramentas e dispositivos encontram-se inclusos como contribuição dos usuários à distribuição do Consórcio X ou podem ser encontradas usando FTPs anônimos na Internet. Bons lugares para se começar são <http://www.x.org> e <http://www.xfree.org>.

Mais ainda, todos os seus aplicativos gráficos, como o seu navegador, seu programa de emails, seus aplicativos de exibição de imagens, ferramentas de execução de som e assim por diante são todos clientes do seu servidor X. Perceba que, no Linux, em uma operação normal, ou seja, no modo gráfico, os clientes X e o servidor X são executados na mesma máquina.

## Nomes de display

A partir da perspectiva do usuário, todo servidor X tem um *nome de display* na forma de:

```
hostname:displaynumber.screennumber
```

Estas informações são usadas pelos aplicativos para determinar como ele deve se conectar ao servidor X e qual tela deve usar por padrão (quando houver múltiplos monitores):

- *nomedamáquina*: O nome da máquina especifica qual o nome da máquina cliente, ao qual o display encontra-se fisicamente ligado. Se o nome da máquina não for informado, será usado o meio mais eficiente de comunicar-se com um servidor na mesma máquina.
- *númerodisplay*: A palavra “display” em geral é usada pra fazer referência a uma coleção de monitores que compartilham os mesmos teclado e ponteiro (mouse, mesa digitalizadora, etc.). A maior parte das estações de trabalho tende a ter apenas um teclado e, portanto, um display. Sistemas maiores, de vários usuários, porém, freqüentemente têm diversos displays de modo que mais de uma pessoa possa realizar trabalhos gráficos a um só tempo. Para evitar confusão, é dado um *númerodedisplay* (começando por 0) a cada display em uma máquina, quando se inicia o servidor X para aquele display. O número do display deve ser sempre informado junto ao seu nome.
- *númerodetela*: Alguns displays compartilham um único teclado e ponteiro entre dois ou mais monitores. Uma vez que cada monitor tem seu próprio conjunto de janelas, cada tela recebe seu próprio *númerodetela* (iniciado em 0), quando se inicia o servidor para aquele display. Se o número da tela não for informado, será usado o 0.

Em sistemas POSIX, o nome de display padrão fica armazenado na sua variável `DISPLAY`. Esta variável é configurada automaticamente pelo emulador de terminal **xterm**. Entretanto, quando você efetuar o login em outra máquina ou rede, é possível que tenha que configurar manualmente a variável `DISPLAY` para indicar o seu display, veja [the section called “Telnet e X”](#).

As páginas man do X fornecem mais informações.

## Gerenciadores de janela e desktop

O formato das janelas na tela é controlado por programas especiais chamados *gerenciadores de janelas*. Embora muitos gerenciadores de janelas honrem as especificações geométricas tais como *dadas*, outros podem escolher ignorá-las (forçando o usuário, por exemplo, a desenhar a região das janelas na tela com o ponteiro).

Uma vez que os gerenciadores de janelas são programas clientes ordinários (ainda que complexos), uma grande variedade de interfaces de usuário podem ser construídas. A distribuição do Consórcio X traz um gerenciador de janelas chamado **twm**, mas a maior parte dos usuários prefere outros mais requintados, quando os recursos de seus sistemas permitem. Sawfish e Enlightenment são exemplos populares que permitem ao usuário ter um desktop de acordo com seu estilo e humor.

Um gerenciador de desktop faz uso de um gerenciador de janelas para convenientemente organizar o seu desktop gráfico com barras de menu, menus de seleção, mensagens de informação, um relógio, um gerenciador de programas, um gerenciador de arquivos e assim por diante. Dentre os gerenciadores de desktop mais populares pode-se mencionar o Gnome e o KDE, os quais rodam em quase todas as distribuições do Linux, bem como vários outros sistemas UNIX.

## Aplicativos do KDE no Gnome e aplicativos Gnome no KDE

Não é necessário iniciar o seu desktop no KDE para conseguir executar seus aplicativos. Se as bibliotecas do KDE estiverem instaladas (o pacote `kdelibs`), é possível executar seus aplicativos a partir dos menus do Gnome ou de um terminal.

Executar aplicativos Gnome em um ambiente KDE é um pouco mais delicado, porquanto inexiste um único jogo de bibliotecas básicas do Gnome. Todavia, as dependências e os pacotes adicionais que será preciso instalar ficarão claros quando se instalar ou executar o aplicativo.

## Configuração do servidor X

A distribuição do X que costumava vir com o Linux, o *XFree86*, usa o arquivo `XF86Config` para sua configuração inicial. Este arquivo configura seu cartão de vídeo e é procurado em diversos locais, embora normalmente possa ser encontrado no `/etc/X11`.

Se você perceber que o arquivo `/etc/X11/XF86Config` está presente no seu sistema, uma descrição completa pode ser encontrada nas páginas man ou Info sobre o `XF86Config`.

Por conta de questões relacionadas à licença do *XFree86*, sistemas mais recentes em geral vêm com a distribuição *X.Org* do servidor e ferramentas do X. Neste, o principal arquivo de configuração é o `xorg.conf`, que normalmente também pode ser encontrado no `/etc/X11`. Este arquivo é formado por um número de seções que podem aparecer em qualquer ordem. As seções contém informações sobre o seu monitor, seu adaptador de vídeo, a configuração de tela, seu teclado e etc. Enquanto usuário não é necessário se preocupar muito com o conteúdo deste arquivo, vez que tudo isto é configurado quando da instalação do sistema.

Entretanto, caso você precise mudar a configuração do servidor gráfico, é possível usar as ferramentas de configuração ou editar os arquivos que compõem a infra-estrutura do servidor XFree86. Procure nas páginas man por mais informações; é possível que a sua distribuição conte com ferramentas próprias. Uma vez que um erro de configuração pode resultar em falhas que tornem o modo gráfico ininteligível, é aconselhável que você faça uma cópia do arquivo de configuração antes de mudá-lo, por questão de segurança.

## Configurações de Região

### Configuração de teclado

Configura-se o layout do teclado através do comando **loadkeys** para consoles no modo texto. Use a sua ferramenta X local ou edite manualmente a seção *Keyboard* do `XF86Config` para configurar o layout no modo gráfico. Você deve procurar pela variável `XkbLayout`:

```
XkbLayout      "us"
```

Este é o padrão. Mude-o para a configuração de sua região alterando o valor entre aspas por quaisquer outros nomes listados nos subdiretórios do seu diretório `keymaps`. Se não conseguir encontrar o `keymaps`, tente exibir sua localização no seu sistema informando o comando:

```
locate keymaps
```

É possível combinar diferentes configurações, como neste exemplo:

```
Xkblayout      "us,ru"
```

Crie um backup do arquivo `/etc/X11/XF86Config` antes de começar a editá-lo! Será necessário usar a conta `root` para fazê-lo.

Saia do sistema e reconecte para recarregar as configurações do X.

O Gnome Keyboard Applet permite a alteração imediata de layouts; não são necessárias permissões especiais para usar este programa. O KDE possui uma ferramenta similar para mudar o layout do teclado.

### Fontes

Use a ferramenta **setfont** para carregar fontes no modo texto. A maior parte dos sistemas vem com um arquivo `inputrc` que permite a combinação de caracteres, como o “é” do Português (meta-caracteres). O administrador do sistema deve então adicionar a linha

```
export INPUTRC="/etc/inputrc"
```

ao arquivo `/etc/bashrc`

### Data e Fuso Horário

A configuração das informações de tempo é geralmente feito quando da instalação. Após, é possível mantê-la atualizada usando um cliente *NTP* (Network Time Protocol - Protocolo de Tempo e Rede). A maior parte dos sistemas Linux padroniza a execução do **ntpd**:

```
debby:~> ps -ef | grep ntpd
ntp      24678      1  0  2002 ?          00:00:33 ntpd -U ntp
```

É possível executar o **ntpdate** manualmente para configurar o tempo, desde que você possa acessar o servidor de tempo. O daemon **ntpd** não deve estar em execução quando você ajustar o tempo usando o **ntpdate**. Use um servidor de tempo como argumento para o comando:

```
root@box:~# ntpdate 10.2.5.200
26 Oct 14:35:42 ntpdate[20364]: adjust time server 10.2.5.200 offset
-0.008049 sec
```

Procure no manual do seu sistema e na documentação que vem com o pacote NTP. A maior parte dos gerenciadores de desktop inclui ferramentas para configurar a hora do sistema, desde que se tenha à conta administrativa.

Para configurar o fuso horário correto, pode-se fazer uso dos comandos **tzconfig** ou **timezone**. A configuração do fuso horário em geral é feita durante a instalação do sistema. Muitos sistemas possuem ferramentas específicas para configurá-lo, procure na documentação de sua distribuição.

### Idioma

Se você preferir o sistema informe mensagens em Português ou Francês, você deve configurar as variáveis de ambiente `LANG` e `LANGUAGE`, de forma a habilitar o suporte local ao idioma escolhido e, eventualmente, as convenções e caracteres neste idioma.

Com a maior parte dos logins gráficos, como o **gdm** ou o **kdm**, é possível configurar estas preferências de idioma antes de efetuar o login.

Repare que na maior parte dos sistemas, o padrão atualmente tende a ser `en_US.UTF-8`. Isto não é um



problema, porque nos sistemas onde isto é o padrão também vêm com todos os programas que dão suporte a este tipo de codificação. Portanto, o **vi** poderá editar todos os arquivos do seu sistema e o **cat** não se comportará de um modo inusitado.

O problema começa quando você se conecta a um sistema antigo que não tem suporte para esta codificação de fonte ou quando se abre um arquivo codificado em *UTF-8* em um sistema preparado apenas para caracteres de 1 byte de extensão. O utilitário **recode** pode ser de grande ajuda para a conversão de um conjunto de caracteres para outro. Leia as páginas man para ver um panorama dos recursos e usos. Outra solução pode ser trabalhar temporariamente com outra codificação, alterando a variável de ambiente `LANG`.

```
debby:~> acroread /var/tmp/51434s.pdf  
Warning: charset "UTF-8" not supported, using "ISO8859-1".  
Aborted
```

```
debby:~> set | grep UTF  
LANG=en_US.UTF-8
```

```
debby:~> export LANG=en_US
```

```
debby:~> acroread /var/tmp/51434s.pdf  
<--new window opens-->
```

Para saber mais, procure no site do [Mozilla](#) e descubra como ter o Firefox no seu idioma. O site [OpenOffice.org](#) provê mais informações sobre a localização da sua suíte OpenOffice.org.

## Informações específicas do país

O [list of HOWTOs](#) contém referências às instruções de localidade Bangla, Bielorrussa, Chinesa, em Esperanto, Finlandês, Francófonas, Hebraica, Helênica, Polonesa, Portuguesa, Sérvia, Eslovaca, Eslovena, Espanhola, Tailandesa e Turca.

## Instalando novos programas

### Geral

A maior parte das pessoas fica surpresa em ver que têm um sistema usável e em plena execução após instalarem o Linux; a maior parte das distribuições contém amplo suporte para cartões de rede e vídeo, monitores e outros dispositivos externos, de forma que não se necessite da instalação de drivers extras. Ferramentas comuns como suítes de escritório, navegadores web, email e outros clientes de rede estão inclusas nas principais distribuições. Mesmo assim, é possível que a instalação inicial não supra todas as suas necessidades.

Se você não conseguir encontrar aquilo que precisa, provavelmente é porque não está instalado no seu sistema. Pode também acontecer que você possua o software desejado, porém ele não faça aquilo que deveria. Lembre-se que o Linux cresce rápido e que os softwares são aperfeiçoados a cada dia. Não perca tempo reportando problemas que já se encontram resolvidos.

Você pode atualizar seu sistema e adicionar novos pacotes a qualquer tempo. A maior parte dos softwares encontra-se disponível em pacotes. Softwares adicionais podem ser encontrados no seu CD de instalação ou na Internet. O site da sua distribuição é um bom lugar para começar a procura por softwares adicionais e contém instruções a respeito de como instalá-los no seu tipo de Linux, veja o [Appendix A, Para aonde ir a partir daqui?](#). Leia sempre a documentação fornecida juntamente com os softwares e outras diretrizes de instalação que o pacote venha a conter. Todos os softwares vêm com um arquivo README, cuja leitura é altamente recomendável.

## Formatos de pacotes

### Pacotes RPM

#### O que é um RPM?

O RPM, RedHat Package Manager, é um poderoso gerenciador de pacotes usado para instalar, atualizar ou remover pacotes. Ele permite que o usuário procure por pacotes e mantém um cadastro dos arquivos que vêm com cada pacote. Um sistema é construído de maneira que seja possível verificar a autenticidade dos pacotes baixados da internet. Usuários avançados podem criar seus próprios pacotes com o RPM.

Um pacote RPM consiste em um conjunto de arquivos e meta-dados usados para instalar e apagar os arquivos. Os meta-dados incluem scripts de ajuda, atributos de arquivos e a descrição de um pacote. Os pacotes vêm em duas variedades: binários, usados para encapsular softwares para instalação e pacotes fonte, que contém o código-fonte usado e a receita necessária para criar pacotes de binários.

Muitas outras distribuições suportam pacotes RPM, dentre as mais populares o RedHat Enterprise Linux, o Mandriva (antigo Mandrake), Fedora Core e o SuSE Linux. Além dos conselhos para a sua distribuição, aconselha-se a leitura da **man rpm**.

#### Exemplos de RPM

A maior parte dos pacotes é instalada com a opção de com a opção de upgrade -u, esteja ou não o pacote instalado. os pacotes RPM contém uma versão completa do programa que sobrescreve versões pré-existentes ou os instala como novos pacotes. O uso rotineiro encontra-se demonstrado a seguir:

```
rpm -Uvh /path/to/rpm-package(s)
```

A opção -v gera uma saída mais prolixa e a -h faz o **rpm** exibir uma barra de progresso:

```
[root@jupiter tmp]# rpm -Uvh totem-0.99.5-1.fr.i386.rpm
Preparing... ##### [100%]
 1:totem ##### [100%]
[root@jupiter tmp]#
```

Novos pacotes de kernel, entretanto, são instalados com a opção de instalação -i, que não sobrescreve a versão anterior do pacote. Desta forma, continuará sendo possível inicializar o sistema com o kernel antigo caso o novo não funcione.

Pode-se também usar o **rpm** para verificar se um pacote está instalado no seu sistema:

```
[david@jupiter ~] rpm -qa | grep vim
vim-minimal-6.1-29
vim-X11-6.1-29
vim-enhanced-6.1-29
vim-common-6.1-29
```

Ou você pode descobrir qual pacote contém um determinado arquivo ou executável:

```
[david@jupiter ~] rpm -qf /etc/profile
setup-2.5.25-1
```

```
[david@jupiter ~] which cat
cat is /bin/cat
```

```
[david@jupiter ~] rpm -qf /bin/cat
coreutils-4.5.3-19
```

Note que não é necessário acesso à conta administrativa para usar o **rpm** para pesquisar sua base de dados. Só é necessário ser o *root* quando se quiser adicionar, modificar ou deletar pacotes.

Abaixo se encontra um último exemplo que demonstra como desinstalar um pacote fazendo uso do **rpm**:

```
[root@jupiter root]# rpm -e totem
[root@jupiter root]#
```

Note que o padrão é que a desinstalação não seja muito prolixa, é normal que não se veja muita coisa acontecendo. Se estiver em dúvida, use o **rpm -qa** novamente e verifique se o pacote foi removido.

O RPM pode fazer muito mais do que as pequenas tarefas discutidas acima; o [RPM HOWTO](#) contém informações adicionais sobre suas funções.

## Pacotes DEB (.deb)

### O que são pacotes Debian?

Este formato de pacote é o padrão no Debian GNU/Linux, em que o **dselect** e, atualmente, o **aptitude**, são as ferramentas padrão para o gerenciamento de pacotes. São usados para selecionar os pacotes que se quer instalar ou atualizar, mas ele também é executado durante a instalação do sistema Debian e auxilia o usuário a definir o método de acesso a ser usado, listar os pacotes disponíveis e configurá-los.

O [o site do Debian](#) contém todas as informações de que você precisa, incluindo a “Documentação do dselect para Iniciantes”.

Recentemente, o formato de pacote Debian têm se tornado cada vez mais popular. Quando da redação deste guia, 5 dentre as distribuições top-10 usam-no. O **apt-get** (veja a [the section called “APT”](#) também tem se tornado extremamente popular, mesmo nos sistemas não-DEB.

### Exemplos com ferramentas DEB

A verificação do status de um programa é feita através do comando **dpkg**. Por exemplo, se você quiser saber qual versão do software Gallery encontra-se instalada na sua máquina:

```
nghtwsh@gorefest:~$ dpkg -l *gallery*
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Installed/Config-files/Unpacked/Failed-config/Half-installed
|/ Err?=(none)/Hold/Reinst-required/X=both-problems (Status,Err: uppercase=bad)
||/ Name          Version          Description
+++-----
ii gallery         1.5-1sarge2     a web-based photo album written in php
```

O prefixo “ii” quer dizer que um pacote está instalado. Caso você encontre o prefixo “un”, isto implica que o pacote está presente na lista que o seu computador mantém, mas que não encontra-se atualmente instalado.

Descobrir a qual pacote pertence um arquivo é feito usando a opção `-s` para o **dpkg**:

```
nghtwsh@gorefest:~$ dpkg -S /bin/cat
coreutils: /bin/cat
```

Pode-se encontrar mais informações nas páginas Info do **dpkg**.

## Pacotes fonte

A maior parte dos programas do Linux é Livre/tem Código-Fonte Aberto, portanto pacotes fonte encontram-se disponíveis para estes programas. Arquivos fonte são necessários para a compilação da sua própria versão do programa. Os Arquivos Fonte podem ser baixados direto do site, em geral compactados em bolas tar (`programa-versão.tar.gz` ou semelhante). Para distribuições baseadas no RPM, o fonte é normalmente fornecido na forma `programa-versão.src.rpm`. O Debian, bem como a maior parte das distribuições derivadas dele, provê o código-fonte adaptado, que pode ser obtido informando o **apt-get source**.

Exigências específicas, dependências e instruções de instalação são fornecidas no arquivo `LEIAME`. Você provavelmente precisará de um compilador C, o **gcc**. Este compilador C do GNU encontra-se incluído na maior parte dos sistemas Linux e é portado para muitas outras plataformas.

## Automatizando o gerenciamento de pacotes e atualizações

### Observações gerais

A primeira coisa a ser feita após a instalação de um novo sistema é atualizá-lo; isto se aplica a todos os sistemas operacionais e com o Linux não é diferente.

As atualizações para a maior parte dos sistemas Linux pode normalmente ser encontrada em sites regionais que espelham a sua distribuição. As listas de sites que oferecem este serviço podem ser encontradas no próprio site da sua distribuição, veja [Appendix A, Para aonde ir a partir daqui?](#).

Atualizações devem ser feitas com regularidade, diariamente, se possível, embora uma vez a cada duas semanas seja um bom começo. Você realmente deve tentar ter a versão mais recente da sua distribuição, uma vez que o Linux muda constantemente. Como já dito, novos recursos, melhorias e correção de erros são fornecidos em um ritmo constante e muitas vezes problemas de segurança são atacados.

A boa notícia é que a maior parte das distribuições do Linux provê ferramentas para auxiliá-lo nesta atividade, dispensando a atualização manual dos pacotes. As seções a seguir fornecem um panorama dos *gerenciadores de gerenciadores de pacotes*. Este assunto é extenso, até mesmo o a atualização rotineira de pacotes fonte pode ser automatizada; discutiremos apenas os sistemas mais conhecidos. Sempre busque informações complementares na documentação da sua distribuição específica para saber os procedimentos aconselhados.

### APT

O Advanced Package Tool é um sistema de gerenciamento de pacotes de software. A ferramenta de gerenciamento de pacotes na linha de comando é o **apt-get**, que tem uma excelente página man que descreve como realizar a instalação, atualização e upgrade de pacotes da sua distribuição. O APT tem sua raiz na distribuição Debian GNU/Linux, em que é o gerenciador padrão para os pacotes Debian. O APT foi portado para que também pudesse trabalhar com pacotes RPM. A sua maior vantagem é ser livre e flexível. Ele permite que você configure sistemas semelhantes às distribuições especiais (algumas vezes comerciais) listadas nas seções a seguir.

Em geral, quando usá-lo pela primeira vez **apt-get**, será necessário obter um rol dos pacotes disponíveis. Isto é feito informando o comando

```
apt-get update
```

Em seguida, pode-se usar o **apt-get** para atualizar seu sistema:

```
apt-get upgrade
```

Faça isto com freqüência, é um modo simples de manter seu sistema atualizado e, conseqüentemente, seguro.

Além deste uso genérico, o **apt-get** também é muito ágil na instalação de pacotes. Isto é feito assim:

```
[david@jupiter ~] su - -c "apt-get install xsnow"
Senha:
Lendo a lista de pacotes... Feito
Montando a árvore de dependências... Feito
Os seguintes NOVOS pacotes serão instalados:
  xsnow
0 pacotes atualizados, 1 novo instalado, 0 removidos e 3 não atualizados.
É preciso baixar 33.6kB de arquivos.
Depois de desempacotar 104kB de espaço em disco serão utilizados.
Get:1 http://ayo.freshrpms.net redhat/9/i386/os xsnow 1.42-10 [33.6kB]
Fetched 33.6kB in 0s (106kB/s)
Executando RPM (-Uvh)...
Preparando... ##### [100%]
```

Note a opção `-c` para o comando **su**, que indica que o shell do root deverá executar somente este comando e em seguida retornar para o ambiente do usuário. Desta forma, você não se esquecerá de sair da conta root.

Se houver quaisquer dependências em outros pacotes, o **apt-get** irá baixar e instalar estes pacotes auxiliares.

Mais informações podem ser encontradas no [APT HOWTO](#).

## Sistemas que usam pacotes RPM

O Update Agent, que originalmente suportava apenas os pacotes RPM, agora já é capaz de lidar com uma extensa gama de softwares, incluindo repositórios não-RedHat. Esta ferramenta fornece um sistema completo para atualizar os pacotes RPM em um sistema RedHat ou Fedora Core. Na linha de comando, digite **up2date** para atualizar seu sistema. No desktop, o padrão é que um pequeno ícone seja ativado, informando o usuário a respeito da existência de atualizações para o sistema.

O Yellowdog's Updater Modified (**yum**) é outra ferramenta que recentemente adquiriu popularidade. É um programa interativo, sem deixar de ser automático, que realiza a instalação, atualização e remoção de pacotes RPM em um sistema. É a ferramenta padrão nos sistemas Fedora.

No SuSE Linux, tudo é feito por meio do YaST, Yet another Setup Tool, que é capaz de realizar uma série de tarefas administrativas, dentre as quais, a atualização de pacotes RPM. A partir do SuSE Linux 7.1, também se tornou possível atualizar o sistema por meio de uma interface web e do YOU, Yast Online Update.

O Mandrake Linux e o Mandriva provêm as chamadas URPMI, um conjunto de programas destinados a tornar a instalação de novos softwares mais simples para o usuário. Estas ferramentas são combinadas com o RPMDrake e o MandrakeUpdate para fornecer tudo aquilo que é necessário à instalação e desinstalação suave de pacotes de software. MandrakeOnline oferece uma vasta gama de serviços e pode notificar os administradores automaticamente quando houver atualizações disponíveis para a versão específica do Mandrake em uso. Para saber mais, veja, dentre outros, o **man urpmi**.

Mais ainda, o KDE e o Gnome possuem suas próprias versões gráficas dos gerenciadores de pacotes.

## Atualizando o seu kernel

A maior parte das instalações do Linux permanece adequada se forem periodicamente atualizadas. O procedimento de atualização irá instalar um novo kernel quando preciso for, bem como realizará no seu sistema todas as mudanças necessárias. É recomendável que você compile ou instale um novo kernel manualmente apenas se realmente precisar de algum recurso indisponível no kernel padrão da sua distribuição.

Quer você compile sua própria versão otimizada do kernel, quer use um pacote pré-compilado, recomenda-se a sua instalação em coexistência com o antigo kernel, até que se tenha certeza de que tudo funciona da forma como deveria.

Crie, então, um sistema dual-boot que lhe permitirá a escolha do kernel a ser utilizado, o que se faz atualizando o arquivo de configuração do seu boot loader `grub.conf`. Segue um exemplo simples:

```
# grub.conf criado por anaconda
#
# Note que não é necessário re-executar o grub após realizar mudanças na configuração.
# NOTA:  Você possui uma partição /boot partition.  Isto significa que
#        todos os caminhos para o kernel e initrd relacionam-se com /boot/, p.ex.
#        root (hd0,0)
#        kernel /vmlinuz-version ro root=/dev/hde8
#        initrd /initrd-version.img
#boot=/dev/hde
default=0
timeout=10
splashimage=(hd0,0)/grub/splash.xpm.gz
title Red Hat Linux new (2.4.9-31)
    root (hd0,0)
    kernel /vmlinuz-2.4.9-31 ro root=/dev/hde8
    initrd /initrd-2.4.9-31.img
title old-kernel
    root (hd0,0)
    kernel /vmlinuz-2.4.9-21 ro root=/dev/hde8
    initrd /initrd-2.4.9-21.img
```

Após ter se certificado de que o novo kernel funciona de maneira adequada, você então poderá remover as linhas para o antigo no arquivo de configuração do GRUB, embora seja aconselhável aguardar uns dias antes de fazê-lo.

## Instalando pacotes extras a partir do CD de instalação

### Montando um CD

Basicamente, isto é feito de maneira semelhante à instalação manual de novos pacotes, com a exceção de que é preciso pensar o sistema de arquivos do CD ao de sua máquina, de forma a torná-lo acessível. Na maior parte dos sistemas, isto é feito automaticamente quando se coloca o CD no drive, por meio do daemon **automount** que é inicializado quando do boot. Se o seu CD não for montado automaticamente, informe o comando **mount** em uma janela de terminal. Dependendo da configuração do seu sistema, uma linha semelhante a esta provavelmente fará o serviço:

```
mount /dev/cdrom /mnt/cdrom
```

Em alguns sistemas, apenas o *root* pode montar mídias removíveis; isto depende da configuração.

Para fins de automação, o drive de CD normalmente possui uma ocorrência no `/etc/fstab`, que lista os sistemas de arquivo de seus pontos de montagem, que criam sua árvore de arquivos. Este é um exemplo de tal linha:

```
[david@jupiter ~] grep cdrom /etc/fstab
/dev/cdrom      /mnt/cdrom      iso9660          noauto,owner,ro 0 0
```

Isto indica que o sistema irá entender o comando **mount /mnt/cdrom**. A opção `noauto` significa que neste sistema, os CDs não são montados quando da inicialização do sistema.

Você também pode tentar clicar com o botão direito do mouse no ícone de CD do seu desktop para tentar montá-lo se o seu gerenciador de arquivos não o fizer por você. Para saber se funcionou, informe o comando **mount**, sem argumentos:

```
[david@jupiter ~] mount | grep cdrom
/dev/cdrom on /mnt/cdrom type iso9660 (ro,nosuid,nodev)
```

## Usando o CD

Após montar o CD, mudando de diretório, em regra para o ponto de montagem `/mnt/cdrom`, é possível acessar o conteúdo do CD-ROM. Para manipular os arquivos do CD, basta usar os mesmos comandos que usaria para o disco rígido.

## Ejetando o CD

Após terminar de usá-lo, para retirar o CD do drive é necessário que o sistema de arquivos dele não esteja em uso. A mera presença em um dos subdiretórios do ponto de montagem, `/mnt/cdrom` no nosso exemplo, é considerado “uso do sistema de arquivos”, portanto, é preciso que você saia dele. Faça-o, por exemplo, informando o comando **cd** sem argumento, que o redirecionará para o seu diretório home. Em seguida, você pode usar quaisquer dos comandos:

```
umount /mnt/cdrom
```

ou

```
eject cdrom
```

## Drives bloqueados

*NUNCA* force o drive. O truque com o clipe de papel é uma má idéia, pois, embora se consiga expulsar o CD, o sistema continuará achando que ele ainda está lá, vez que os procedimentos usuais não foram realizados. Provavelmente você precisará reiniciar o sistema para que o volte a apresentar um estado consistente.

Se você repetidas vezes obtiver avisos de “aparelho ocupado”, verifique se todas as sessões do shell saíram do sistema de arquivos do CD, bem como se nenhum aplicativo gráfico o está usando. Caso esteja em dúvida, use a ferramenta **lsof** para rastrear o(s) processo(s) que continuem usando o CD.

## Resumo

Quando tudo estiver em seu lugar, metade do trabalho terá sido realizado.

Embora manter-se em ordem seja importante, é de igual importância sentir-se em casa no seu ambiente, seja ele gráfico ou texto. O ambiente texto é controlado pelos arquivos de configuração do shell. O ambiente gráfico é precipuamente dependente da configuração do servidor do X, sobre o qual muitos outros aplicativos são construídos, como gerenciadores de janelas e desktop e aplicativos gráficos, cada um com seus próprios arquivos de configuração. Você deveria ler a documentação específica do sistema e programas em uso para descobrir como configurá-los.

O melhor momento para realizar as configurações regionais, como as de teclado, fontes e idioma, é o da instalação do sistema.

Os softwares são gerenciador automática ou manualmente usando o sistema de pacotes.

Os comandos a seguir foram introduzidos neste capítulo:

### Table 7.2. Novos comandos no capítulo 7: Fazendo-se sentir em casa

Comando	Significado
<b>aptitude</b>	Gerenciar pacotes no estilo Debian
<b>automount</b>	Incluir automaticamente sistemas arquivos recentemente inseridos.
<b>dpkg</b>	Gerenciador de pacotes do Debian
<b>dselect</b>	Gerenciar pacotes no estilo Debian
<b>loadkeys</b>	Carrega a configuração de teclado.
<b>lsof</b>	Identifica processos.
<b>mount</b>	Inclui um novo sistema de arquivos na árvore de arquivos do sistema.
<b>ntpdate</b>	Configura o horário e data do sistema usando um servidor de tempo.
<b>quota</b>	Exibe informações sobre o uso de espaço em disco permitido.
<b>recode</b>	Converte arquivos para outro conjunto de caracteres.
<b>rpm</b>	Gerencia pacotes RPM
<b>setfont</b>	Escolhe uma fonte.
<b>timezone</b>	Configura o fuso-horário
<b>tzconfig</b>	Configura o fuso-horário
<b>ulimit</b>	Configura ou exibe os limites de recursos.
<b>up2date</b>	Gerencia pacotes RPM
<b>urpmi</b>	Gerencia pacotes RPM
<b>yum</b>	Gerencia pacotes RPM

## Exercícios

### Ambiente shell

- Exiba suas configurações de ambiente. Qual variável pode ser usada para armazenar o tipo de CPU da sua máquina?
- Crie um script que diga alguma coisa como “olá, mundo.” Dê-lhe as devidas permissões de forma que ele possa ser executado. Teste-o.
- Crie um diretório na sua home e mova o script para lá. Adicione de modo permanente este novo diretório ao seu PATH. Teste se é possível executá-lo sem a necessidade de informar seu caminho completo.
- Crie subdiretórios na sua home e armazene diversos arquivos, por exemplo, um diretório `musicas` para os arquivos de áudio, outro `documentos` para as suas anotações e assim por diante. Use-os!
- Crie um prompt personalizado.
- Exiba os limites de recursos. Você consegue mudá-los?
- Tente ler páginas man compactadas sem descompactá-las.
- Crie um alias `ll` que execute o comando `ls -la`.
- Por que o comando `tail testfile > testfile` não funciona?
- Monte um CD de dados, como o de instalação do Linux, e mova-se por ele. Não se esqueça de desmontá-lo quando não precisar mais dele.
- O script do [the section called “Alguns exemplos simples”](#) não é perfeito. Ele gera erros para arquivos que sejam diretórios. Adapte-o de forma que ele somente escolha arquivos simples para copiá-los. Use o `find` para realizar a seleção. Não se esqueça de torná-lo executável antes de tentar rodá-lo.

### Ambiente gráfico

- Tente usar os botões do mouse em diferentes regiões (terminal, plano de fundo, barra de tarefas).
- Explore os menus.
- Personalize a sua janela de terminal
- Use os botões do mouse para copiar e colar texto de um terminal para outro.
- Descubra como configurar seu gerenciador de janelas; tente diferentes ambientes de trabalho (telas virtuais).
- Adicione um applet, como um monitor de uso, à barra de tarefas.
- Use um tema diferente.
- Acione o recurso de foco `sloppy` - que faz com que uma janela seja ativada quando o mouse passar

sobre ela, de forma que torne-se desnecessário clicar nela para usá-la.

- Mude para um gerenciador de janelas diferente.
- Saia do sistema e selecione um tipo de sessão diferente, como o KDE, se vocês estiver usando o Gnome. Repita os passos acima.

## Chapter 8. Impressoras e impressão

### Table of Contents

#### [Imprimir arquivos](#)

[Imprimir a partir da linha de comando](#)

[Formatação](#)

#### [O lado do servidor](#)

[Geral](#)

[Configuração gráfica da impressora](#)

[Comprando uma impressora para o Linux](#)

#### [Problemas de impressão](#)

[Arquivo errado](#)

[Minha impressão não saiu](#)

[Resumo](#)

[Exercícios](#)

### Abstract

Neste capítulo aprenderemos mais sobre impressora e impressão de arquivos. Após ler esta parte, você será capaz de:

- Formatar documentos
- Exibir uma prévia antes de encaminhar o arquivo para a impressora
- Escolher uma boa impressora que funcione com o Linux
- Imprimir arquivos e verificar o status da impressora
- Resolver problemas de impressão
- Encontrar a documentação para instalar uma impressora

## Imprimir arquivos

### Imprimir a partir da linha de comando

#### Encaminhando o arquivo à impressora

Imprimir a partir de um aplicativo é muito simples, selecione a opção Imprimir do menu.

A partir da linha de comando, use o **lp** ou o **lpr**.

```
lp file(s)
```

```
lpr file(s)
```

Estes comando conseguem ler a partir de um pipe, de modo que você pode imprimir a saída de comando usando

```
command | lp
```

Existem muitas opções disponíveis para aperfeiçoar o layout da página, o número de cópias, a impressora em que você deseja imprimir, caso mais de uma esteja disponível, o tamanho do papel, impressão em uma ou ambas as faces, se a sua impressora for capaz de fazê-lo, margens e assim por diante. Leia as páginas man para um panorama sobre o assunto.

#### Status das suas impressões

Tendo o arquivo sido aceito na fila de impressão, ele recebe um número de identificação de trabalho de impressão

```
davy:~> lp /etc/profile
request id is blob-253 (1 file(s))
```

Para visualizar a fila de impressão, use o comando **lpq** ou **lpstat**. Quando informado sem argumentos, ele

exibe o conteúdo da fila de impressão padrão.

```
davy:~> lpq
blob is ready and printing
Rank      Owner      Job      File(s)      Total Size
active    davy       253     profile      1024 bytes
davy:~> lpstat
blob-253   davy       1024     Tue 25 Jul 2006 10:20_01 AM CEST
```

## Status da sua impressora

Qual é a impressora padrão em um sistema que tem acesso à diversas impressoras?

```
lpstat -d
```

```
davy:~> lpstat -d
system default destination: blob
```

Qual é o status da minha impressora(s)?

```
lpstat -p
```

```
davy:~> lpstat -p
printer blob now printing blob-253. enabled since Jan 01 18:01
```

## Removendo trabalhos da fila de impressão

Se você não gosta do que vê nos comandos de status, use **lprm** ou **cancel** para deletar trabalhos.

```
davy:~> lprm 253
```

No ambiente gráfico, você provavelmente verá uma janela informando que o trabalho foi cancelado.

Em ambientes maiores, o **lpc** pode ser usado para controlar múltiplas impressoras. Veja as páginas Info ou man para cada um destes comandos.

Existem muitas ferramentas gráficas usadas como interface para o **lp** e a maior parte dos aplicativos gráficos têm uma função de impressão que usa o **lp**. Veja a função de Ajuda interna e a documentação específica para saber mais.

## Por que existem dois comandos para todas as tarefas relacionadas à impressão?

A impressão nos sistemas UNIX e similares tem uma longa história. Havia inicialmente duas diferentes abordagens: os estilos de impressão BSD e SystemV. Por questão de compatibilidade, o Linux com CUPS suporta ambos os comandos. Perceba também que o **lp** não se comporta de modo inteiramente idêntico ao **lpr**, o **lpq** possui algumas opções diferentes do **lpstat** e o **lprm** é somente parecido com o **cancel**. O comando que você vai usar não importa, escolha aquele que você se sentir confortável em usar ou que já tiver alguma experiência prévia.

## Formatação

### Ferramentas e Idiomas

Se quisermos obter algo inteligível da impressora, os arquivos devem estar devidamente formatados. Além de uma imensa variedade de softwares de edição, o Linux também possui algumas ferramentas básicas de formatação e idiomas.

Os sistemas Linux modernos suportam impressão direta, sem qualquer formatação prévia do usuário, de alguns formatos de arquivo: text, PDF, PostScript e diversos tipos de imagem como PNG, JPEG BMP e GIF.

Para aqueles formatos de arquivo que precisam de formatação, o Linux vem com muitas ferramentas de formatação, como o **pdf2ps**, o **fax2ps** e o **a2ps**, que converte outros formatos para PostScript. Estes comandos podem criar arquivos que poderão então ser usados em outros sistemas que não possuem todas as ferramentas de conversão instaladas.

Além destas ferramentas da linha de comando existem muitos programas de processamento de texto. Muitos suítes de escritório completos encontram-se disponíveis, a maior parte deles livremente. Estes realizam a formatação automaticamente quando um trabalho for encaminhado à impressão. Apenas para citar alguns: OpenOffice.org, KOffice, AbiWord, WordPerfect, etc.

As linguagens a seguir são comuns no contexto de impressão:

- **groff**: a versão GNU do comando UNIX **roff**. É uma interface para o sistema de formatação de documentos groff. Normalmente ele executa o **troff** e um pós-processador apropriado ao equipamento selecionado. Ele permite a criação de arquivos PostScript.
- O TeX e o pacote macro LaTeX: uma das linguagens de marcação mais usadas nos sistemas UNIX. Usualmente invocadas como **tex**, ele formata arquivos e informa uma saída que corresponde a uma



representação do documento independente do equipamento.

Trabalhos técnicos *ainda* são freqüentemente escritos no LaTeX por conta de seu suporte a formulas matemáticas, embora o [W3C](#) (o Consórcio da World Wide Web) venha realizando esforços para incluir este recurso em outros aplicativos.

- O SGML e o XML: analisadores livres encontram-se disponíveis para UNIX e Linux. O XML é a próxima geração do SGML e perfaz a base para o DocBook XML, um sistema de documentação (este livro, por exemplo, foi escrito em XML).

## Imprimindo documentos

As páginas man contém dados pré-formatados pelo **troff**, requisito para a impressão. A impressão é feita usando a opção `-t` para o comando **man**:

```
man -t command > man-command.ps
```

Imprima, então, o arquivo PostScript. Se uma impressora padrão estiver configurada para o sistema/conta, pode-se simplesmente informar o comando **man -tcomando** para enviar a página à impressora:

## Exibindo arquivos pré-formatados

Qualquer coisa que se possa enviar à impressora em geral pode ser enviada também ao monitor. Dependendo do formato do arquivo, é possível usar um destes comandos:

- Arquivos PostScript: com o **gv** (GhostView).
- Arquivos TeX dvi: com o **xdvi** ou o **kdvi** do KDE.
- Arquivos PDF: **xpdf**, **kpdf**, **gpdf** ou visualizador da Adobe, **acroread**, que é gratuito, porém não é livre. O visualizador da Adobe suporta o PDF 1.6, os demais apenas as versões até a 1.5. A versão de um arquivo PDF pode ser determinada pelo comando **file**.
- A partir de aplicativos como o Firefox ou o OpenOffice, geralmente é possível selecionar Visualizar Impressão em algum dos menus.

## O lado do servidor

### Geral

Até poucos anos atrás, a escolha para todos os usuários Linux era simples: todos executavam o mesmo velho LPD a partir de um código Net-2 do BSD. Tornou-se, então, popular o LPRng, porém hoje em dia a maior parte das distribuições do Linux usa o [CUPS](#), o Common UNIX Printing System. O CUPS é uma implementação do IPP (Internet Printing Protocol), um padrão RFC semelhante ao HTTP, desenvolvido para substituir o venerável protocolo LPD. O CUPS é distribuído através da Licença Pública GNU, sendo o sistema de impressão padrão no MacOS X.

### Configuração gráfica da impressora

A maior parte das distribuições vem com um aplicativo gráfico para a configuração das impressoras locais ou em rede (portas paralelas ou USB). Eles permitem que o usuário escolha o modelo de impressora a partir de uma lista e permitem alguns testes simples. Não é preciso se preocupar com sintaxe ou localização dos arquivos de configuração. Verifique a documentação do seu sistema antes de instalar a sua impressora.

O CUPS também pode ser configurado usando uma interface web que é executada na porta 631 do seu computador. Para verificar se este recurso se encontra habilitado, tente acessar [localhost:631/help](http://localhost:631/help) ou [localhost:631/](http://localhost:631/).

### Comprando uma impressora para o Linux

Como é cada vez maior o número de comerciantes que têm disponibilizado drivers para o CUPS, ele permite uma conexão simples com praticamente todas as impressoras que puderem ser conectadas a uma porta serial, paralela, USB, além de quaisquer outras em rede. O CUPS garantirá uma apresentação uniforme para você e todos os seus aplicativos, de quaisquer modelos de impressora.

Impressoras que tenham apenas um driver Win9x podem se tornar problemáticas se não tiverem outro tipo de suporte. Verifique no <http://linuxprinting.org/> em caso de dúvida.

No passado, a melhor escolha seria uma impressora com suporte nativo ao PostScript na firmware, vez que praticamente todos os softwares UNIX ou Linux produzem saídas de impressão em PostScript, a linguagem de impressão preferida pela indústria de publicação. Impressoras PostScript são em geral mais caras, porém independentes do equipamento usado e programadas em linguagem aberta, de modo que é garantido que elas irão funcionar. Atualmente, entretanto, a importância desta orientação tem decaído.

## Problemas de impressão

Nesta seção, discutiremos o que você, enquanto usuário, pode fazer quando algo der errado. Não discutiremos quaisquer problemas relacionados aos daemons de impressão, vez que este tema é de interesse apenas dos administradores de sistema.

## Arquivo errado

Se você encaminhar o arquivo errado para a fila de impressão, ele poderá ser cancelado informando o comando **lprm jobId** em que jobId tem o formato *nomedaimpressora-númerodaimpressão* (obtenha estas informações com os comandos **lpq** ou **lpstat**). Isto funcionará quando houver outros trabalhos na fila de impressão. Entretanto, você precisa ser realmente rápido se for o único usando a impressora, vez que os trabalhos em geral são encaminhados à impressora em questão de segundos. Quando chegam à impressora já é tarde para impedi-los usando as ferramentas do Linux.

O que você pode tentar nestes casos ou naqueles em que o driver de impressão estiver configurado de maneira incorreta, resultando em impressões ilegíveis, é desligá-la. Todavia, este procedimento pode resultar em atolamentos de papel e outras irregularidades.

## Minha impressão não saiu

Use o comando **lpq** e procure pelo seu trabalho:

```
elly:~> lpq
Printer: lp@blob
Queue: 2 printable jobs
Server: pid 29998 active
Unspooler: pid 29999 active
Status: waiting for subserver to exit at 09:43:20.699
Rank  Owner/ID          Class Job Files      Size Time
1     elly@blob+997      A    997 (STDIN)    129 09:42:54
2     elly@blob+22      A     22 /etc/profile  917 09:43:20
```

Muitas impressoras possuem interface web atualmente, sendo possível exibir as informações sobre o seu status informando o endereço IP do navegador:

**Figure 8.1. Status da Impressora através da interface web**



## A interface web do CUPS versus a interface web da impressora

Perceba que esta não é a interface web do CUPS e funciona apenas com impressoras que suportam este recurso. Verifique a documentação para a sua impressora.

Se o seu ID de trabalho não estiver lá ou na impressora, fale com o administrador do seu sistema. Se o seu trabalho estiver na lista, verifique se a impressora está em uso. Se sim, aguarde, seu trabalho será impresso em seu devido tempo.

Se a impressora não estiver imprimindo, verifique se ela tem papel ou se as conexão de dados e eletricidade estão devidamente plugadas. Se tudo estiver certo, pode ser que a impressora precise ser reiniciada. Busque ajuda com o administrador do seu sistema.

No caso de uma impressora em rede, tente imprimir a partir de outra máquina. Se for possível acessar a impressora a partir da sua máquina (veja o [Chapter 10, Redes](#) sobre o utilitário **ping**), você pode tentar colocar o arquivo formatado nela, como `arquivo.ps` no caso de uma impressora PostScript, que utiliza um cliente FTP. Se isto não funcionar, é possível que a impressora não compreenda o formato com que você a está alimentando.

O [Site de Impressão GNU/Linux](#) contém mais dicas e truques.

## Resumo

O serviço de impressão do Linux vem com um conjunto de ferramentas de impressão baseadas no padrão UNIX LPD, em ambas as implementações SystemV e BSD. Segue uma lista de comandos relacionados com a impressão.

**Table 8.1. Novos comandos no capítulo 8: Impressão**

Comando	Significado
<b>lpr</b> or <b>lp</b>	Imprime um arquivo
<b>lpq</b> or <b>lpstat</b>	Consulta a fila de impressões
<b>lprm</b> or <b>cancel</b>	Remove uma impressão
<b>acroread</b>	Visualizador de PDF
<b>groff</b>	Ferramenta de formatação
<b>gv</b>	Visualizador PostScript
<b>printconf</b>	Configura impressoras
<b>xdvi</b>	Visualizador DVI

<b>xpdf</b>	Visualizador de PDF
<b>*2ps</b>	Converte um arquivo para PostScript

## Exercícios

Configurar e testar impressoras envolve a posse de uma, bem como o acesso à conta *root*. Se você possui ambos, tente:

- Instalar a impressora usando a interface gráfica do seu sistema.
- Imprimir uma página teste usando a interface gráfica.
- Imprimir uma página teste usando o comando **lp**.
- Imprimir a partir de um aplicativo, por exemplo, o Mozilla ou o OpenOffice, selecionando Arquivo → Imprimir no menu.
- Desconecte a impressora da rede ou máquina/servidor. O que acontece quando se tenta imprimir algo?

Os exercícios a seguir podem ser realizados sem acesso a uma impressora ou conta *root*.

- Tente criar arquivos PostScript a partir de diferentes arquivos fonte, (p. ex. HTML, PDF, páginas man). Teste os resultados com o visualizador **gv**.
- Verifique se o daemon de impressão está em execução.
- Imprima os arquivos mesmo assim. O que acontece?
- Crie um arquivo PostScript usando o Mozilla. Teste-o com o **gv**.
- Converta-o para PDF. Teste-o com o **xpdf**.
- Como você faria para imprimir um arquivo GIF a partir da linha de comando?
- Use o **a2ps** para imprimir o arquivo `/etc/profile` para um arquivo de saída. Teste-o novamente com o **gv**. O que acontece se você não especifica um arquivo de saída?

## Chapter 9. Técnicas Fundamentais de Backup

### Table of Contents

#### [Introdução](#)

##### [Preparando seus dados](#)

#### [Movendo seus dados para um dispositivo de backup](#)

##### [Criando uma cópia em um disquete](#)

##### [Criando uma cópia com o gravador de CDs](#)

##### [Backups em/a partir de jazz drives, dispositivos USB e outros removíveis](#)

##### [Fazendo backup de dados com um dispositivo de fita](#)

##### [Ferramentas para a sua distribuição](#)

#### [Usando o rsync](#)

##### [Introdução](#)

##### [Um exemplo: rsync em um dispositivo de armazenamento USB](#)

#### [Criptografia](#)

##### [Notas gerais](#)

##### [Gerando uma chave](#)

##### [Sobre a sua chave](#)

##### [Criptografando seus dados](#)

##### [Descriptografando arquivos](#)

#### [Resumo](#)

#### [Exercícios](#)

### Abstract

Acidentes acontecerão cedo ou tarde. Neste capítulo, discutiremos como encaminhar dados para um local seguro usando outras máquinas, disquetes, CD-ROMs e fitas. Discutiremos ainda os comandos mais comuns para compressão de arquivos.

Ao terminar o presente capítulo, você saberá como:

- Criar, consultar e descompactar arquivos

- Manipular disquetes e criar um disquete de inicialização para o seu sistema
- Gravar CD-ROMs
- Adicionar arquivos ao arquivo de backup
- Criar arquivos Java
- Encontrar documentação para o uso de outros programas e dispositivos de backup.
- Criptografar seus dados

## Introdução

Embora o Linux seja um dos sistemas operacionais mais seguros e ainda que tenha sido projetado para continuar funcionando, é possível que dados se percam. A perda de dados em geral acontece por culpa do usuário, mas ocasionalmente uma falha do sistema, como uma queda de energia ou falha no disco, pode ocasionar erros, de modo que é importante manter sempre uma cópia sobressalente dos dados mais vitais.

## Preparando seus dados

### Arquivando com o tar

Na maior parte dos casos, iremos primeiramente armazenar em um único arquivo todos os dados dos quais queremos fazer o backup, o qual em seguida comprimiremos. O processo de arquivamento envolve a concatenação de todos os arquivos listados, retirando os vazios desnecessários. No Linux, isto normalmente é feito com o comando **tar**. O **tar** inicialmente foi projetado para arquivar dados em fitas, porém também pode criar arquivos conhecidos como *tarballs*.

O **tar** possui muitas opções, sendo as mais importantes:

- **-v**: verborrágico (ou detalhado)
- **-t**: teste, exibe o conteúdo da tarball
- **-x**: extrai os conteúdos do ficheiro
- **-c**: cria um ficheiro
- **-f**: usa o dispositivo de arquivo como uma fonte/destino para a tarball, o padrão é o primeiro dispositivo de fita (normalmente `/dev/st0` ou algo semelhante)
- **-j**: filter through **bzip2**, see [the section called "Adição ao backup com o tar"](#)

É comum deixar de fora o prefixo hífen nas opções do **tar**, como se pode ver nos exemplos abaixo.

### Use o GNU tar por motivos de compatibilidade

Os arquivos criados em um sistema com a versão proprietária do **tar** podem ser incompatíveis com o **tar** em outro sistema proprietário. Isto pode causar sérias dores de cabeça, como por exemplo quando o arquivo precisar de um sistema que já não existe mais. Use a versão GNU do **tar** em todos os sistemas para prevenir que o administrador do seu sistema fique aos prantos. O Linux sempre usa o GNU tar. Quando estiver trabalhando em outras máquinas UNIX, informe o **tar --help** para descobrir que versão está em uso. Entre em contato com o administrador do seu sistema se não encontrar a palavra GNU em lugar algum.

No exemplo abaixo, um arquivo é criado e descompactado.

```
gaby:~> ls images/
me+tux.jpg  nimf.jpg

gaby:~> tar cvf images-in-a-dir.tar images/
images/
images/nimf.jpg
images/me+tux.jpg

gaby:~> cd images

gaby:~/images> tar cvf images-without-a-dir.tar *.jpg
me+tux.jpg
nimf.jpg

gaby:~/images> cd

gaby:~> ls */*.tar
images/images-without-a-dir.tar

gaby:~> ls *.tar
```

```
images-in-a-dir.tar
```

```
gaby:~> tar xvf images-in-a-dir.tar
```

```
images/  
images/nimf.jpg  
images/me+tux.jpg
```

```
gaby:~> tar tvf images/images-without-dir.tar
```

```
-rw-r--r-- gaby/gaby 42888 1999-06-30 20:52:25 me+tux.jpg  
-rw-r--r-- gaby/gaby 7578 2000-01-26 12:58:46 nimf.jpg
```

```
gaby:~> tar xvf images/images-without-a-dir.tar
```

```
me+tux.jpg  
nimf.jpg
```

```
gaby:~> ls *.jpg
```

```
me+tux.jpg nimf.jpg
```

Este exemplo também ilustra a diferença entre um diretório comprimido com o **tar** e um conjunto de arquivos individualmente comprimidos. Recomenda-se a compressão apenas de diretórios, de forma que os arquivos não fiquem todos espalhados quando da descompressão da **tarball** (que pode ser feita em outro sistema, em que não se sabe exatamente quais arquivos já estavam lá e quais foram extraídos do ficheiro).

Quando um drive de fita estiver conectado à sua máquina e configurado pelo administrador do seu sistema, os nomes dos arquivos terminados por **.tar** serão substituídos pelo nome de dispositivo da fita, por exemplo:

```
tar cvf /dev/tape mail/
```

O diretório **mail** e todos os arquivos que ele contém são comprimidos em um arquivo que é escrito na fita imediatamente. A lista de conteúdos é exibida porque foi usada a opção verborrágica.

### Adição ao backup com o **tar**

A ferramenta **tar** permite a adição de novos arquivos ao backup, por meio da opção **-N**. Com esta opção, é possível especificar uma data e o comando **tar** verificará se houve modificações de quaisquer dos arquivos especificados desde tal data. Se qualquer dos arquivos tiver sido alterado em data mais recente, eles serão incluídos ao backup. O primeiro exemplo abaixo usa o carimbo de hora em um ficheiro anterior como o valor da data. Primeiro, o ficheiro inicial é criado e a data do arquivo é exibida. Então, um novo arquivo é criado, a partir do qual se realiza um novo backup, contendo apenas este novo arquivo:

```
jimmy:~> tar cvpf /var/tmp/javaproggies.tar java/*.java
```

```
java/btw.java  
java/error.java  
java/hello.java  
java/income2.java  
java/income.java  
java/inputdevice.java  
java/input.java  
java/master.java  
java/method1.java  
java/mood.java  
java/moodywaitress.java  
java/test3.java  
java/TestOne.java  
java/TestTwo.java  
java/Vehicle.java
```

```
jimmy:~> ls -l /var/tmp/javaproggies.tar
```

```
-rw-rw-r-- 1 jimmy jimmy 10240 Jan 21 11:58 /var/tmp/javaproggies.tar
```

```
jimmy:~> touch java/newprog.java
```

```
jimmy:~> tar -N /var/tmp/javaproggies.tar \
```

```
-cvp /var/tmp/incremental1-javaproggies.tar java/*.java 2> /dev/null  
java/newprog.java
```

```
jimmy:~> cd /var/tmp/
```

```
jimmy:~> tar xvf incremental1-javaproggies.tar
```

```
java/newprog.java
```

Os erros padrão são enviados para **/dev/null**. Se você não fizer isto, o **tar** exibirá uma mensagem para cada arquivo não modificado, avisando-o de que ele não será adicionado.

Este modo de operar tem a desvantagem de buscar pelo carimbo de hora nos arquivos. Digamos que você baixe um arquivo para o diretório que contém os seus backups e que ele contenha arquivos que foram criados há dois anos. Quando chegar o carimbo de tempo destes arquivos e compará-los com aqueles do ficheiro inicial, os novos arquivos parecerão mais antigos para o **tar**, que não os adicionará ao backup quando for informada a opção **-N**.

Uma alternativa melhor seria a opção **-g**, que irá criar uma lista dos arquivos dos quais se quer fazer o backup. Quando novos backups forem realizados, os arquivos serão comparados com esta lista. Isto

funciona da seguinte maneira:

```
jimmy:~> tar cvpf work-20030121.tar -g snapshot-20030121 work/  
work/  
work/file1  
work/file2  
work/file3
```

```
jimmy:~> file snapshot-20030121  
snapshot-20030121: ASCII text
```

No dia seguinte, o usuário *jimmy* trabalha um pouco mais no arquivo3 e cria o arquivo4. No final do dia, ele faz um novo backup:

```
jimmy:~> tar cvpf work-20030122.tar -g snapshot-20030121 work/  
work/  
work/file3  
work/file4
```

Estes são alguns exemplos muito simples, mas você também pode usar este tipo de comando em um trabalho cron (veja o [the section called "Cron e crontab"](#)), que especifica um snapshot para o backup semanal e outro para o diário. Neste caso, os snapshots deverão ser substituídos quando se fizer backups integrais.

Mais informações podem ser encontradas na documentação do **tar**.

## A realidade

Como você provavelmente pode perceber, o **tar** é ótimo quando estamos falando de um simples diretório, um conjunto de arquivos que permanecem juntos. Todavia, existem muitas ferramentas que são de uso mais simples, quando se trata de fazer backups de partições, discos ou projetos maiores. Tratamos aqui do **tar** porque ele é uma ferramenta muito popular para a distribuição de arquivos. Você muitas vezes se deparará com a necessidade de instalar softwares que vêm comprimidos em "tarballs". Veja o [the section called "Usando o rsync"](#) para aprender um meio mais simples de realizar backups rotineiros.

## Compactando e descompactando com o gzip ou o bzip2

Dados, incluindo tarballs, podem ser compactados por meio de ferramentas zip. O comando **gzip** adicionará o sufixo .gz ao nome do arquivo e removerá o arquivo original.

```
jimmy:~> ls -la | grep tar  
-rw-rw-r-- 1 jimmy jimmy 61440 Jun 6 14:08 images-without-dir.tar
```

```
jimmy:~> gzip images-without-dir.tar
```

```
jimmy:~> ls -la images-without-dir.tar.gz  
-rw-rw-r-- 1 jimmy jimmy 50562 Jun 6 14:08 images-without-dir.tar.gz
```

Descompacte arquivos gzipados com a opção `-d`.

O **bzip2** funciona de uma maneira semelhante, porém faz uso de algoritmo de compressão melhor e, portanto, cria arquivos menores. Veja a página info do **bzip2** para saber mais.

Os pacotes de softwares do Linux são muitas vezes distribuídos em tarballs gzipadas. A melhor coisa a ser feita após descompactar este tipo de arquivo é procurar pelo README e lê-lo. Normalmente ele fornece as diretrizes para a instalação do pacote.

O comando **tar** do GNU é capaz de lidar com arquivos gzipados. Use o comando

```
tar zxvf file.tar.gz
```

para deszipar e destarar arquivos .tar.gz ou .tgz. Use

```
tar jxvf file.tar.bz2
```

para descompactar arquivos **tar** que foram compactados com o **bzip2**.

## Ficheiros Java

O projeto GNU nos fornece a ferramenta **jar** para a criação de ficheiros Java. Eles são um aplicativo Java que combina vários arquivos em um único ficheiro JAR. Embora ele também seja uma ferramenta genérica de arquivamento e compressão, baseado nos formatos ZIP e ZLIB, o **jar** foi criado principalmente para facilitar o empacotamento de código Java, applets e/ou aplicativos em um único arquivo. Quando combinados em um único arquivo, os componentes do aplicativo Java podem ser baixados muito mais rápido.

Ao contrário do **tar**, o padrão do **jar** é comprimir independentemente de outras ferramentas - já que ele é basicamente a versão Java do **zip**. Além disso, ele permite que entradas individuais sejam assinadas pelo autor, de forma que a origem possa ser autenticada.

A sintaxe é quase idêntica à do comando **tar**, para saber detalhes sobre suas diferenças, leia a página

info *jar*.

## tar, jar e links simbólicos

Um recurso digno de nota que não é mencionado na documentação padrão é que o **jar** segue os links simbólicos. Os dados para os quais estes links apontam serão incluídos no ficheiro. O padrão no **tar** é fazer o backup apenas dos links simbólicos, este comportamento, contudo, pode ser alterado usando a opção `-h` para o **tar**.

### Transportando os seus dados

Salvar cópias dos seus dados em outra máquina é um modo simples, porém eficaz de realizar backups. Veja o [Chapter 10, Redes](#) para saber sobre **scp**, **ftp** e mais.

Na seção seguinte, discutiremos dispositivos de backup local.

## Movendo seus dados para um dispositivo de backup

### Criando uma cópia em um disquete

#### Formatando o disquete

Na maior parte dos sistemas Linux, os usuário têm acesso ao drive de disquetes. O nome do drive pode variar de acordo com o tamanho e número de drives, contate o seu administrador em caso de dúvida. Em alguns sistemas, provavelmente existirá um link `/dev/floppy` direcionando para o dispositivo correto, normalmente `/dev/fd0` (o dispositivo de auto-deteção de disquetes) ou `/dev/fd0H1440` (para disquetes de 1,44MB).

O **fdformat** é a ferramenta de baixo-nível para formatar disquetes. Possui o nome do dispositivo de disquete como uma opção. O **fdformat** irá exibir caso o disquete esteja protegido contra a gravação.

```
emma:~> fdformat /dev/fd0H1440
Double-sided, 80 tracks, 18 sec/track. Total capacity 1440 kB.
Formatting ... done
Verifying ... done
emma:~>
```

O comando **mformat** (do pacote **mtools**) é usado para criar disquetes compatíveis com o MSDOS, que poderão então ser acessados com o **mcopy**, o **mdir** e outros m-comandos.

Ferramentas gráficas também se encontram disponíveis

#### Figure 9.1. Floppy formatter



Após a formatação do disquete, ele pode ser montado no sistema de arquivos e acessado como um diretório normal, ainda que pequeno, geralmente através de `/mnt/floppy`.

Caso você precise, instale o utilitário **mkbootdisk**, que cria um disquete a partir do qual o sistema atual pode ser iniciado.

#### Usando o dd para copiar dados

O comando **dd** pode ser usado para colocar dados em um disco ou dele retirá-los, dependendo de quais os dispositivos de entrada e saída. Um exemplo :

```
gaby:~> dd if=images-without-dir.tar.gz of=/dev/fd0H1440
98+1 records in
98+1 records out

gaby~> dd if=/dev/fd0H1440 of=/var/tmp/images.tar.gz
2880+0 records in
2880+0 records out

gaby:~> ls /var/tmp/images*
/var/tmp/images.tar.gz
```

Perceba que este método de cópia é feito em dispositivos não-montados. Disquetes criados usando este método não serão montáveis ao sistema de arquivos, porém este é o procedimento para criar discos de inicialização ou resgate. Para mais informações nas possibilidades do **dd**, leia as páginas man.

Esta ferramenta é parte do pacote GNU *coreutils*.

## Despejando discos

O comando **dd** também pode ser usado para criar uma imagem crua de um disco rígido.

## Criando uma cópia com o gravador de CDs

Em alguns sistemas os usuários têm permissão para usar o gravador de CDs. Primeiro seus dados precisarão ser formatados. Use o comando **mkisofs** para fazê-lo em um diretório que contenha arquivos dos quais você quer fazer o backup. Verifique com o **df** se há espaço suficiente para tanto, já que será criado um novo arquivo com aproximadamente o mesmo tamanho do seu diretório atual:

```
[rose@blob recordables] df -h .
Filesystem      Size  Used Avail Use% Mounted on
/dev/hde5       19G   15G  3.2G  82% /home
```

```
[rose@blob recordables] du -h -s .
325M  .
```

```
[rose@blob recordables] mkisofs -J -r -o cd.iso .
<--snap-->
making a lot of conversions
<--/snap-->
98.95% done, estimate finish Fri Apr  5 13:54:25 2002
Total translation table size: 0
Total rockridge attributes bytes: 35971
Total directory bytes: 94208
Path table size(bytes): 452
Max brk space used 37e84
166768 extents written (325 Mb)
```

As opções **-J** e **-r** são usadas para tornar o CD-ROM montável em diferentes sistemas, veja as páginas man para saber mais. Depois disto, o CD poderá ser criado usando o **cdrecord** com as opções adequadas:

```
[rose@blob recordables] cdrecord -dev 0,0,0 -speed=8 cd.iso
Cdrecord 1.10 (i686-pc-linux-gnu) (C) 1995-2001 Joerg Schilling
scsidev: '0,0,0'
scsibus: 0 target: 0 lun: 0
Linux sg driver version: 3.1.20
Using libscg version 'schily-0.5'
Device type      : Removable CD-ROM
Version          : 0
Response Format: 1
Vendor_info      : 'HP          '
Identification  : 'CD-Writer+ 8100 '
Revision        : '1.0g'
Device seems to be: Generic mmc CD-RW.
Using generic SCSI-3/mmc CD-R driver (mmc_cdr).
Driver flags    : SWABAUDIO
Starting to write CD/DVD at speed 4 in write mode for single session.
Last chance to quit, starting real write in 0 seconds.
Operation starts.
```

Dependendo do seu gravador de CD, você agora terá algum tempo para fumar^H^H^H^H comer uma fruta bem saudável e/ou tomar uma xícara de café. Quando o trabalho tiver terminado, será informada uma mensagem de confirmação:

```
Track 01: Total bytes read/written: 341540864/341540864
(166768 sectors).
```

Existem algumas ferramentas gráficas disponíveis que tornam esta tarefa mais simples para o usuário. Um dos mais populares é o **xcdroast**, que é distribuído livremente no [the X-CD-Roast web site](#) e encontra-se incluso na maior parte dos sistemas e no diretório GNU. Ambos, KDE e Gnome, possuem ferramentas próprias para a gravação de CDs.

## Backups em/a partir de jazz drives, dispositivos USB e outros removíveis

Estes dispositivos em geral são montados no sistema de arquivos. Após o procedimento de montagem, eles podem ser acessados como diretórios normais, de forma que é possível usar os comandos usuais para manipular os arquivos.

No exemplo abaixo, imagens são copiadas de uma câmera USB para o disco rígido:

```
robin:~> mount /mnt/camera
robin:~> mount | grep camera
/dev/sda1 on /mnt/camera type vfat (rw,nosuid,nodev)
```

Se a câmera for o único dispositivo USB de armazenamento que estiver conectado ao seu sistema, isto é seguro. Mantenha em mente, todavia, que os dispositivos USB recebem uma entrada no **/dev** de acordo com a ordem em que são conectados ao sistema. Portanto, se você conectar um pendrive ao sistema ele receberá a entrada **/dev/sda**, e se, após isto, você conectar sua câmera, ela receberá a entrada **/dev/sdb** - desde que você não tenha quaisquer discos SCSI, que também são alocados no **/dev/sd\***. Em sistemas mais recentes, a partir da versão 2.6 do kernel, um sistema de chamado HAL (Hardware Abstraction Layer ou Camada de Abstração de Hardware) garante que os usuários não precisem lidar com este fardo. Se você quiser verificar onde está o seu dispositivo, digite **dmesg** após inseri-lo.



Você então pode copiar os arquivos:

```
robin:~> cp -R /mnt/camera/* images/
```

```
robin:~> umount /mnt/camera
```

Semelhantemente, o drive jazz será montado no /mnt/jazz.

Linhas adequadas deverão ser adicionadas no /etc/modules.conf e /etc/fstab de modo que isto funcione. Procure nos HOWTOs para saber mais. Em sistemas com um kernel 2.6.x ou superior, você também pode querer conferir as páginas man para **modprobe** e modprobe.conf.

## Fazendo backup de dados com um dispositivo de fita

Isto é feito com o comando **tar** (detalhes acima). A ferramenta **mt** é usada para controlar o dispositivo de fita magnética, como o /dev/st0. Livros inteiros foram escritos sobre o backup em fitas, portanto, para saber mais procure em nossa lista de leituras no [Appendix B, Comandos DOS vs comandos Linux](#) para mais informações. Tenha em mente que bases de dados podem precisar de outro tipo de processo de backup por conta de sua estrutura.

Os comandos de backup apropriados em geral são colocados em um dos diretórios *cron* de modo que eles sejam executados rotineiramente. Em ambientes maiores, o suíte [Amanda](#), gratuitamente disponível, ou uma solução comercial podem ser implementadas para fazer o backup de múltiplas máquinas. Trabalhar com fitas, entretanto, é uma tarefa administrativa que encontra-se além do propósito deste guia.

## Ferramentas para a sua distribuição

A maior parte das distribuições do Linux oferece suas próprias ferramentas, o que torna sua vida mais fácil. Um breve panorama:

- SuSE: o YaST agora possui módulos de backup e restauração expandidos.
- RedHat: a ferramenta File Roller possibilita o gerenciamento visual de arquivos (comprimidos). Eles parecem ser favoráveis à ferramenta X-CD-Roast para o transporte dos backups para dispositivos externos.
- Mandrake: X-CD-Roast
- A maior parte das distribuições vêm com os utilitários **dump** e **restore** do BSD para a criação de backups de sistemas de arquivos ext2 e ext3. Esta ferramenta pode escrever para uma larga gama de dispositivos e literalmente despeja (dump) os arquivos ou sistema de arquivos bit por bit no dispositivo especificado. Como no **dd**, isto permite fazer backups de tipos especiais de arquivos como aqueles no /dev.

## Usando o rsync

### Introdução

O programa **rsync** é uma ferramenta rápida e flexível para realizar backups remotos. Ele é comum em sistemas UNIX ou nele baseados, fácil de configurar e usar em scripts. Embora o *r* de **rsync** signifique “remoto”, não tome isto ao pé da letra. Seu dispositivo “remoto” pode bem ser um disco de armazenamento USB ou outra partição em seu disco rígido, não é necessário que existam duas máquinas distintas.

### Um exemplo: rsync em um dispositivo de armazenamento USB

Como discutido no [the section called “Pontos de montagem”](#), primeiro nós teremos que montar o dispositivo. É provável que isto tenha que ser feito como *root*:

```
root@theserver# mkdir /mnt/usbstore
```

```
root@theserver# mount -t vfat /dev/sda1 /mnt/usbstore
```

### Amigável

Uma quantidade crescente de distribuições tem dado acesso à dispositivos removíveis para usuários não-privilegiados e montam aparelhos USB, CD-ROMs e outros removíveis automaticamente.

Note que esta diretriz supõe que seu sistema suporte USB. Veja [o Guia USB](#) para obter informações se isto não funcionar. Verifique com o **dmesg** se o /dev/sda1 é de fato o dispositivo que se quer montar.

Então você poderá começar o backup propriamente dito, digamos do diretório /home/carlos:

```
karl@theserver:~> rsync -avz /home/karl/ /mnt/usbstore
```

Como de costume, para saber mais, leia as páginas man.

# Criptografia

## Notas gerais

### Por que criptografar dados?

Criptografia é sinônimo de segredo. No contexto de backups, a criptografia pode ser muito útil, por exemplo, se você precisar deixar seus dados de backup em um local onde não é possível controlar quem tem acesso, como no servidor do seu provedor.

Além disso, a criptografia pode ser usada em emails também: normalmente, os emails não são criptografados e são enviados abertos na rede ou na internet. Se a sua mensagem contiver dados sigilosos, é melhor criptografá-los.

### GNU Privacy Guard

Nos sistemas Linux você encontra o GnuPG, o GNU Privacy Guard, que é um suíte de programas que são compatíveis com as ferramentas do PGP (Pretty Good Privacy) disponíveis comercialmente.

Neste guia discutiremos somente o uso mais básico das ferramentas de criptografia e mostraremos o que você precisará para gerar uma chave de criptografia e como usá-la para criptografar seus dados, que posteriormente poderão ser armazenados de maneira segura em lugares públicos. As páginas man contém informações para o uso mais avançado dos diversos comandos.

## Gerando uma chave

Antes de começar a criptografar seus dados, é preciso criar um par de chaves. O par consiste em uma chave pública e outra privada. Você pode enviar a chave pública para os correspondentes, que poderão usá-la para criptografar dados para você, os quais você descriptografará com sua chave privada. Mantenha consigo sua chave privada nunca a forneça a terceiros ou eles serão capazes de descriptografar os dados destinados a você. Apenas para garantir que acidentes não aconteçam, a chave privada é protegida com uma senha. O par de chaves é criado com este comando:

```
willy@ubuntu:~$ gpg --key-gen
gpg (GnuPG) 1.4.2.2; Copyright (C) 2005 Free Software Foundation, Inc.
Este programa vem SEM QUAISQUER GARANTIAS.
Ele é software livre e você é incentivado a distribuí-lo
sob algumas condições. Veja o arquivo COPYING para mais detalhes.
```

```
gpg: diretório `/home/willy.gnupg' criado
gpg: novo arquivo de configuração `/home/willy/.gnupg/gpg.conf' criado
gpg: AVISO: as opções do `/home/willy/.gnupg/gpg.conf' ainda não estão
ativas durante esta execução
gpg: chaveiro `/home/willy/.gnupg/secring.gpg' criado
gpg: chaveiro `/home/willy/.gnupg/pubring.gpg' criado
Por favor, selecione o tipo de chave que você deseja:
  (1) DSA e Elgamal (padrão)
  (2) DSA (assinatura somente)
  (5) RSA (assinatura somente)
Sua escolha? 1
0 par de chaves DSA terá 1024 bits.
As chaves ELG-E podem ter entre 1024 e 4096 bits.
Qual o tamanho da chave desejada? (2048) 4096
0 tamanho escolhido foi 4096 bits
Por favor, especifique o prazo de validade da chave.
  0 = chave não expira
  <n> = chave expira em n dias
  <n>w = chave expira em n semanas
  <n>m = chave expira em n meses
  <n>y = chave expira em n anos
A chave é válida por? (0) 0
A chave não expira
Isto está correto? (s/N) s
```

Você precisa de um ID de usuário para identificar a sua chave; o software constrói a chave a partir da sua ID de usuário Nome Real, Comentário e Endereço de Email da seguinte forma:

```
"Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>"
```

Nome real: **Willy De Wandel**

Endereço de email: **wdw@mvg.vl**

Comentário: **Willem**

Você selecionou esta ID de usuário:

```
"Willy De Wandel (Willem) <wdw@mvg.vl>"
```

Mudar (N)ome, (C)omentário, (E)mail or (0)k/(Q)uit? 0

Você precisa de uma Frase Secreta para proteger sua chave secreta.

Frase Secreta:

Agora informe sua senha. Isto pode ser uma frase, quanto maior, melhor, a única condição é que você se lembre dela a qualquer tempo. Para verificação, você precisa informar a mesma frase novamente.

Agora o par de chaves é gerado por um programa que gera número aleatórios e que é, dentre outros fatores, alimentado pela atividade dos dados no sistema. Então é uma boa idéia iniciar alguns programas agora, mover o cursor do mouse ou digitar alguns caracteres aleatórios na janela do terminal. Desta forma, as chances de gerar números que contém uma grande variedade de dígitos aumentará e a chave será mais difícil de ser desvendada.

## Sobre a sua chave

Quando a sua chave tiver sido criada, você receberá uma mensagem sobre a *digital*. Esta é uma seqüência de 40 números hexadecimais, que é tão extensa que é muito, muito difícil gerar a mesma chave duas vezes, em qualquer computador. Você pode ficar seguro de que esta é uma seqüência única. A forma abreviada desta chave consiste no seu nome, seguido por ao menos 8 números hexadecimais.

Você pode obter informações sobre a sua chave da seguinte maneira:

```
willy@ubuntu:~$ gpg --list-keys
/home/willy/.gnupg/pubring.gpg
-----
pub      1024D/BF5C3DBB 2006-08-08
uid           Willy De Wandel (Willem) <wdw@mvg.vl>
sub      4096g/A3449CF7 2006-08-08
```

O *ID da chave* é "BF5C3DBB". Você pode enviar sua ID de chave e o seu nome para um *servidor de chaves*, de modo que outras pessoas possam obter esta informação sobre você e usá-la para criptografar seus dados. Alternativamente, você pode mandar a chave pública diretamente para as pessoas que precisarem dela. A parte pública da sua chave é a longa série de números que você quando usa a opção `--export` para o comando **gpg**:

```
gpg --export -a
```

Entretanto, para os fins deste guia, assumimos que você somente precisará da sua chave para criptografar e descriptografar dados para você mesmo. Leia as páginas man do **gpg** se quiser saber mais.

## Criptografando seus dados

Agora você pode criptografar um ficheiro `.tar` ou um ficheiro comprimido, antes de salvá-lo para uma mídia de backup ou transportá-lo para o servidor de backup. Use o comando **gpg** como a seguir:

```
gpg -e -r (part of) uid archive
```

A opção `-e` informa o **gpg** para criptografar, a opção `-r` indica para quem criptografar. Tenha em mente que apenas os nomes de usuário que seguem esta opção `-r` poderão descriptografar os dados novamente. Um exemplo:

```
willy@ubuntu:~$ gpg -e -r Willy /var/tmp/home-willy-20060808.tar
```

## Descriptografando arquivos

Usando a opção `-d`, você pode descriptografar arquivos que foi criptografados por você. Os dados serão exibidos na sua tela, mas uma cópia criptografada permanecerá no disco. Então, para formatos de arquivos outros que não texto simples, você provavelmente querará salvar os dados descriptografados, de forma que você possa vê-los usando o programa apropriado. Isto é feito usando a opção `-o` ao comando **gpg**:

```
willy@ubuntu:~$ gpg -d -o /var/tmp/home-willy-decrypt.tar /var/tmp/home-willy-20060808.tar.gpg
```

```
É necessária uma frase de acesso para desbloquear a chave secreta para
o usuário: "Willy De Wandel (Willem) <wdw@mvg.vl>"
chave 4096 ELG-E, ID A3449CF7, criada em 2006-08-08 (ID da chave principal BF5C3DBB)
```

```
gpg: criptografado com chave de 4096-bit ELG-E, ID A3449CF7, criada em 2006-08-08
"Willy De Wandel (Willem) <wdw@mvg.vl>"
```

## Sem senha = sem dados

Se você não lembrar sua senha, os dados estão perdidos. Nem mesmo o administrador do sistema será capaz de descriptografar os dados. É por isto que às vezes uma cópia de chaves importantes é mantida em um cofre de banco.

## Resumo

Aqui está uma lista dos comandos relacionados ao backup de arquivos

**Table 9.1. Novos comandos no capítulo 9: Backup**

Comando	Significado
<b>bzip2</b>	Um compressor de arquivos que os distribui em blocos
<b>cdrecord</b>	Grava áudio ou dados em CDs.

<b>dd</b>	Converte e copia um arquivo
<b>fdformat</b>	Formatação de baixo nível em disquetes
<b>gpg</b>	Criptografando e descriptografando dados
<b>gzip</b>	Um compressor ou expansor de arquivos.
<b>mcopy</b>	Copia arquivos do MSDOS do e para o UNIX
<b>mdir</b>	Exibe um diretório de MSDOS
<b>mformat</b>	Adiciona um sistema de arquivos MSDOS para um disquete formatado em baixo nível
<b>mkbootdisk</b>	Cria um disquete de inicialização para o sistema em execução.
<b>mount</b>	Monta o sistema de arquivos (integra-o com o sistema de arquivos atual conectando-o a um ponto de montagem).
<b>rsync</b>	Sincroniza diretórios
<b>tar</b>	Utilitário de arquivamento em fita, também usado para criar ficheiros em disco ao invés de fita.
<b>umount</b>	Desmonta sistemas de arquivos.

## Exercícios

- Crie uma cópia de backup da sua pasta home no `/var/tmp` usando o comando **tar**. Comprima, então, o arquivo usando o **gzip** ou o **bzip2**. Faça com que o arquivo criado seja limpo, ou seja, não crie uma bagunça quando descompactado.
- Formate um disquete e coloque alguns arquivos da sua pasta home nele. Mude de disquete com outro trainee e recupere os dados do disquete dele(a) na sua pasta home.
- Formate o disquete para o DOS. Use as *mtools* para colar e transportar os arquivos para ele.
- O que acontece com um disquete não formatado quando você tenta montá-lo no sistema de arquivos?
- Se você possuir um dispositivo de armazenamento USB, tente colocar um arquivo nele.
- Usando o **rsync**, crie uma cópia da sua pasta home em outro sistema de arquivos local ou remoto.
- Quando deixar arquivos em uma rede de servidores, é melhor criptografá-los. Crie um ficheiro tar da sua pasta home e criptografe-a.

## Chapter 10. Redes

### Table of Contents

#### [Panorama de Redes](#)

[O modelo OSI](#)

[Alguns protocolos populares de rede](#)

#### [Configurações e informações sobre a rede](#)

[Configuração de interfaces de rede](#)

[Arquivos de configuração de rede](#)

[Comandos de configuração de rede](#)

[Nomes das interfaces de rede](#)

[Verificando a configuração da máquina com o `netstat`](#)

[Outras máquinas](#)

#### [Aplicativos Internet/Intranet](#)

[Tipos de servidores](#)

[Email](#)

[Web](#)

[File Transfer Protocol \(Protocolo de Transferência de Arquivos\)](#)

[Conferências e Bate-papo](#)

[Serviços de notícias](#)

[O Domain Name System](#)

[DHCP](#)

[Serviços de autenticação](#)

#### [Execução remota de aplicativos](#)

[Introdução](#)

[Rsh, rlogin e telnet](#)

[O Sistema de Janelas X](#)

[O suíte SSH](#)

[VNC](#)

[O protocolo rdesktop](#)

## [Segurança](#)

[Introdução](#)

[Serviços](#)

[Atualize o sistema com regularidade](#)

[Firewalls e política de acesso](#)

[Detecção de invasões](#)

[Mais dicas](#)

[Eu fui invadido?](#)

[Recuperando-se de uma invasão](#)

## [Resumo](#)

## [Exercícios](#)

[Redes em geral](#)

[Conexões remotas](#)

[Segurança](#)

## Abstract

No que diz respeito a redes, o Linux é o sistema operacional de escolha, não apenas porque o trabalho em rede esteja intimamente ligado à estrutura do SO e exista uma grande variedade de ferramentas disponíveis, mas por sua robustez sob enormes cargas, a qual somente pôde ser atingida após anos de correções e testes em um projeto Open Source.

Prateleiras cheias de livros com informações sobre o Linux e as redes foram escritas, mas tentaremos fornecer um panorama neste capítulo. Após terminar de lê-lo, você saberá mais sobre:

- Protocolos de rede suportados
- Arquivos de configuração de rede
- Comandos para configuração e teste da rede
- Daemons e clientes que permitem diferentes usos da rede
- Compartilhamento de arquivos e impressão
- Execução remota de comandos e aplicativos
- Interconexão básica de redes
- Execução segura de aplicativos remotos
- Firewalls e detecção de intrusos

## Panorama de Redes

### O modelos OSI

Um protocolo é, em termos simples, um conjunto de regras para comunicação.

Para transmitir dados em uma rede, por exemplo, um email do seu computador para outro computador do outro lado do mundo, diversos tipos de hard- e softwares precisam trabalhar em conjunto.

Todos estes pedaços de hardware e diferentes softwares falam línguas diferentes. Imagine o seu programa de email: ele é capaz de falar com o sistema operacional do seu computador por intermédio de um protocolo específico, mas não é capaz de comunicar-se com o hardware. É preciso um programa especial no sistema operacional para realizar esta tarefa. Em troca, o computador precisa ser capaz de comunicar-se com a linha de telefone ou outro método de ligação com a Internet. Nos bastidores, os hardwares de conexão em rede precisam ser capazes de se comunicar para que o seu email seja transmitido de um ponto a outro até o destino final da mensagem.

Todos estes diferentes tipos de protocolos de comunicação são classificados em 7 camadas, que são conhecidas como *Open Systems Interconnection Reference Model*, ou *OSI Model* na forma abreviada. Para facilitar a compreensão, este modelo é reduzido para a descrição de um protocolo com 4 camadas, como abaixo descrito:

**Table 10.1. O modelo OSI simplificado**

Nome da camada	Protocolos das Camadas
Camada dos Aplicativos	HTTP, DNS, SMTP, POP, ...
Camada de transporte	TCP, UDP
Camada de Rede	IP, IPv6
Camada de acesso à rede	PPP, PPPoE, Ethernet

Cada camada pode usar apenas a funcionalidade da camada abaixo; cada camada pode apenas exportar funcionalidades para a camada acima. Em outras palavras: as camadas se comunicam apenas com aquelas que lhes forem adjacentes. Usemos novamente como exemplo a sua mensagem de email: você a informa através da camada de aplicativo. No seu computador, ela desce para a camada de transporte e rede. Seu computador coloca-a na rede através da camada de acesso à rede. Esta é também a camada que moverá a mensagem ao redor do mundo. No seu destino, o computador que a receber a aceitará por meio de sua própria camada de rede e irá exibi-la para o destinatário usando a camada de transporte e aplicativos.

## A realidade é muito mais complexa

A seção acima e como as que se seguem foram incluídas porque você cedo ou tarde se deparará com termos de rede; elas lhe fornecerão alguns pontos de partida, caso você queira aprender em detalhes.

## Alguns protocolos populares de rede

O Linux suporta muitos diferentes protocolos de rede. Listaremos apenas os mais relevantes.

### TCP/IP

O *Transport Control Protocol* e o *Internet Protocol* são os meios mais populares de comunicação na internet. Muitos aplicativos, como o seu navegador e cliente de emails, são desenvolvidos em cima deste suíte de protocolos.

Em uma abordagem simplista, o IP fornece uma solução para o envio de pacotes de informação de uma máquina à outra, enquanto o TCP garante que os pacotes sejam organizados em fluxos, de modo que pacotes de diferentes aplicativos não se misturem e sejam enviados na ordem correta.

Um bom ponto de partida para aprender mais sobre TCP e IP são os seguintes documentos:

- **man 7 ip**: Descreve a implementação do protocolo IPv4 no Linux (sendo a versão 4 a que atualmente se encontra mais difundida).
- **man 7 tcp**: Implementação do protocolo TCP.
- RFC793, RFC1122, RFC2001 para o TCP e RFC791, RFC1122 e RFC1112 para o IP.

Os documentos [Request For Comments](#) contém descrições dos padrões de redes, protocolos, aplicativos e implementações. Estes documentos são gerenciados pelo Internet Engineering Task Force, uma comunidade internacional que cuida de manter a internet operacional e também do desenvolvimento e evolução de sua arquitetura.

O seu provedor de acesso normalmente tem um arquivo RFC disponível ou você pode navegar por eles através do <http://www.ietf.org/rfc.html>.

### TCP/IPv6

Ninguém esperava que a internet crescesse tão rápido quanto aconteceu. O IP comprovou ter muitas vantagens quando um número realmente elevado de computadores está em rede, a mais importante sendo a disponibilidade de endereços únicos para cada máquina na rede. Por conta disto, foi concebida a versão 6 do IP, suprimindo as necessidades da internet de hoje.

Infelizmente, nem todos os aplicativos e serviços suportam o IPv6. Uma migração está sendo colocada em prática em muitos ambientes que podem se beneficiar do IPv6. Para alguns aplicativos, o antigo protocolo ainda é usando, para alguns que já foram refeitos a nova versão já se encontra ativa. Portanto, a verificação da sua configuração da rede, às vezes, pode ser um pouco confusa já que todos do tipos de medidas podem ser tomadas para esconder um protocolo do outro para que não venham a se misturar.

Mais informações podem ser encontradas nos seguintes documentos:

- **man 7 ipv6**: a implementação Linux do protocolo IPv6.
- RFC1883 descreve o protocolo IPv6

### PPP, SLIP, PLIP, PPPOE

O kernel do Linux tem suporte interno para PPP (Protocolo Ponto-a-Ponto), SLIP (Serial Line IP), PLIP (Parallel Line IP) e PPPP sobre EThernet. PPP é o modo mais popular de usuários individuais acessarem seus provedores de serviço de internet, embora em regiões muito povoadas ele seja substituído pelo PPOE, o protocolo usado para conexões ADSL (Asymmetric Digital Subscriber Line).

A maior parte das distribuições do Linux provê ferramentas amigáveis para configurar a conexão com a internet. Serão necessários apenas um nome de usuário e uma senha para se conectar com o seu provedor de acesso, bem como uma linha de telefone no caso do PPP. Estes dados são informados na ferramenta de configuração gráfica, que provavelmente também permitirá iniciar ou encerrar a conexão com o provedor.

## ISDN

O kernel do Linux possui capacidade ISDN interna. O `Idsn4linux` controla cartões PC de ISDN e pode emular um modem com o conjunto de comandos Hayes (comandos "AT"). É possível desde um simples uso de terminal até uma conexão completa com a internet.

Verifique a documentação para o sistema.

## AppleTalk

AppleTalk é o nome da pilha de conexões da Apple. Ele permite redes p2p que fornecem funcionalidades básicas como o compartilhamento de arquivos e impressora. Cada máquina pode simultaneamente agir como um cliente e servidor, e os softwares e hardwares necessários são fornecidos com todos os computadores Apple.

O Linux fornece comunicação completa com o AppleTalk. O `Netatalk` é uma implementação do kernel do suíte de protocolos AppleTalk, originalmente desenvolvido para sistemas derivados do BSD. Ele inclui suporte para roteamento AppleTalk, servindo sistemas de arquivos UNIX e AFS através do `AppleShare` e servindo impressoras UNIX e acessando impressoras AppleTalk.

## SMB/NMB

Para compatibilidade com ambientes MS Windows, o suíte Samba, que inclui suporte para os protocolos NMB, pode ser instalado em qualquer sistema derivado do UNIX. O protocolo Server Message Block (também chamado de Session Message Block, NetBIOS ou protocolo LanManager) é usado no MS Windows 3.11, NT, 95/98, 2000 e XP para compartilhar discos e impressoras.

As funções básicas do suíte Samba são: compartilhar drives do Linux com máquinas Windows, acessando partes do SMB das máquinas Linux, compartilhando impressoras do Linux com máquinas Windows e vice-versa.

A maior parte das distribuições do Linux fornece um pacote *samba*, que realiza a maior parte da configuração do servidor e os comandos **smbd**, o servidor Samba, e **nmbd**, o servidor de nomes netbios, são iniciados junto com o sistema. O Samba pode ser configurado no modo gráfico, pela interface web ou pela linha de comando por meio dos arquivos de configuração. Os daemons fazem com que a máquina Linux pareça ser uma máquina MS Windows na janela Meus Locais da Rede/Rede Vizinha; um compartilhamento vindo da máquina Linux será indistinto de qualquer compartilhamento proveniente de uma máquina MS Windows.

Mais informações pode ser encontradas nos seguintes locais:

- **man smb.conf**: que descreve o formato do principal arquivo de configuração do Samba.
- O [Samba Project Documentation](#) (ou verifique o seu espelho local do [samba.org](#)) contém um guia de instalação e testes de fácil compreensão, que também explica como configurar o seu servidor Samba como um Controlador Primário de Nome de Domínio. Outrossim, as páginas man também se encontram disponíveis aqui.

## Protocolos diversos

O Linux também possui suporte para Radio Amador, interconexão WAN (X25, Frame Relay, ATM), infravermelho e outros tipos de conexões sem fio, mas uma vez que estes protocolos normalmente necessitam de hardwares especiais, não os discutiremos neste documento.

# Configurações e informações sobre a rede

## Configuração de interfaces de rede

Todas as grandes distribuições do Linux vêm com diversas ferramentas gráficas, que permitem uma configuração simples do computador em uma rede local, para conectá-lo a um servidor de acesso ou para acesso sem fio. Estas ferramentas podem ser iniciadas a partir da linha de comando ou de um menu:

- A configuração do Ubuntu é feita selecionando Sistema → Administração → Redes.
- O RedHat Linux vem com o **redhat-config-network**, que possui interface gráfica e texto.
- O YAST e o YAST2 do Suse são ferramentas de configuração multi-tarefa.
- O Mandrake/Mandriva vem com o Assistente de Configuração de Redes e Internet, que preferencialmente é iniciado a partir do Painel de Controle do Mandrake.
- Em sistemas Gnome: **gnome-network-preferences**.
- Em sistemas KDE: **knetworkconf**.

A documentação do sistema fornece muitas dicas e informações sobre a disponibilidade e uso destas ferramentas.

Informações que você precisará fornecer:

- Para se conectar à rede local, por exemplo, aos seus computadores pessoais ou no trabalho: nome da máquina, nome de domínio e endereço de IP. Se você quer configurar a sua própria rede, o ideal é fazer algumas leituras. No trabalho, estas informações provavelmente serão fornecidas ao seu computador automaticamente quando da inicialização. Em caso de dúvida, é melhor não especificar nenhuma informação ao invés de inventá-las.
- Para conecta-se à internet: nome de usuário e senha para o seu provedor de acesso, número de telefone, se estiver usando um modem. Em geral os provedores de acesso automaticamente definem um endereço de IP e tudo o que for necessário para que seus aplicativos de internet funcionem.

## Arquivos de configuração de rede

As ferramentas de assistência gráfica editam um conjunto específico de arquivos, por meio de alguns comandos básicos. Os nomes exatos dos arquivos de configuração e sua localização no sistema de arquivos depende em grande parte da distribuição e versão do Linux em uso. Entretanto, alguns arquivos de configuração são comuns a todos os sistemas UNIX:

### /etc/hosts

O arquivo `/etc/hosts` sempre contém o endereço IP da *máquina local*, `127.0.0.1`, que é usado para a comunicação de processos entre processos. Nunca remova esta linha! Algumas vezes há endereços de máquinas adicionais, que podem ser contatadas sem o uso de um serviço de nomeação externo, como o DNS (o Domain Name Server ou Servidor de Nomes de Domínio).

Um exemplo de arquivo `hosts` para uma rede doméstica:

```
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1    localhost.localdomain  localhost
192.168.52.10    tux.mylan.com            tux
192.168.52.11    winxp.mylan.com          winxp
```

Leia mais no **man** `hosts`.

### /etc/resolv.conf

O arquivo `/etc/resolv.conf` configura o acesso a um servidor DNS, veja o [the section called "O Domain Name System"](#). Este arquivo contém seu nome de domínio e o nome do servidor a ser contatado:

```
search mylan.com
nameserver 193.134.20.4
```

Leia mais na página `man` do `resolv.conf`.

### /etc/nsswitch.conf

O arquivo `/etc/nsswitch.conf` define a ordem em que se deve contatar os diferentes serviços de nome. Para uso da internet, é importante que o `dns` apareça na linha "máquinas":

```
[bob@tux ~] grep hosts /etc/nsswitch.conf
hosts:          files dns
```

Isto instrui o seu computador a procurar por nomes de máquinas ou endereços de IP primeiro no arquivo `/etc/hosts` e contatar o servidor DNS se uma dada máquina não se encontrar no arquivo `hosts` local. Outros serviços de nome que podem ser contatados são o LDAP, o NIS e o NIS+.

Mais em **man** `nsswitch.conf`.

## Comandos de configuração de rede

### O comando ip

Os scripts específicos das distribuições e as ferramentas gráficas são interfaces para o **ip** (ou **ifconfig** e **route** em sistemas mais antigos) para exibir e modificar as configurações de rede do kernel.

O comando **ip** é usado para definir endereços de IP para as interfaces, para configurar rotas para a internet e outras redes, exibir as configurações TCP/IP e etc.

Os comandos a seguir exibem os endereços de IP e informações sobre rotas:

```
benny@home benny> ip addr show
1: lo: <LOOPBACK,UP> mtu 16436 qdisc noqueue
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 brd 127.255.255.255 scope host lo
   inet6 ::1/128 scope host
2: eth0: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast qlen 100
   link/ether 00:50:bf:7e:54:9a brd ff:ff:ff:ff:ff:ff
   inet 192.168.42.15/24 brd 192.168.42.255 scope global eth0
```



```
inet6 fe80::250:bfff:fe7e:549a/10 scope link
```

```
benny@home benny> ip route show
192.168.42.0/24 dev eth0 scope link
127.0.0.0/8 dev lo scope link
default via 192.168.42.1 dev eth0
```

### Informações importantes:

- duas interfaces de rede, mesmo em um sistema que tem apenas um cartão de interface de rede: “lo” é o loop local, usado para comunicações internas da rede; “eth0” é um nome comum para um interface *real*. Jamais modifique a configuração do loop local ou a sua máquina começará a apresentar problemas! Interfaces de redes sem fio normalmente são definidas como “wlan0”; interfaces de modem como “ppp0”, mas também podem existir outros nomes.
- Endereços de IP, marcados com “inet”: o loop local sempre têm 127.0.0.1, a interface física pode ter quaisquer outras combinações.
- Os endereços de hardware da sua interface, que podem ser exigidos como parte do procedimento de autenticação para se conectar a uma rede, é marcado como “ether”. O loop local tem 6 pares de todos os zeros, o loop físico tem 6 pares de caracteres hexadecimais, dentre os quais os três primeiros são específicos do comerciante.

### O comando ifconfig

Embora o **ip** seja o meio mais atual de configurar um sistema Linux, o **ifconfig** ainda permanece popular. Use-o sem opções para exibir as informações da interface de rede:

```
els@asus:~$ /sbin/ifconfig
eth0      Link encap:Ethernet  HWaddr 00:50:70:31:2C:14
          inet addr:60.138.67.31  Bcast:66.255.255.255  Mask:255.255.255.192
          inet6 addr: fe80::250:70ff:fe31:2c14/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:31977764 errors:0 dropped:0 overruns:0 frame:0
          TX packets:51896866 errors:0 dropped:0 overruns:0 carrier:0
          collisions:802207 txqueuelen:1000
          RX bytes:2806974916 (2.6 GiB)  TX bytes:2874632613 (2.6 GiB)
          Interrupt:11 Base address:0xec00

          lo          Link encap:Local Loopback

          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:765762 errors:0 dropped:0 overruns:0 frame:0
          TX packets:765762 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:624214573 (595.2 MiB)  TX bytes:624214573 (595.2 MiB)
```

Aqui, também, apontam-se os aspectos mais importantes da configuração da interface:

- O endereço de IP é marcado com “inet addr”.
- O endereço de hardware segue a etiqueta “HWaddr”.

Ambos **ifconfig** e **ip** exibem informações mais detalhadas da configuração e muitas estatísticas sobre cada interface, bem como, provavelmente o mais importante, se a rede está “UP” e “RUNNING”.

### Comandos PCMCIA

No seu laptop, o qual você em geral conecta à rede da companhia usando a conexão Ethernet, mas que você agora terá que configurar para uma conexão discada em casa ou em um hotel, é possível que precise ativar o cartão PCMCIA. Isto é feito usando o utilitário de controle **cardctl** ou o **pccardctl** em distribuições mais novas.

Um exemplo de uso

```
cardctl insert
```

Agora é possível configurar o cartão, tanto usando a interface gráfica como a linha de comando. Antes de retirar o cartão, use este comando:

```
cardctl eject
```

Entretanto, uma boa distribuição deveria fornecer suporte PCMCIA nas ferramentas de configuração, poupando os usuários da necessidade de executar comandos PCMCIA manualmente.

### Mais informações

Discussões mais aprofundadas sobre a configuração de redes fogem ao escopo deste documento. Sua primeira fonte de informações adicionais são as páginas man para os serviços que você deseja configurar. Leituras complementares:

- [The Modem-HOWTO](#): Ajuda para selecionar, conectar, configurar, corrigir erros e entender modems analógicos para um PC.
- [LDP HOWTO Index, section 4.4](#): lista de COMO FAZER dividida em categorias sobre redes em geral, protocolos, conexão discada, DNS, CPNs, pontes, roteamento, segurança e mais.
- A maior parte dos sistemas tem uma versão do arquivo `ip-cref` (para localizá-lo use o comando **locate**); para ler o formato PS deste documento pode-se usar o **gv**.

## Nomes das interfaces de rede

Em uma máquina Linux, o dispositivo de nome `lo` ou *local loop* é vinculado ao endereço interno 127.0.0.1. O computador terá dificuldades em fazer com que seus aplicativos funcionem se este dispositivo não estiver presente; ele sempre está lá, mesmo nos computadores que não estão em rede.

O primeiro dispositivo ethernet, `eth0` no caso de um cartão de interface de rede padrão, aponta para o seu endereço de LAN IP local. Máquinas clientes locais têm somente um cartão de interface de rede. Roteadores, que conectam redes, possuem um dispositivo de rede para cada rede que servem.

Se você usa um modem para conectar-se à internet, seu dispositivo de rede provavelmente se chamará `ppp0`.

Existem muitos outros nomes, por exemplo as interfaces Virtual Private Network (VPNs), e interfaces múltiplas podem estar simultaneamente ativas, de modo que a saída dos comandos **ifconfig** ou **ip** pode torna-se um tanto extensa se não forem informadas opções. Mesmo interfaces múltiplas do mesmo tipo podem estar ativas. Neste caso, elas são numeradas seqüencialmente: a primeira recebe o número zero, a segunda o sufixo 1, a terceira 2 e assim por diante. Este é o caso de muitos servidores de aplicativos, em máquinas que possuem uma configuração failover, roteadores, firewalls e muitos outros.

## Verificando a configuração da máquina com o netstat

Além do comando **ip** para exibir a configuração da rede, existe o comando **netstat** que possui muitas opções e é muito útil em quaisquer sistemas UNIX.

Informações sobre roteamento podem ser exibidas com a opção `-nr` para o comando **netstat**:

```
bob:~> netstat -nr
Kernel IP routing table
Destination Gateway Genmask Flags MSS Window irtt Iface
192.168.42.0 0.0.0.0 255.255.255.0 U 40 0 0 eth0
127.0.0.0 0.0.0.0 255.0.0.0 U 40 0 0 lo
0.0.0.0 192.168.42.1 0.0.0.0 UG 40 0 0 eth0
```

Esta é uma típica máquina cliente em uma rede IP. Ela possui apenas um dispositivo de rede, o `eth0`. O `lo` é o loop local.

### O modo moderado

O novo modo de obter estas informações do seu sistema é através do comando **ip**:

```
ip route show
```

Quando esta máquina tente contactar uma outra em uma rede externa, indicada na linha iniciada por 0.0.0.0, irá enviar os pedidos de conexão para a máquina (roteador) com endereços de IP 192.168.42.1, e irá usar sua interface primária, `eth0`, para fazê-lo.

Máquinas que estejam na mesma rede, a linha iniciada por 192.168.42.0, também serão contactadas através da interface de rede primária, mas não será necessária a intermediação do roteador, os dados serão colocados diretamente na rede.

Podem haver tabelas de roteamento muito mais complexas que esta, com muitos pares diferentes de "Destinos-Portas" para conectarem-se a diferentes redes. Se você tiver a chance de conectar-se a um servidor de aplicativos, por exemplo em seu trabalho, é uma experiência muito instrutiva verificar as configurações de roteamento.

## Outras máquinas

Uma quantidade impressionante de ferramentas são focadas no gerenciamento de redes e administração remota de máquinas Linux. O seu espelho local de softwares Linux fornecerá diversos deles. Discuti-las nos levaria muito longe do propósito deste documento, portanto, procure pela documentação específica de cada programa.

Discutiremos nesta seção somente algumas ferramentas texto comuns do UNIX/Linux.

### O comando host

Para exibir informações sobre máquinas ou domínios, use o comando **host**:

```
[emmy@pc10 emmy]$ host www.eunet.be
www.eunet.be. has address 193.74.208.177
```

```
[emmy@pc10 emmy]$ host -t any eunet.be
eunet.be. SOA dns.eunet.be. hostmaster.Belgium.EU.net.
 2002021300 28800 7200 604800 86400
eunet.be. mail is handled by 50 pophost.eunet.be.
eunet.be. name server ns.EU.net.
eunet.be. name server dns.eunet.be.
```

Informações similares podem ser exibidas usando o comando **dig**, que fornece dados adicionais sobre como as gravações estão armazenadas no servidor de nomes.

## O comando ping

Para saber se uma máquina está ativa, use o **ping**. Se o seu sistema estiver configurado para enviar mais de um pacote, interrompa o comando **ping** com a combinação das teclas **Ctrl+C**:

```
[emmy@pc10 emmy]$ ping a.host.be
PING a.host.be (1.2.8.3) from 80.20.84.26: 56(84) bytes of data.
64 bytes from a.host.be(1.2.8.3):icmp_seq=0 ttl=244 time=99.977msec
--- a.host.be ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max/mdev = 99.977/99.977/99.977/0.000 ms
```

## O comando traceroute

Para saber qual o trajeto percorrido pelos pacotes até uma máquina em rede, use o comando **traceroute**:

```
[emmy@pc10 emmy]$ /usr/sbin/traceroute www.eunet.be
traceroute to www.eunet.be(193.74.208.177),30 hops max,38b packets
 1 blob (10.0.0.1)
   0.297ms  0.257ms  0.174ms
 2 adsl-65.myprovider.be (217.136.111.1)
   12.120ms 13.058ms 13.009ms
 3 194.78.255.177 (194.78.255.177)
   13.845ms 14.308ms 12.756ms
 4 gigabitethernet2-2.intl2.gam.brussels.skynet.be (195.238.2.226)
   13.123ms 13.164ms 12.527ms
 5 pecbru2.car.belbone.be (194.78.255.118)
   16.336ms 13.889ms 13.028ms
 6 ser-2-1-110-ias-be-vil-ar01.kpnbelgium.be (194.119.224.9)
   14.602ms 15.546ms 15.959ms
 7 unknown-195-207-939.eunet.be (195.207.93.49)
   16.514ms 17.661ms 18.889ms
 8 S0-1-0.Leuven.Belgium.EU.net (195.207.129.1)
   22.714ms 19.193ms 18.432ms
 9 dukat.Belgium.EU.net (193.74.208.178) 22.758ms * 25.263ms
```

Em alguns sistemas, o **traceroute** foi renomeado para **tracpath**.

## O comando whois

Pode-se consultar informações sobre nomes de domínios usando o comando **whois**, como explicado em muitos servidores **whois**, como no exemplo abaixo:

```
[emmy@pc10 emmy]$ whois cnn.com
[whois.crsnic.net]

Whois Server Version 1.3

    $<--snap server message-->

Domain Name: CNN.COM
Registrar: NETWORK SOLUTIONS, INC.
Whois Server: whois.networksolutions.com
Referral URL: http://www.networksolutions.com
Name Server: TWDNS-01.NS.AOL.COM
Name Server: TWDNS-02.NS.AOL.COM
Name Server: TWDNS-03.NS.AOL.COM
Name Server: TWDNS-04.NS.AOL.COM
Updated Date: 12-mar-2002
>>> Last update of whois database: Fri, 5 Apr 2002 05:04:55 EST <<<

The Registry database contains ONLY .COM, .NET, .ORG, .EDU domains
and Registrars.

[whois.networksolutions.com]

    $<--snap server message-->

Registrant:
Turner Broadcasting (CNN-DOM)
 1 CNN Center
 Atlanta, GA 30303
```

Domain Name: CNN.COM

Administrative Contact:

\$<--snap contactinfo-->

Technical Contact:

\$<--snap contactinfo-->

Billing Contact:

\$<--snap contactinfo-->

Record last updated on 12-Mar-2002.

Record expires on 23-Sep-2009.

Record created on 22-Sep-1993.

Database last updated on 4-Apr-2002 20:10:00 EST.

Domain servers in listed order:

TWDNS-01.NS.AOL.COM	149.174.213.151
TWDNS-02.NS.AOL.COM	152.163.239.216
TWDNS-03.NS.AOL.COM	205.188.146.88
TWDNS-04.NS.AOL.COM	64.12.147.120

Para outros nomes de domínio que não .com, .net, .org and .edu, pode ser necessário especificar o servidor whois, como neste para domínios .be:

`whois domain.be@whois.dns.be`

## Aplicativos Internet/Intranet

O sistema Linux é uma ótima plataforma para oferecer serviços de rede. Nesta seção, tentaremos fornecer um panorama dos servidores e aplicativos de rede mais comuns.

### Tipos de servidores

#### Servidores autônomos

A oferta de um serviço aos usuários pode ser enfrentada de dois modos. Um daemon ou serviço pode ser executado autonomamente ou pode ser dependente de outro serviço a ser ativado.

Serviços de rede que são de uso contínuo ou intenso, normalmente executados de modo autônomo: eles são programas independentes que estão sempre em execução. Via de regra são inicializados juntamente com o sistema e aguardam por conexões em pontos ou portas específicas para os quais foram configurados a ouvir. Quando um pedido aparece, ele é processado e a audição continua até o próximo pedido. Um servidor web é um exemplo típico: você quer estar disponível 24h por dia e se for muito solicitado, o melhor é que existam muitas instâncias de audição para servir a diversos usuários simultaneamente. Outros exemplos são grande repositórios de softwares como o [Sourceforge](#) ou o [your Tucows mirror](#), que diariamente precisam lidar com milhares de pedidos FTP.

Um exemplo de serviço de rede autônomo no seu computador pessoal pode ser o **named** (daemon nome), um servidor que efetua caching de nomes. Serviços autônomos têm seus próprios processos em execução, é possível consultá-los usando o **ps**:

```
bob:~> ps auxw | grep named
named  908  0.0  1.0 14876 5108 ?    S   Mar14  0:07 named -u named
```

Entretanto, existem alguns serviços que você pode usar no seu PC, mesmo que não exista um processo servidor em execução para aquele serviço. Como exemplo pode-se citar o FTP, o serviço de cópia segura ou o serviço dedo. Estes serviços são ouvidos pelo Internet Daemon (**inetd**).

#### (x)inetd

No seu computador pessoal as coisas são em geral mais tranquilas. Você pode ter uma pequena rede, por exemplo, e ter que transferir arquivos de um PC para outro de tempos em tempos, por meio do FTP ou do Samba (para conexão com máquinas Windows). Nestes casos, iniciar todos os serviços que só ocasionalmente serão necessários e tê-los em execução durante todo o tempo seria um desperdício de recursos. Assim sendo, em conjuntos menores você encontrará os daemons necessários em dependência de um programa central, que escuta em todas as portas dos serviços pelos quais é responsável.

Este super-servidor, o daemon de serviços de internet, é iniciado juntamente com o sistema. Duas são as implementações deste serviço: **inetd** e **xinetd** (o daemon de serviços de internet eXtendido). Um ou outro estão sempre em execução em qualquer sistema Linux:

```
bob:~> ps -ef | grep inet
root  926  1 0 Mar14 ?    00:00:00 xinetd-ipv6 -stayalive -reuse \
-pidfile /var/run/xinetd.pid
```

Os serviços pelos quais o daemon de internet é responsável encontram-se listados em seu arquivo de configuração, `/etc/inetd.conf`, para o **inetd**, e o directory `/etc/xinetd.d` para o **xinetd**. Serviços comumente gerenciados incluem o compartilhamento de arquivos e impressoras, SSH, FTP, telnet, o daemon de configuração do Samba, serviços de voz e tempo.

Assim que o pedido de conexão é recebido, o servidor central irá iniciar uma instância do serviço desejado. Assim, no exemplo abaixo, quando o usuário *bob* inicia uma sessão FTP na máquina local, um daemon FTP permanece em execução enquanto a sessão estiver ativa.

```
bob:~> ps auxw | grep ftp
bob      793  0.1  0.2 3960 1076 pts/6    S   16:44   0:00 ncftp localhost
ftp      794  0.7  0.5 5588 2608 ?          SN  16:44   0:00 ftpd:
localhost.localdomain: anonymous/bob@his.server.com: IDLE
```

Obviamente o mesmo se dá quando é iniciada uma conexão com uma máquina remota: ou o daemon responde diretamente ou um **(x)inetd** remoto inicia o serviço necessário e o encerra quando você sai.

## Email

### Servidores

O Sendmail gerente de emails server padrão ou o Mail Transport Agent para plataformas UNIX. Ele é robusto, escalonável e, desde que propriamente configurado com hardware adequado, pode manipular milhares de usuários sem suar. Mais informações sobre a configuração do Sendmail podem ser encontradas nos pacotes *dendmail* e *sendmail-cf*, nos quais você pode consultar os arquivos *LEIAME* E *LEIAME.cf* no `/usr/share/doc/sendmail`. O **man sendmail** e o **man aliases** também podem ser úteis.

O Qmail é outro servidor de emails que vem ganhando popularidade por alegar maior segurança que o Sendmail. Enquanto o Sendmail é um programa monolítico, o Qmail é formado por pequenos programas que interagem entre si e podem ser melhor controlados. Há também o Postfix que vem crescendo em popularidade.

Estes servidores manipulam listas de emails, filtros, procuram por vírus e muito mais. Anti-vírus livres e comerciais estão disponíveis para o Linux. Exemplos de softwares de listagem de emails são o Mailman, o Listserv, o Majordomo e o EZmlm. Procure no site do seu anti-vírus predileto para saber sobre suporte para Linux. O Amavis e o Spamassassin são implementações livres de anti-vírus e anti-spams.

### Servidores de email remotos

Os protocolos mais populares para acesso remoto de emails são *POP3* e *IMAP4*. Ambos, IMAP e POP, permitem operações offline, acesso remoto a novos emails e dependem de um servidor SMTP para enviar emails.

Embora o POP seja um protocolo simples, de fácil implementação e suportado por quase todos os clientes de email, é preferível o uso do IMAP, porque:

- Ele pode manipular mensagens reiteradas de flags de status.
- Ele pode tanto armazenar quanto buscar mensagens de email.
- Ele pode acessar e manipular múltiplas caixas de email.
- Ele suporta atualizações concomitantes e caixas de emails compartilhadas.
- É adequado para o acesso de mensagens Usenet e outros documentos.
- O IMAP funciona tanto online quanto offline.
- ele é otimizado para a performance online, especialmente em ambientes de velocidade reduzida.

### Gerenciadores de emails

Existem muito clientes de email tanto no modo texto como no modo gráfico, listaremos apenas alguns deles. Escolha o de sua preferência.

O comando UNIX **mail** está em cena há alguns anos, antes mesmo das redes existirem. Ele é uma interface simples para o envio de mensagens e pequenos arquivos para outros usuários, que podem então salvar a mensagem, redirecioná-la, respondê-la e etc.

Embora não seja mais usado como cliente, o programa **mail** permanece útil, por exemplo, para enviar a saída de um comando para alguém:

```
mail <future.employer@whereIwant2work.com> < cv.txt
```

O leitor de emails **elm** é um aperfeiçoamento muito necessário do **mail** e, também, o é o **pine** (Pine Is Not ELM). O leitor de emails **mutt** é ainda mais recente e oferece recursos como o threading.

Para aqueles que preferem uma interface gráfica para os seus emails (e uma L.E.R. pelo uso contínuo do mouse), existem centenas de opções. As mais populares para os usuários iniciantes são o Mozilla Thunderbird, que possui um anti-spam de fácil configuração e o Evolution, o Clone do MS Outlook. Há também o Kmail, muito popular entre os usuários do KDE.

### Figure 10.1. O leitor de emails e notícias Evolution



Existem também infinitos aplicativos de email web, como Squirrelmail, Yahoo! mail, gmail do Google e o Hotmail.

Um panorama encontra-se disponível no [Linux Mail User HOWTO](#).

A maior parte das distribuições do Linux incluem o **fetchmail**, um utilitário que recupera e encaminha emails. Ele busca emails em seu servidor de emails remoto (POP, IMAP ou outro) e os encaminha para o seu sistema de entregas local. Você pode então manipular os emails recuperados usando clientes de emails normais. Ele pode ser executado na forma de um daemon que pesquisa um ou mais sistemas reiteradas vezes em intervalos específicos. Informações e exemplos sobre o uso podem ser encontrados nas páginas Info; o diretório `/usr/share/doc/fetchmail[-<version>]` contém uma lista completa dos recursos e uma seção de Perguntas Frequentes para iniciantes.

O filtro **procmail** pode ser usado pra a filtragem de emails recebidos, para criar listas de emails, pré-processamento de mensagens, encaminhamento seletivo e mais. O programa **formail** que o acompanha, dentre outros recursos, permite criar respostas automáticas e a divisão de caixas de entrada. O Procmail está circulando há anos em máquinas UNIX e Linux e é um sistema muito robusto, arquitetado para funcionar mesmo nas circunstâncias mais adversas. Mais informações podem ser encontradas no diretório `/usr/share/doc/procmail[-<version>]` e nas páginas man.

## Uma nota sobre a etiqueta nos correios eletrônicos

Algumas pessoas hoje em dia parecem pensar que as mensagens de email não devem ser muito formais. Isto, é claro, depende. Se você está escrevendo para alguém que não conhece, o melhor é manter alguma distância, da mesma forma que faria em uma carta tradicional. E não se esqueça: as pessoas que você não conhece podem ser homens ou mulheres...

## Web

### O servidor web Apache

O Apache é, de longe, o servidor web mais popular, usado em mais da metade dos servidores web da internet. A maior parte das distribuições do Linux inclui o Apache. Dentre suas vantagens estão a sua arquitetura modular, suporte para SSL, estabilidade e velocidade. Na presença de um equipamento e configuração adequados, ele pode suportar as mais altas cargas.

Nos sistemas Linux, a configuração de servidores é normalmente feita no diretório `/etc/httpd`. O arquivo de configuração mais importante é o `httpd.conf`, que é auto-explicativo. Caso você precise de ajuda, poderá encontrá-la na página man do **httpd** ou no [site do Apache](#).

### Navegadores web

Existe um grande número de navegadores, tanto livres como proprietários, para a plataforma Linux. Já vão longe os tempos em que o Netscape Navigator era a única opção satisfatória, já que o Mozilla/Firefox oferece uma alternativa competitiva que pode rodar em diversos outros sistemas operacionais, como o MS Windows e o MacOS X.

O Amaya é o navegador do W3C. O Opera é um navegador proprietário, compacto e rápido. Muitos gerentes de desktop oferecem recursos de navegação web nos seus gerenciadores de arquivos, como o **nautilus**.

Dentre os navegadores no modo texto, são populares o **lynx** e o **links**. Pode ser que você precise definir servidores de proxy no seu shell, informando as variáveis adequadas. Navegadores textuais são rápidos e úteis quando não há ambiente gráfico disponível, como no contexto dos scripts.

### Servidores proxy

#### O que é um servidor proxy?

Organizações e empresas muitas vezes querem que seus usuários usem servidores proxy. Em especial em ambientes com muitos usuários, um servidor proxy permite downloads mais rápidos das páginas na web. O servidor de proxy armazena páginas web. Quando um usuário solicita uma página que já foi acessada, o servidor proxy apresentará diretamente a página ao usuário, de forma que ele(a) não precise buscá-la na internet, o que requer mais tempo. É claro, podem ser tomadas medidas para que o servidor proxy faça uma rápida verificação e apresente sempre a versão mais atual da página. Em alguns ambientes, o uso de um servidor proxy é compulsório, em outros é possível escolher usá-lo ou não.

#### Configuração de proxy

Se você tiver o nome e porta do servidor proxy, parece óbvio informar estes dados ao navegador. Entretanto, muitos aplicativos (na linha de comando) dependem das variáveis `http_proxy` and `ftp_proxy` para funcionarem corretamente. Pode ser-lhe conveniente adicionar ao seu arquivo `~/.bashrc` uma linha como a seguinte:

```
export http_proxy=http://username:password@proxy_server_name:port_number
```

Por exemplo:

```
export http_proxy=http://willy:Appelsi3ntj3@proxy:80
```

Se não for necessário informar o nome de usuário e a senha, simplesmente deixe tudo aquilo que precede o sinal "@", ele também incluso.

## File Transfer Protocol (Protocolo de Transferência de Arquivos)

### Servidores FTP

Em um sistema Linux, um servidor FTP é normalmente executado a partir do **xinetd**, usando o WU-ftpd, embora o servidor FTP possa ser configurado para execução autônoma em sistemas com tráfego de dados intenso. Veja os exercícios.

Outros servidores FTP incluem o vsftpd, Ncftpd e o Proftpd.

A maior partes das distribuições Linux contém o pacote anonftp, que configura uma árvore de servidores FTP anônimos, com os respectivos arquivos de configuração.

### Clientes FTP

A maior parte das distribuições Linux inclui o **ncftp**, uma versão aperfeiçoada do comando UNIX **ftp**, que é possível que você conheça da linha de comando do Windows. O programa **ncftp** oferece recursos adicionais, como uma interface mais agradável e compreensiva, auto-complemento de nomes de arquivos, funções de retorno e pensamento, favoritos, gerenciamento de sessão e mais:

```
thomas:~> ncftp blob
NcFTP 3.0.3 (April 15, 2001) by Mike Gleason (ncftp@ncftp.com).
Connecting to blob...
blob.some.net FTP server (Version wu-2.6.1-20) ready.
Logging in...
Guest login ok, access restrictions apply.
Logged in to blob.
ncftp / > help
Commands may be abbreviated. 'help showall' shows hidden and
unsupported commands.
'help <command>' gives a brief description of <command>.

ascii      cat        help       lpage     open      quote     site
bgget      cd         jobs       lpwd      page      rename    type
bgput      chmod     lcd        lrename   pdir      rhelp     umask
bgstart    close     lchmod    lrm       pls       rm        version
binary     debug     lls       lrmdir   put       rmdir
bookmark   dir       lmkdir    ls        pwd       set
bookmarks  get       lookup    mkdir     quit      show
ncftp / >
```

Uma ajuda excelente, com muitos exemplos, pode ser encontrada nas páginas man. E, novamente, encontram-se disponíveis diversos aplicativos gráficos.

### O FTP é inseguro!

Não use o FTP para logins não-anônimos a menos que você saiba o que está fazendo. O seu nome de usuário e senha podem ser capturados por usuários mal intencionados! Use o FTP seguro em seu lugar; o **sftp** vem com o suíte Secure SHell, veja o [the section called "Cópia remota segura"](#).

## Conferências e Bate-papo

Diversos clientes e sistemas estão disponíveis em cada distribuição, em substituição ao antiquado bate-papo IRC no modo texto. Uma lista curta e incompleta dos programas mais populares:

- **Pidgin** (antigo Gaim): cliente de mensagens instantâneas multi-protocolo para Linux, Windows e Mac. Compatível com o MSN Messenger, ICQ, IRC e muito mais; veja as páginas Info ou o [site do Pidgin](#) para saber mais.
- **xchat**: Cliente IRC para o sistema de janelas X

### Figure 10.2. X-Chat



A sua página no [SourceForge](#).

- **aMSN**: um clone do MSN
- **Konversation**, **kopete**, **KVIrc** e muitas outras K-ferramentas do suíte KDE.

- **gnomemeeting**: programa de videoconferência para o UNIX (agora Ekiga).
- **jabber**: Plataforma open source de mensagens instantâneas compatível com ICQ, AIM, Yahoo, MSN, IRC, SMTP e muitas outras.
- **psi**: cliente jabber, veja [a página do cliente Jabber PSI](#).
- **skype**: programa para realizar chamadas telefônicas pela internet para outros usuários Skype, veja <http://www.skype.com> para saber mais. O Skype é gratuito, porém não é livre.
- Gizmo: um telefone gratuito (mas não livre) para o seu computador, veja o <http://www.gizmoproject.com>.

## Serviços de notícias

Executar um Usenet server requer muita experiência precisão, então leia a [página do INN](#) para saber mais.

Existem alguns grupos de notícias interessantes na hierarquia *comp.\**, que podem ser acessados por meio de diversos clientes gráficos ou texto. Muitos clientes de email suportam a navegação por grupos de notícias, verifique o seu programa ou procure no seu espelho local de softwares open source como o **tin**, **slrnn** e **mutt** ou então baixe o Mozilla ou qualquer outro aplicativo gráfico.

O [Deja.com](#) mantém um arquivo de todos os grupos de notícias, os quais podem ser buscados pelo motor de pesquisas Google. Esta é uma ferramenta muito poderosa para conseguir ajuda: são altas as chances de que alguém tenha se deparado com o mesmo problema, encontrado a solução e postado em algum dos grupos de notícia.

## O Domain Name System

Todos estes aplicativo precisam de serviços DNS que ligue os endereços de IP com os nomes de domínio e vice-versa. Um servidor de DNS não conhece todos os nomes de domínio do mundo, porém está em rede com outros servidores DNS que podem consultar para encontrar um endereço desconhecido. A maior parte dos sistemas UNIX pode executar o **named**, que é parte do pacote BIND (Berkeley Internet Name Domain) distribuído pelo Internet Software Consortium. Ele pode ser executado como um caching de *nomesdeservidores* autônomo, o que é freqüentemente feito nos sistemas Linux para acelerar a velocidade da conexão.

O principal arquivo de configuração é o `/etc/resolv.conf`, que determina a ordem em que os servidores DNS são contatados.

```
search somewhere.org
nameserver 192.168.42.1
nameserver 193.74.208.137
```

Mais informações podem ser encontradas nas páginas Info do **named**, nos arquivos `/usr/share/doc/bind[-<version>]` e na página do [projeto Bind](#). O [DNS HOWTO](#) disserta sobre o uso do BIND como um servidor DNS.

## DHCP

O DHCP ou Dynamic Host Configuration Protocol vem gradualmente substituindo o bom e velho **bootp** em ambientes maiores. Ele é usado para controlar parâmetros vitais de rede como os endereços de IP e nomes de servidor das máquinas. O DHCP é compatível com o **bootp**. Para configurar o servidor, será necessário ler o HOWTO.

Máquinas clientes DHCP normalmente são configuradas usando uma interface gráfica que determina o **dhcpcd**, o daemon do cliente DHCP. Verifique a documentação do seu sistema se precisar configurar sua máquina como um cliente DHCP.

## Serviços de autenticação

### Tradicionais

Tradicionalmente, os usuários são localmente autenticados, usando as informações armazenadas no `/etc/passwd` e no `/etc/shadow` de cada sistema. Porém, mesmo quando se usa um serviço de rede para a autenticação, os arquivos locais sempre estarão presentes para configurar contas do sistema de uso administrativo, como a conta root, contas daemon e também outras contas usadas por programas adicionais.

Estes arquivos são muitas vezes os primeiros candidatos para um exame por invasores, então certifique-se de que as permissões e propriedades estão definidas conforme deveriam.

```
bob:~> ls -l /etc/passwd /etc/shadow
-rw-r--r-- 1 root root 1803 Mar 10 13:08 /etc/passwd
-r----- 1 root root 1116 Mar 10 13:08 /etc/shadow
```

## PAM



O Linux pode usar o PAM, o Pluggable Authentication Module, um método flexível de autenticação UNIX. Vantagens do PAM:

- Um esquema de autenticação comum que pode ser usado por uma larga gama de aplicativos.
- O PAM pode ser implementado com vários aplicativos sem precisar ser recompilá-los para que o suportem.
- Grande flexibilidade e controle das autenticações pelo administrador e desenvolvedor de aplicativos.
- Desenvolvedores não precisam preocupar-se em criar métodos de autenticação em seus softwares, podendo permanecer focados apenas nos detalhes de seus programas.

O diretório `/etc/pam.d` contém os arquivos de configuração do PAM (costumava ser o `/etc/pam.conf`). Cada aplicativo ou serviços possui seu próprio arquivo. Cada linha possui quatro elementos:

- *Module*:
  - `auth`: fornece a autenticação propriamente dita (talvez perguntando e verificando uma senha) e configura credenciais, como a filiação a um grupo ou tickets do Kerberos.
  - `account`: verifica se o usuário tem acesso autorizado (se a conta não expirou, se o usuário pode efetuar o login a esta hora do dia e assim por diante).
  - `password`: usado para configurar senhas
  - `session`: usado após a autenticação do usuário, Este módulo realiza tarefas adicionais que são necessárias ao acesso (por exemplo, montagem da home do usuário ou disponibilizar a caixa de entrada de emails).

A ordem em que os módulos são dispostos, de forma que vários possam ser usados, é muito importante.

- *Control Flags*: informam o PAM quais ações tomar em caso de falha ou sucesso. Valores podem ser `required`, `requisite`, `sufficient` OU `optional`.
- *Module Path*: caminha para o módulo plugável a ser usado, geralmente no `/lib/security`.
- *Arguments*: informações para os módulos

Arquivos sombra de senhas são automaticamente detectados pelo PAM.

Mais informações podem ser encontradas nas páginas man do **pam** ou na página [do projeto Linux-PAM](#).

## LDAP

O Lightweight Directory Access Protocol é um sistema cliente-servidor para acessar serviços de diretórios locais ou remotos em uma rede. No Linux, a implementação OpenLDAP é usada. Ela inclui o **slapd**, um servidor autônomo; o **slurpd**, um servidor autônomo LDAP de replicação; bibliotecas implementando o protocolo LDAP e uma série de utilitários, ferramentas e clientes de demonstração.

O principal benefício de usar o LDAP é a consolidação de certos tipos de informação na sua organização. Por exemplo, todas as diferentes listas de usuários da sua organização podem ser combinadas em um diretório LDAP. Este diretório pode ser consultado por qualquer aplicativo que reconheça o LDAP e precise destas informações. Pode também ser acessado por usuários que necessitem de informações sobre diretórios.

Outro benefício do LDAP ou X.500 Lite inclui a facilidade na sua implementação (comparado ao X.500) e sua bem definida Application Programming Interface (API), o que significa que o número de aplicativos que podem usar o LDAP e suas portas deve crescer num futuro próximo.

Em contrapartida, se você quiser usar o LDAP, precisará de aplicativos que o entendam e tenham a habilidade de usar suas portas. Embora o uso do LDAP tenda apenas a crescer, atualmente não existem muitos aplicativos preparados para o LDAP. Além disso, embora o LDAP não suporte algumas formas de controle de acesso, ele não possui tantos recursos de segurança quanto o X.500.

Uma vez que o LDAP é um protocolo aberto e configurável, ele pode ser usado para armazenar quase qualquer tipo de informação relacionada a uma determinada organização estrutural. Exemplos comuns são lookups de endereços de email, autenticação central em conjunto com o PAM, diretórios para telefone e configuração de bases de dados.

Veja a documentação do seu sistema e as páginas man para conhecer os detalhes sobre comandos relacionados, como o **ldapmodify** e o **ldapsearch**. Mais informações podem ser encontradas no [LDAP Linux HOWTO](#), que discute a instalação, configuração, execução e manutenção de um servidor LDAP no Linux. A autora desta Introdução ao Linux também escreveu o [LDAP Operations HOWTO](#), descrevendo o conhecimento básico necessário para gerenciar o LDAP, suas operações e a integração de seus serviços.

## Execução remota de aplicativos

### Introdução

Existem diferentes formas de executar comandos ou programas em uma máquina remota e receber a saída, seja ela texto ou gráfica, em sua estação de trabalho. As conexões podem ser seguras ou inseguras. Embora seja obviamente recomendável usar conexões seguras ao invés de transportar sua senha por uma rede aberta, discutiremos algumas aplicações práticas dos antigos (inseguros) mecanismos, vez que eles permanecem úteis em um ambiente de rede moderno, como para a apuração de erros e execução de programas exóticos.

## Rsh, rlogin e telnet

Os comandos **rlogin** e **rsh** para login e execução remota de comandos foram herdados do UNIX. Embora raramente sejam usados, porquanto são descaradamente inseguros, eles ainda são fornecidos por quase todas as versões do Linux por motivos de compatibilidade com os programas UNIX.

O Telnet, por outro lado, ainda é muito usado, em geral por administradores de sistemas e redes. Ele é uma das ferramentas mais poderosas para acesso remoto de arquivos e administração remota, permitindo conexões a partir de qualquer ponto da internet. Combinado com o servidor X, aplicativos gráficos podem ser exibidos localmente. Inexiste diferença entre trabalhar na máquina local e acessar uma remota.

Uma vez que toda a conexão é aberta, permitir o uso do **telnet** é assumir consideráveis riscos. Para a execução remota de programas o Secure SHell ou **ssh** é recomendado. Discutiremos este método mais adiante nesta seção.

Não obstante, o **telnet** ainda é muito usado. Abaixo estão alguns exemplos em que um servidor de emails e outro web são testados:

Verificando se o servidor de emails está funcionando:

```
[jimmy@blob ~] telnet mailserver 25
Trying 192.168.42.1...
Connected to mailserver.
Escape character is '^]'.
220 m1.some.net ESMTP Sendmail 8.11.6/8.11.6; 200302281626
ehlo some.net
250-m1.some.net Hello blob.some.net [10.0.0.1], pleased to meet you
250-ENHANCEDSTATUSCODES
250-8BITMIME
250-SIZE
250-DSN
250-ONEX
250-ETRN
250-XUSR
250 HELP
mail from: jimmy@some.net
250 2.1.0 jimmy@some.net... Sender ok
rcpt to: davy@some.net
250 2.1.5 davy@some.net... Recipient ok
data
354 Enter mail, end with "." on a line by itself
test
.
250 2.0.0 g2MA1R619237 Message accepted for delivery
quit
221 2.0.0 m1.some.net closing connection
Connection closed by foreign host.
```

Verificando se o servidor web responde a pedidos simples:

```
[jimmy@blob ~] telnet www.some.net 80
Trying 64.39.151.23...
Connected to www.some.net.
Escape character is '^]'.
HEAD / ;HTTP/1.1

HTTP/1.1 200 OK
Date: Fri, 22 Mar 2002 10:05:14 GMT
Server: Apache/1.3.22 (UNIX) (Red-Hat/Linux)
 mod_ssl/2.8.5 OpenSSL/0.9.6
 DAV/1.0.2 PHP/4.0.6 mod_perl/1.24_01
Last-Modified: Fri, 04 Jan 2002 08:21:00 GMT
ETag: "70061-68-3c3565ec"
Accept-Ranges: bytes
Content-Length: 104
Connection: close
Content-Type: text/html
```

Connection closed by foreign host.

```
[jimmy@blob ~]
```

Isto é perfeitamente seguro, porque você jamais precisa informar um nome de usuário e/ou senha para levar os dados para onde quer, assim, ninguém pode retirar aquelas informações importantes do cabo.

# O Sistema de Janelas X

## Recursos do X

Como explicado no capítulo 7 veja [the section called "Configuração do servidor X"](#)), o sistema X Window vem com um servidor X que fornece gráficos aos clientes que precisam exibi-los.

É importante perceber a distinção entre o servidor X e o(s) aplicativo(s) cliente(s) do X. O servidor X controla diretamente os gráficos e é responsável por todas as entradas e saídas do teclado, mouse e monitor. O cliente X, por sua vez, não acessa diretamente os dispositivos de entrada e saída. É o cliente X que realiza o trabalho de verdade, como a computação de valores, execução de aplicativos e assim por diante. O servidor X apenas abre janelas para manipular entrada e saída para o cliente especificado.

Em uma operação normal (no modo gráfico), toda estação de trabalho Linux é um servidor X de si mesma, mesmo quando executa apenas aplicativos clientes. Todos os aplicativos que você está executando (por exemplo, o Gimp, uma janela de terminal, seu navegador, o suíte de escritório, a ferramenta de execução de CDs e assim por diante) são clientes do seu servidor X. Servidor e cliente são executados pela mesma máquina neste caso.

A natureza cliente/servidor do sistema X torna-o um ambiente ideal para a execução remota de aplicativos e programas. Uma vez que o processo está em execução em uma máquina remota, muito pouco da capacidade do CPU é consumida na máquina local. Tais máquinas, que atuam como servidores puros do X, são denominadas terminais X e já foram um dia muito populares. Mais informações podem ser encontradas no [Remote X applications mini-HOWTO](#).

## Telnet e X

Se você quiser usar o **telnet** para exibir aplicativos gráficos em execução em uma máquina remota, primeiro é necessário fornecer a ela acesso à sua tela (seu servidor X!), usando o comando **xhost**, informando no seu terminal um comando similar ao que abaixo se lê:

```
davy:~> xhost +remote.machine.com
```

Após isto, conecte-se à máquina remota e determine que ela exiba gráficos na máquina local, configurando a variável de ambiente DISPLAY:

```
[davy@remote ~] export DISPLAY="local.host.com:0.0"
```

Após completar este passo, quaisquer aplicativos iniciados nesta janela de terminal serão exibidos na sua máquina local, usando recursos remotos para a computação, mas recursos locais para gráficos (seu servidor X) para a exibição dos aplicativos.

Este procedimento pressupõe que você possui algum tipo de servidor X (XFree86, X.org, Exceed, Cygwin) pré-instalado na máquina em que você pretende exibir as imagens. A arquitetura e sistema operacional da máquina cliente é de pouca relevância desde que elas permitam que o usuário execute um servidor X.

Perceba que exibir uma janela de terminal a partir de uma máquina remota também é considerado uma exibição de imagem.

## O suíte SSH

### Introdução

A maior parte dos sistemas UNIX e Linux roda hoje o Secure SHell como forma de se livrar dos riscos à segurança inerentes ao **telnet**. A maior parte dos sistemas Linux usam uma versão do **OpenSSH**, uma implementação open source do protocolo SSH, que fornece uma comunicação segura e criptografada entre máquinas não confiáveis em uma rede suspeita. Na configuração padrão, as conexões do X são automaticamente redirecionadas, mas portas TCP/IP arbitrárias também podem ser redirecionadas usando um canal seguro.

O cliente **ssh** conecta-se e efetua o login na máquina especificada. O usuário precisa identificar-se à máquina remota na forma explicitada no arquivo `sshd_config`, que via de regra pode ser encontrado no `/etc/ssh`. O arquivo de configuração é auto-explicativo e por padrão habilita os recursos mais comuns. Caso você precise de ajuda, poderá encontrá-la nas páginas man do **sshd**.

Quando a identidade do usuário tiver sido aceita pelo servidor, ele poderá executar o comando informado ou efetuar o login na máquina e fornecer ao usuário um shell normal naquela máquina. Todas as comunicações com a máquina remota ou com o shell serão automaticamente criptografados.

A sessão expira quando o comando ou o shell na máquina remota sair e todas as conexões X11 e TCP/IP tiverem sido fechadas.

Quando se conectar com uma máquina pela primeira vez, usando quaisquer dos programas que estão inclusos na coleção SSH, é preciso estabelecer a autenticidade daquela máquina e confirmar sua vontade de conectar-se:

```
lenny ~> ssh blob
The authenticity of host 'blob (10.0.0.1)' can't be established.
RSA fingerprint is 18:30:50:46:ac:98:3c:93:1a:56:35:09:8d:97:e3:1d.
```

```
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'blob,192.168.30.2' (RSA) to the list of
known hosts.
Last login: Sat Dec 28 13:29:19 2002 from octarine
This space for rent.
```

```
lenny is in ~
```

É importante que você digite “yes”, em três caracteres, não apenas “y”. Isto edita o seu arquivo `~/.ssh/known_hosts`, veja o [the section called “Autenticação com o servidor”](#).

Se você deseja apenas verificar alguma coisa em uma máquina remota e, então, ter o seu prompt de volta na máquina local, você pode informar os comandos que deseja executar remotamente como argumentos para o **ssh**:

```
lenny ~-> ssh blob who
jenny@blob's password:
root      tty2      Jul 24 07:19
lena      tty3      Jul 23 22:24
lena      0:       Jul 25 22:03

lenny ~-> uname -n
magrat.example.com
```

## Redirecionamento de X11 e TCP

Se a variável `X11Forwarding` estiver configurada como `yes` em uma máquina alvo e o usuário estiver usando aplicativos X, a variável de ambiente `DISPLAY` é configurada, a conexão com o X11 automaticamente redirecionada para o lado remoto de forma que todos os programas X11 iniciados a partir do shell passarão por um canal seguro e a conexão com o servidor X real será feita a partir da máquina local. O usuário não deve configurar manualmente a variável `DISPLAY`. O redirecionamento de conexões X11 pode ser feito na linha de comando ou no arquivo **sshd**.

O valor de `DISPLAY` configurado por **ssh** irá apontar para o servidor, mas com um número de exibição maior que zero. Isto é normal e acontece porque o **ssh** cria um servidor *proxy* do X na máquina servida (a que executa os aplicativos X) para redirecionar as conexões pelo canal criptografado.

Tudo isto é feito automaticamente, então quando você digitar o nome de um aplicativo gráfico, ele será exibido na sua máquina local e não na remota. Usamos o **xclock** no exemplo, vez que ele é um pequeno programa em geral pré-instalado e ideal para testes:

### Figure 10.3. SSH X11 forwarding



O SSH também irá automaticamente configurar os dados do `Xauthority` na máquina servida. Para este fim, ela irá gerar um cookie de autorização aleatório, armazená-lo no `Xauthority` do servidor e verificar se todas as conexões redirecionadas carregam este cookie e substituem-no pelo cookie real quando a conexão é aberta. O cookie de autenticação real nunca é enviado para a máquina servida (em verdade, não são enviados quaisquer cookies).

Redirecionando conexões TCP/IP arbitrárias através de um canal seguro pode ser informado tanto na linha de comando como em um arquivo de configuração.

## O servidor X

Este procedimento pressupõe que você possui um servidor X em execução no cliente em que você deseja exibir o aplicativo da máquina remota. Desde que ele possa executar um servidor X, como o `Cygwin` (que implementa o servidor `X.org` em clientes `MS Windows` e outros) ou `Exceed`, o cliente pode usar um sistema operacional ou ser de uma arquitetura diferente e ainda assim ser capaz de estabelecer uma conexão remota com quaisquer máquinas `Linux` ou `UNIX`.

## Autenticação com o servidor

O sistema cliente/servidos **ssh** automaticamente mantém e verifica uma base de dados contendo identificações para todas as máquinas para as quais já foi usado. As chaves para as máquinas são armazenadas no `$HOME/.ssh/known_hosts` na pasta home do usuário. Além disto, o arquivo `/etc/ssh/ssh_known_hosts` é automaticamente consultado a respeito de máquinas conhecidas. Quaisquer novas máquinas são automaticamente adicionadas ao arquivo do usuário. Se a identificação de uma máquina for alterada, o **ssh** emite um aviso e desabilita a autenticação da senha, para prevenir que um cavalo de Tróia obtenha a senha do usuário. Um propósito secundário é evitar ataques de intermediários que, de outra forma, poderiam ser usados para contornar a criptografia. Em ambientes em que a segurança é primordial, o **sshd** pode até mesmo ser configurado para proibir que se efetue o login em máquinas para as quais as chaves tenham sido mudadas ou sejam desconhecidas.

## Cópia remota segura

O suíte SSH fornece o comando **scp** como uma alternativa segura ao comando **rmp**, que era muito

popular quando existia apenas o **rsh**. O **scp** usa o **ssh** para a transferência de dados, usa a mesma autenticação e fornece a mesma segurança que o **ssh**. Ao contrário do **rcp**, o **scp** pergunta por senhas ou frases de segurança, caso sejam necessárias para a autenticação:

```
lenny /var/tmp> scp Schedule.sdc.gz blob:/var/tmp/
lenny@blob's password:
Schedule.sdc.gz 100% |*****| 100 KB 00:00
```

```
lenny /var/tmp>
```

Qualquer nome de arquivo pode conter uma especificação de máquina ou usuário para indicar que um arquivo deve ser copiado para/de uma máquina. São permitidas cópias entre duas máquinas remotas. Veja as páginas Info para saber mais.

Se você prefere usar uma interface similar ao FTP, use o **sftp**:

```
lenny /var/tmp> sftp blob
Connecting to blob...
lenny@blob's password:

sftp> cd /var/tmp

sftp> get Sch*
Fetching /var/tmp/Schedule.sdc.gz to Schedule.sdc.gz
```

```
sftp> bye
```

```
lenny /var/tmp>
```

## Cópia segura ou interfaces gráficas para FTP

Ainda não se sente confortável com a linha de comando? Tente usar as ferramentas do Konqueror para cópia remota segura ou instale o Putty.

## Chaves de autenticação

O comando **ssh-keygen** gera, gerencia e converte chaves de autenticação para o **ssh**. Ele pode criar chaves RSA para uso pelo protocolo SSH versão 1 e chaves RSA ou DSA para uso com a versão 2 do protocolo.

Normalmente, cada usuário que deseja usar o SSH com autenticação RSA ou DSA executa isto uma vez para criar a chave de autenticação no `$HOME/.ssh/identity`, `id_dsa` ou `id_rsa`. Adicionalmente, o administrador do sistema pode usar isto para criar chaves para o sistema.

Normalmente este programa gera a chave e pergunta por um arquivo no qual ele possa alocar a chave privada. A chave pública é armazenada em um arquivo com o mesmo nome, mas com o sufixo `.pub`. O programa também pergunta por uma frase de segurança. Ela poderá ser vazia para indicar que inexistente (chaves do sistema precisam ter uma frase de segurança vazia), ou pode ser uma seqüência de extensão arbitrária.

Não existe modo de recuperar uma frase de segurança perdida. Caso a frase seja perdida ou esquecida, uma nova chave precisará ser criada e copiada para as chaves públicas correspondentes.

Estudaremos chaves SSH nos exercícios. Todas as informações podem ser encontradas nas páginas man ou Info.

## VNC

O VNC ou Virtual Network Computing é na verdade uma exibição remota system que permite a visualização de um ambiente de trabalho não apenas na máquina local em que está sendo executada, mas também a partir de qualquer ponto da internet, por uma larga gama de máquinas e arquiteturas incluindo aqui o MS Windows e diversas distribuições do Linux. Você pode, por exemplo, executar o MS Word em um Windows NT e exibir a saída no seu computador Linux. O VNC fornece servidores e clientes, de forma que o contrário também é verdadeiro, sendo possível exibir programas Linux em clientes Windows. O VNC é provavelmente o meio mais simples de realizar conexões X em um computador. Os recursos a seguir distinguem o VNC de um servidor normal do X ou implementações comerciais:

- Nenhum conteúdo é armazenado pelo lado que visualiza: você pode abandonar sua mesa e continuar de onde parou em outra máquina. Quando você está executando um servidor X e o computador trava ou é reiniciado, todos os aplicativos remotos que estavam em execução são encerrados. Com o VNC eles continuam em execução.
- É pequeno e simples, não carece de instalação, se preciso for pode ser executado a partir de um disquete.
- É independente da arquitetura, com o cliente Java ele pode ser executado em praticamente qualquer coisa que suporte o X.
- Compartilhável: um ambiente pode ser exibido para múltiplos visualizadores.
- Livre.

Mais informações podem ser encontradas nas páginas man do VNC (**man vncviewer**) ou na [página do VNC](#).

## O protocolo rdesktop

Para um gerenciamento simples de máquinas MS Windows, distribuições recentes do Linux suportam o Remote Desktop Protocol (RDP), que é implementado no cliente **rdesktop**. O protocolo é usado em um grande número de produtos Microsoft, incluindo o Windows NT Terminal Server, Windows 2000 Server, Windows XP e Windows 2003 Server.

Surpreenda os seus amigos (ou gerente) com o modo tela cheia, diversos tipos de disposição de teclado e modo de aplicativo único, exatamente como o original. O manual **man rdesktop** fornece mais informações. A página do projeto é <http://www.rdesktop.org/>.

## Cygwin

O [Cygwin](#) fornece substancial funcionalidade UNIX em sistemas MS Windows. Além de fornecer ferramentas da linha de comando UNIX e aplicativos gráficos, ele também pode ser usado para exibir um ambiente Linux em uma máquina MS Windows, usando um X remoto. De um shell Cygwin Bash, informe o comando

```
/usr/X11R6/bin/XWin.exe -query nome_da_sua_máquina_linux_ou_IP
```

A conexão é negada por padrão. Você precisa mudar a configuração do X Display Manager (XDM) e provavelmente a configuração do X Font Server (XFS) para permitir este tipo de conexão, em que você recebe uma tela de login na máquina remota. Dependendo do seu gerente de desktop (Gnome, KDE, ou outro), você pode ter que mudar algumas configurações lá também.

Se você não precisa exibir a sua área de trabalho inteira, você pode usar o SSH no Cygwin, exatamente como explicado no [the section called "O suíte SSH"](#). Sem todo o trabalho de edição dos seus arquivos de configuração.

## Segurança

### Introdução

A partir do momento em que um computador se conecta a uma rede, todo tipo de abuso torna-se possível, qualquer que seja o sistema. Pilhas e pilhas de papel já foram escritas sobre este assunto e iríamos muito longe caso adentrássemos o tema da segurança em seus detalhes. Existem, contudo, um par de atitudes lógicas que mesmo um novato pode tomar para obter um sistema muito seguro, vez que a maior parte das intromissões acontecem por conta de usuários descuidados ou ignorantes.

Talvez você esteja se indagando se isto se aplica a você, usando o seu computador em casa ou trabalhando em um computador de mesa em ambiente bem protegido de um escritório. As perguntas que você deveria estar se fazendo, contudo, seguem a linha das seguintes:

- Você deseja estar no controle do seu próprio sistema?
- Você deseja participar (sem saber) de atividades criminosas?
- Você deseja que o seu computador seja usado por terceiros?
- Você deseja correr o risco de perder sua conexão com a internet?
- Você deseja reconstruir seu sistema toda vez que ele for invadido?
- Você deseja arriscar perder dados pessoais?

Presumindo que não, listaremos alguns passos que você pode tomar para tornar segura a sua máquina. Mais informações podem ser encontradas no [Linux Security HOWTO](#).

## Serviços

O objetivo é executar tanto menos serviços quanto for possível. Se o número de portas que estiverem abertas ao mundo exterior for mantido no mínimo, torna-se muito mais simples visualizar o todo. Se os serviços não puderem ser desligados para a rede local, tente ao menos desabilitá-los para conexões exteriores.

Pode-se adotar o princípio de que se você não reconhece um serviço em especial, é bem provável que também não precise dele. Tenha também em mente que alguns serviços não são destinados para uso através da internet. Não confie naquilo que *deveria* estar em execução, verifique quais serviços estão ouvindo quais portas TCP, usando o comando **netstat**:

```
[elly@mars ~] netstat -l | grep tcp
tcp    0    0  *:32769          *:*    LISTEN
tcp    0    0  *:32771          *:*    LISTEN
tcp    0    0  *:printer        *:*    LISTEN
tcp    0    0  *:kerberos_master *:*    LISTEN
tcp    0    0  *:sunrpc         *:*    LISTEN
```

```

tcp    0    0 *:6001                *:*    LISTEN
tcp    0    0 *:785                 *:*    LISTEN
tcp    0    0 localhost.localdom:smtp *:*    LISTEN
tcp    0    0 *:ftp                 *:*    LISTEN
tcp    0    0 *:ssh                 *:*    LISTEN
tcp    0    0 :::1:x11-ssh-offset  *:*    LISTEN

```

Coisas a se evitar:

- **exec**, **rlogin** e **rsh**, e **telnet** por prevenção.
- X11 em servidores
- Não deve haver lp se não houver impressora.
- Se não houver máquinas MS Windows na rede, o Samba é desnecessário.
- Não permita conexões FTP a menor que um servidor FTP seja necessário.
- Não permita NFS e NIS pela internet, desabilite todos os serviços relacionados em uma instalação autônoma.
- Não execute um MTA se você não estiver em um servidor de emails.
- ...

Encerre a execução de serviços usando o comando **chkconfig**, os initscripts ou editando os arquivos de configuração do **(x)inetd**.

## Atualize o sistema com regularidade

A habilidade do Linux de se adaptar rapidamente em um ambiente sempre em mudança é o que lhe traz sucesso. Mas também é o que cria possibilidade de que atualizações de segurança tenham sido disponibilizadas enquanto você está instalando uma versão novíssima do sistema. Assim, a primeira coisa a ser feita (e isto em qualquer sistema que você possa pensar) após a instalação é a atualização do sistema. Após isto, atualize *todos* os pacotes que você usa com frequência.

Algumas atualizações podem precisar de novos arquivos de configuração e arquivos antigos podem ser substituídos. Verifique a documentação e certifique-se de que tudo opera normalmente após a atualização.

A maior parte das distribuições do Linux fornece uma lista de emails para em que são anunciadas as atualizações de segurança, bem como ferramentas para realizar atualizações ao sistema. Problemas relacionados ao Linux são anunciados no [Linuxsecurity.com](http://Linuxsecurity.com) dentre outros.

A atualização é um processo ininterrupto e, portanto, deve ser um hábito diário.

## Firewalls e política de acesso

### O que é um firewall?

Na seção anterior, já mencionamos o uso de firewall no Linux. Embora a administração de firewalls seja uma tarefa para administradores de redes, convém que você saiba algumas coisas sobre eles.

*Firewall* é um termo genérico que pode significar qualquer coisa que atue como uma barreira de proteção entre nós e o mundo exterior, a internet. Um firewall pode ser um sistema inteiro ou aplicativo específico que proporcione esta funcionalidade. Pode também ser a combinação de componentes, incluindo aqui várias combinações de hardwares e softwares. Os firewalls são construídos a partir de “regras” que são usadas para definir o que pode entrar e/ou sair de um dado sistema ou rede.

Após desabilitar serviços desnecessários, podemos, então, restringir os serviços aceitos, de forma a permitir somente o mínimo de conexões necessárias. Um bom exemplo é trabalhar em casa: apenas a conexão específica entre o seu escritório e sua casa deve ser permitida, todas as demais máquinas na internet devem ser bloqueadas.

### Filtros de pacotes

A primeira linha de defesa é o *filtro de pacotes*, que consegue olhar o interior dos pacotes IP e tomar decisões baseadas em seus conteúdos. O mais comum é o pacote Netfilter que fornece o comando **iptables**, um filtro de última geração para o Linux.

Um dos aprimoramentos dignos de nota nos kernels mais recentes é o recurso de *inspeção de estado*, que não apenas informa o conteúdo de um pacote, mas também detecta se um pacote pertence ou está relacionado a uma conexão nova ou pré-existente.

O Shoreline Firewall ou Shorewall é uma interface para as funções de firewall padrão no Linux.

Mais informações podem ser encontradas no [the Netfilter/iptables project page](http://the Netfilter/iptables project page).

### TCP wrappers

O embrulho TCP fornece os mesmos resultados que os filtro de pacotes, mas funciona de modo distinto. O embrulhador aceita a tentativa de conexão, para então examinar os arquivos de configuração e decidir se aceita ou não o pedido de conexão. Ele controla as conexões ao nível dos aplicativos, ao invés de fazê-lo ao nível da rede.

Embrulhadores TCP são tipicamente usado em conjunto com o **xinetd** para fornecer um controle de acesso baseado em nomes de máquinas e endereços de IP. Além disto, estas ferramentas incluem recursos de gerenciamento de login e utilização de fácil configuração.

As vantagens dos embrulhadores de TCP são que o cliente em conexão não está ciente a respeito do uso dos embrulhadores, bem como o fato de que a sua operação é separada daquela dos aplicativos que protegem.

O acesso baseado em máquinas é controlado nos arquivos `hosts.allow` e `hosts.deny`. Mais informações podem ser encontradas na documentação dos embrulhadores TCP no `/usr/share/doc/tcp_wrappers[-<version>/]` ou `/usr/share/doc/tcp`, bem como nas páginas `man` para os arquivos de controle de acesso baseado em máquinas, as quais contém exemplos.

## Proxies

Os proxies podem executar diversos trabalhos, dentre os quais nem todos são relacionados à segurança. Mas o fato de que eles são intermediários torna os proxies um bom lugar para impor políticas de controle de acesso, limitar conexões diretas através de um firewall e controlar o modo como a rede por detrás do proxy enxerga a internet.

Em geral, quando combinados com filtros de pacotes, mas às vezes também quando sozinhos, os proxies fornecem um nível extra de controle. Mais informações podem ser encontradas no [Firewall HOWTO](#) ou na página do Squid.

## Acesso a aplicativos individuais

Alguns servidores podem ter seus recursos próprios para o controle de acesso. Exemplos comuns incluem o Samba, o X, o Bind, o Apache e o CUPS. Para cada serviço que você queira ofertar, verifique quais os arquivos de configuração são aplicáveis.

## Arquivos de log

Dentre outras coisas, o modo do UNIX de armazenar todos os tipos de atividades em todos os tipos de arquivos confirma que “ele está fazendo alguma coisa.”. É claro arquivos de deveriam ser verificados com frequência, manual ou automaticamente. Firewalls e outros meios de controle de acesso tendem a criar enormes quantidades de arquivos de log, então o truque é tentar armazenar apenas atividades anormais.

## Detecção de invasões

Sistemas de Detecção de Invasões são desenhados capturar aquilo que pode ter passado pelo firewall. Eles tanto podem ser desenhados para perceber uma tentativa de invasão em curso ou para detectar uma invasão que já aconteceu. Neste caso, já é tarde demais para prevenir quaisquer danos, mas ao menos descobrimos cedo a existência do problema. Existem dois tipos básicos de SDI: aqueles que protegem redes e aqueles que protegem máquinas individuais.

Para SDIs em máquinas, isto é feito por meio de utilitários que monitoram mudanças no sistema de arquivos. Arquivos do sistema que tenham sido de alguma forma alterados são um forte indício de que existe alguma coisa errada. Qualquer um que invada o sistema e consiga acesso à conta root provavelmente fará alterações em algum ponto do sistema. Isto é, em geral, a primeira coisa a ser feita, ou para facilitar o seu retorno ao sistema ou para que possa lançar um ataque contra terceiro, casos em que ele de qualquer maneira precisará adicionar ou alterar arquivos do sistema. Alguns sistemas vêm com o sistema de monitoramento **tripwire**, que está documentado na página [Tripwire Open Source Project](#).

A detecção de invasões em redes é feita por um sistema que monitora todo o tráfego que passa pelo firewall (não por scanners de portas, que notifica portas usáveis). Um exemplo open source é o [Snort](#). O [whitehats.com](#) possui uma base de dados para detecção de invasões, [arachNIDS](#).

## Mais dicas

Algumas coisas que você deve ter em mente:

- Não permita logins do root. Os desenvolvedores do UNIX inventaram o **su** há mais de duas décadas para reforçar a segurança do sistema.
- Acesso direto à conta root é sempre perigoso e passível de erros humanos, seja pela permissão de logins com o root ou por meio do comando **su** -. Ao invés de usar o **su**, o ideal é usar o **sudo**, para executar somente o comando que carece de permissões especiais e retornar ao ambiente do usuário normal após a sua execução.
- Leve as senhas à sério. Use sombras de senhas. Mude suas senhas com regularidade.
- Tente sempre usar SSH ou SSL. Evite o **telnet**, o FTP, clientes de emails e outros programas que enviam senhas na rede sem criptografá-las. A segurança não é apenas do seu computador, mas



também de suas senhas.

- Estabeleça um limite de recursos usando o **quota** e/ou o **ulimit**.
- As mensagens para o root devem ser enviadas para, ou ao menos lidas por, uma pessoa.
- O [instituto SANS](#) fornece outras dicas e truques, organizados por distribuição, incluindo um serviços de lista de emails.
- Verifique a origem de novos softwares, obtenha-os apenas de lugares/páginas confiáveis. Verifique os novos pacotes antes de instalá-los.
- Quando estiver usando uma conexão temporária com a internet, feche-a assim que terminar de usá-la.
- Execute serviços privados em portas estranhas ao invés daquelas em que são esperados por invasores.
- Conheça o seu sistema. Após um tempo, você pode quase sentir quando algo está acontecendo.

## Eu fui invadido?

Como você pode saber? Esta é uma lista de eventos suspeitos:

- Misteriosas portas abertas, processos estranhos.
- Utilitários do sistema (comandos usuais) agindo de modo estranho.
- Problemas para efetuar o login.
- Consumo inexplicável de banda.
- Arquivos de log corrompidos ou ausentes, daemon syslog agindo de forma estranha.
- Interfaces em modos não-usuais.
- Arquivos de configuração alterados de maneira não esperada.
- Entradas estranhas no histórico do shell.
- Arquivos temporários não identificados.

## Recuperando-se de uma invasão

Antes de mais nada, fique calmo. Então, tome os seguintes passos, em ordem:

- Desconecte a máquina da rede.
- Tente descobrir tudo o que puder sobre como seu sistema de segurança foi quebrado.
- Faça o backup de dados pessoais importantes. Se possível, compare estes dados com aqueles de backups anteriores, para certificar-se de que não houve corrupção de dados.
- Reinstale o sistema
- Use novas senhas.
- Recupere os dados a partir de backups do sistema e dos seus arquivos.
- Atualize o sistema.
- Re-examine o sistema: bloqueie serviços desnecessários, verifique as regras do seu firewall e outras políticas de acesso.
- Reconecte o sistema.

## Resumo

O Linux anda de mãos dadas com as redes. O kernel do Linux possui suporte para todos os protocolos de rede comuns e também aos pouco usuais. As ferramentas de rede padrão do UNIX são fornecidas por todas as distribuições. Além delas, a maior parte das distribuições também dispõe de ferramentas simples para a instalação e gerenciamento de redes.

O Linux é famoso por ser uma plataforma estável para a execução de diversos serviços da Internet. Existe uma quantidade sem fim de softwares para a Internet. Assim como o UNIX, o Linux também pode ser usado e administrado de um local remoto, por meio de diversas ferramentas para a execução de programas à distância.

Passamos rapidamente pelo tema da segurança. O Linux é um sistema de firewall ideal, leve e barato, mas pode ser usado em diversas outras funções de redes como roteadores e servidores proxy.

A atualização freqüente do sistema e o bom senso são os principais meios de se aumentar a segurança da rede.

Segue um panorama dos comandos relacionados a redes:

**Table 10.2. Novos comandos no capítulo 10: Redes**

Comando	Significado
<b>ftp</b>	Transfere arquivos para outra máquina (inseguro).
<b>host</b>	Obtém informações sobre outras máquinas.
<b>ifconfig</b>	Exibe informações sobre endereços de IP.
<b>ip</b>	Exibe informações sobre endereços de IP.
<b>netstat</b>	Exibe informações sobre roteamento e estatísticas da rede.
<b>ping</b>	Envia pedidos de resposta para outras máquinas.
<b>rdesktop</b>	Exibe um MS Windows no seu sistema Linux.
<b>route</b>	Exibe informações de roteamento.
<b>scp</b>	Faz uma cópia segura de e para outras máquinas.
<b>sftp</b>	FTP seguro de e para outras máquinas.
<b>ssh</b>	Cria uma conexão segura com outra máquina.
<b>ssh-keygen</b>	Generate authentication keys for Secure SHell.
<b>telnet</b>	Cria uma conexão insegura com outras máquinas.
<b>tracpath/traceroute</b>	Exibe a rota que os pacotes seguem para outras máquinas.
<b>whois</b>	Obtém informações sobre um nome de domínio.
<b>xclock</b>	Aplicativo relógio do X, útil para testar a exibição remota.
<b>xhost</b>	Controle de acesso do sistema X

## Exercícios

### Redes em geral

- Exibe informações de redes para a sua estação de trabalho: endereço IP, rotas, nomes de servidores.
- Suponha que não há DNS disponível. O que você faria para contatar a máquina do seu vizinho sem ter que digitar o endereço de IP o tempo todo?
- Como você armazenaria permanentemente informações de proxy para um navegador no modo texto como **links**?
- Quais servidores de nomes cuidam do domínio redhat.com?
- Envie um email para a sua conta local. Tente dois modos diferentes de envio e leia-o. Como você pode verificar que ele chegou?
- A sua máquina aceita conexões FTP anônimas? Como você usa o programa **ncftp** para realizar a autenticação com o seu nome de usuário e senha?
- A sua máquina roda um servidor web? Se não, faça-a rodar. Verifique os arquivos de log!

### Conexões remotas

- A partir de sua estação de trabalho local, exiba um aplicativo gráfico, como o **xclock** na tela do seu vizinho. As contas necessárias precisaram estar configuradas, use uma conexão segura.
- Configure chaves SSH para que você possa se conectar à máquina do seu vizinho sem ter que informar uma senha.
- Faça uma cópia de backup da sua pasta home no `/var/tmp` no “servidor de backup” do seu vizinho com o **scp**. Empacote e comprima antes de começar a transferir os dados! Conecte-se à máquina remota usando o **ssh**, descompacte o backup e coloque um dos arquivos na máquina original usando o **sftp**.

### Segurança

- Faça uma lista de portas aberta (ouvindo) na sua máquina.
- Suponha que você deseje rodar um servidor web, quais serviços você desativaria? Como faria isto?
- Instale todos as atualizações disponíveis.
- Como você pode ver quem se conectou ao seu sistema?

- Crie uma rotina que o lembre de mudar a sua senha todos os meses e de preferência a senha *root* também.

## Chapter 11. Som e vídeo

### Table of Contents

#### [Áudio Básico](#)

- [Instalação](#)
- [Driver e arquitetura](#)

#### [Execução de som e vídeo](#)

- [Execução e cópia de CDs](#)
- [Tocando arquivos de música](#)
- [Gravação](#)

#### [Execução de vídeo, fluxos e televisão](#)

#### [Telefonia de Internet](#)

- [O que é?](#)
- [O que você vai precisar?](#)

#### [Resumo](#)

#### [Exercícios](#)

### Abstract

Este capítulo aborda as seguintes tarefas (sucintamente, já que o campo de som e vídeo é muito amplo):

- Configuração do cartão de som
- Execução e cópia de CDs,
- Tocar arquivos de música
- Controle de volume
- Vídeo e televisão
- Gravação de som

## Áudio Básico

### Instalação

Provavelmente o seu sistema já instalou automaticamente os drivers de audio e a configuração foi feita quando da instalação. Da mesma forma, caso você um dia precise trocar o seu hardware de áudio, a maior parte dos sistemas fornece ferramentas que permitem facilmente configurar o equipamento. A maior parte dos cartões plug-and-play deve ser reconhecida automaticamente. Se você consegue ouvir as amostras que são tocadas quando da configuração, clique OK e a configuração será realizada automaticamente.

Se o seu cartão não for detectado automaticamente, pode ser que o sistema apresente uma lista de cartões de som ou de propriedades deles, a partir dos quais você deve realizar uma escolha. Após isto, você tem que fornecer a porta de E/S correta e as configurações de IRQ e DMA. Informações sobre estas configurações podem ser obtidas na documentação do seu cartão de som. Se você estiver usando um sistema em paralelo com o MS Windows, estas informações também podem ser obtidas no Painel de Controle dele.

### Se a detecção automática do cartão de som falhar

Se o seu cartão de som não for suportado por padrão, você terá que usar outras técnicas. Elas estão descritas no [Linux Sound HOWTO](#).

### Driver e arquitetura

Existem basicamente dois tipos de arquiteturas de som: o antigo Open Sound System ou OSS, que funciona em quaisquer sistemas UNIX e o mais recente Advanced Linux Sound Architecture ou ALSA, que tem melhor suporte para o Linux, como o seu nome indica. O ALSA possui mais recursos e permite um desenvolvimento mais rápido do driver. Focaremos no ALSA.

Atualmente, praticamente todos os chipsets de áudio têm suporte. Apenas algumas soluções profissionais e alguns cartões desenvolvidos por empresas que se recusam a documentar as especificações de seus produtos permanecem sem suporte. Um panorama dos dispositivos suportados pode ser encontrado no site do ALSA <http://www.alsa-project.org/alsa-doc/index.php?vendor=All#matrix>.

A configuração de sistemas instalados com o ALSA é feita pela ferramenta **alsacnf**. Além disso, as

distribuições normalmente fornecem suas próprias ferramentas para a configuração dos cartões de som; estas ferramentas podem até mesmo integrar o novo e o velho meio de manipular equipamentos de som.

## Execução de som e vídeo

### Execução e cópia de CDs

O pacote *cdp* vem na maior parte das distribuições e fornece o **cdp** ou **cdplay**, um tocador de CDs no modo texto. Os gerentes de desktop normalmente fornece uma ferramenta gráfica, como o **gnome-cd** no Gnome, que pode ser iniciado do menu.

Entenda a diferença entre um CD de áudio e um CD de dados. Não é necessário montar um CD de áudio no sistema de arquivos para conseguir ouvi-lo. Isto porque os dados em tais CDs não são dados do sistema de arquivos do Linux; eles são acessados e encaminhados ao canal de saída de áudio diretamente, por meio de um programa de execução de CDs. Se o seu CD é um CD de dados que contém arquivos *.mp3*, então será preciso primeiro montar o CD no sistema de arquivos e usar um dos programas abaixo para tocar as músicas. O processo de montagem do CD no sistema de arquivos está descrito no [the section called "Instalando pacotes extras a partir do CD de instalação"](#).

O **cdparanoia** do pacote homônimo lê áudio diretamente do CD, sem realizar conversões analógicas e escreve dados em um arquivo ou redireciona-os em diferentes formatos, dentre os quais o *.wav* é provavelmente o mais popular. Várias ferramentas realizam a conversão para outros formatos, tais como o *.mp3*, os quais vêm com a maior parte das distribuições e podem ser baixados como pacotes individuais. O projeto GNU fornece diversas ferramentas para a execução, decodificação e cópia de CDs; veja o [Free Software Directory, Audio section](#) para obter informações detalhadas.

A criação de CDs de áudio é facilitada pela ferramenta **kaudiocreator** do suíte KDE, dentre outras disponíveis. Ela vem com informações claras do Centro de Ajuda do KDE.

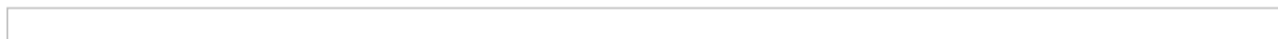
A gravação de CDs é abordada no [the section called "Criando uma cópia com o gravador de CDs"](#).

### Tocando arquivos de música

#### Arquivos mp3

O popular formato *.mp3* tem amplo suporte no Linux. A maior parte das distribuições inclui múltiplos programas que podem executar estes arquivos. Dentre muitos outros aplicativos, o XMMS, que é exibido na captura de tela abaixo, é um dos mais difundidos, em parte porque ele tem a mesma aparência da ferramenta Windows.

**Figure 11.1. O tocador mp3 XMMS**



Muito popular para a execução de músicas são o Amarok, um aplicativo KDE que vem continuamente ganhando em popularidade e o MPlayer, que também executa filmes.

### Restrições

Algumas distribuições não permitem a execução de MP3 sem modificar suas configurações, em decorrência de restrições nas licenças das ferramentas MP3. É preciso instalar softwares extras para conseguir executar suas músicas.

No modo texto, pode-se usar o comando **mplayer**:

```
[tille@octarine ~]$ mplayer /opt/mp3/oriental/*.mp3
MPlayer 1.0pre7-RPM-3.4.2 (C) 2000-2005 MPlayer Team
CPU: Advanced Micro Devices Duron Spitfire (Family: 6, Stepping: 1)
Detected cache-line size is 64 bytes
CPUflags: MMX: 1 MMX2: 1 3DNow: 1 3DNow2: 1 SSE: 0 SSE2: 0
Playing /opt/oldopt/mp3/oriental/Mazika_Diana-Krozon_Super-Star_Ensani-Ma-
Bansak.mp3.
Cache fill: 1.17% (98304 bytes) Audio file detected.
Clip info:
Title: Ensani-Ma-Bansak.mp3
Artist: Diana-Krozon
Album: Super-Star
Year:
Comment:
Genre: Unknown
=====
Opening audio decoder: [mp3lib] MPEG layer-2, layer-3
mpg123: Can't rewind stream by 450 bits!
AUDIO: 44100 Hz, 2 ch, s16le, 160.0 kbit/11.34% (ratio: 20000->176400)
Selected audio codec: [mp3] afm:mp3lib (mp3lib MPEG layer-2, layer-3)
=====
Checking audio filter chain for 44100Hz/2ch/s16le -> 44100Hz/2ch/s16le...
AF_pre: 44100Hz/2ch/s16le
A0: [oss] 44100Hz 2ch s16le (2 bps)
```

Building audio filter chain for 44100Hz/2ch/s16le -> 44100Hz/2ch/s16le...  
Video: no video  
Starting playback...  
A: 227.8 (03:23:.1) 1.8% 12%

## Outros formatos

Nos afastaríamos muito do propósito deste documento se intentássemos discutir todos os formatos de áudio possíveis e como executá-los. Um panorama (incompleto) de outros softwares comuns para a execução e manipulação de som

- Ogg Vorbis: Formato livre de áudio: veja [o diretório de áudio do GNU](#) para conhecer as ferramentas - pode ser que elas já estejam inclusas em sua distribuição. Este formato foi desenvolvido porque o MP3 foi patenteado.
- Áudio e vídeo Real: **realplay** do [RealNetworks](#).
- SoX ou Sound eXchange: na verdade é um conversor de som, vem com o program**play**. Toca .wav, .ogg e muitos outros formatos, incluindo binários puros.
- Playmidi um tocados MIDI, veja o diretório GNU.
- Alsa Player: do projeto Advanced Linux Sound Architecture, veja o [AlsaPlayer web site](#).
- **mplayer**: toca praticamente qualquer coisa, incluindo arquivos mp3. Mais informações na [página do MPlayerHQ](#).
- **hxplay**: suporta RealAudio e ReaVideo, mp3, áudio mp4, Flash, wav e mais, veja [HelixDNA](#) (nem todos os componentes deste software são livres).
- **rhythmbox**: baseado no GStreamer, pode tocar tudo o que tiver suporte no GStreamer, que alega ser capaz de tocar qualquer coisa, veja as páginas do [Rhythmbox](#) e do [GStreamer](#).

Confira a documentação do sistema e as páginas man para saber sobre ferramentas específicas e obter explicações detalhadas sobre o seu uso.

## Eu não tenho estes aplicativos no meu sistema!

Muitas das ferramentas discutidas nas seções acima são softwares opcionais. É possível que estes aplicativos não estejam instalados no seu sistema por padrão, mas é possível encontrá-los nos pacotes adicionais da sua distribuição. Pode bem ser também que o aplicativo que você está procurando não faça parte da sua distribuição. Neste caso, é preciso baixá-los a partir de suas páginas na web.

## Controle de volume

O **aumix** e o **alsamixer** são duas ferramentas texto comuns para o ajuste dos controles de áudio. Use as setas do teclado para ajustar as configurações. O **alsamixer** possui uma interface gráfica quando iniciado a partir do menu do Gnome ou como **gnome-alsamixer** a partir da linha de comando. A ferramenta **kmix** comporta-se de maneira idêntica no KDE.

Independentemente da sua escolha para ouvir suas músicas ou outros sons, lembre-se que outras pessoas podem não estar interessadas em ouvi-los também. Tente ser cortês, principalmente no trabalho. Use um fone de ouvido de boa qualidade, ao invés daqueles pequenos que se encaixam no ouvido. Eles são mais para os seus ouvidos e não causam incômodos para os seus colegas.

## Gravação

Existem várias ferramentas que permitem que você grave voz e música. Para a gravação de voz, pode-se usar o **arecord** na linha de comando:

```
alexey@russia:~> arecord /var/tmp/myvoice.wav  
Recording WAVE '/var/tmp/myvoice.wav' : Unsigned 8 bit, Rate 8000 Hz, Mono  
Aborted by signal Interrupts...
```

“Interrompido” quer dizer que o aplicativo recebeu um **Ctrl+C**. Execute a amostra usando o comando **play**.

Este é um bom teste que você pode executar antes de testar aplicativos que necessitam de entrada de voz, como Voice over IP (VoIP). Tenha em mente que a entrada de microfone deve estar ativa. Se você não ouvir sua própria voz, verifique suas configurações de som. Com frequência o microfone está mudo ou em um volume muito baixo. Isto pode ser facilmente ajustado usando o **alsamixer** ou a interface gráfica específica de sua distribuição.

No KDE você pode iniciar o utilitário **krec**, o Gnome fornece o **gnome-sound-recorder**.

## Execução de vídeo, fluxos e televisão

Diversos tocadores encontram-se disponíveis:

- **xine**: um tocador de vídeo livre
- **ogle**: um tocador de DVD
- **okle**: a versão KDE do **ogle**
- **mplayer**: tocador de filmes para o Linux
- **totem**: toca ambos, arquivos de áudio e vídeo, CDs de áudio, VCDs e DVDs.
- **realplay**: do RealNetworks.
- **hxplay**: uma alternativa para o Real, veja o [HelixDNA](#).
- **kaffeine**: tocador para o KDE3.

Você provavelmente encontrará um destes no menu gráfico.

Tenha em mente que todos os codecs necessários para a exibição dos diferentes tipos de vídeo podem não estar instalados no seu sistema por padrão. Você pode economizar muito trabalho baixando o W32codecs e o libdvdcss.

O [LDP](#) lançou um documento que é muito apropriado para esta seção. Ele se intitula [DVD Playback HOWTO](#) e descreve as diferentes ferramentas disponíveis para a exibição de filmes em um sistema que possui drive de DVD. É um ótimo complemento ao [DVD HOWTO](#) que explica a instalação do drive.

Para assistir TV, pode-se escolher uma das seguintes ferramentas, dentre as muitas existentes para assistir e capturar exibições de TV, vídeo e outros fluxos:

- **tvtime**: ótimo programa com gerenciamento de estações, interação com teletexto, modo filme e [muito mais](#).
- **zapping**: visualizador de TV específico do Gnome
- **xawtv**: visualizador de TV do X11.

## Telefonia de Internet

### O que é?

Telefonia de internet, mais conhecida como Voice over IP (VoIP) ou telefonia digital, permite que sejam transmitidos dados de voz pela rede. A grande diferença é que a data trafega por uma rede genérica, a internet, ao contrário da telefonia convencional, que usa uma rede específica para a transmissão de dados de voz. As duas redes podem ser ligadas, entretanto, em circunstâncias especiais, mas por enquanto isto não é padrão. Em outras palavras: é bem possível que você não consiga ligar para pessoas que estão usando linhas telefônicas convencionais. Se for possível, o mais provável é que você tenha que pagar por algum tipo de subscrição.

Embora atualmente existam diversos aplicativos para baixar gratuitamente, tanto livres como proprietários, existem enormes defeitos na telefonia pela internet. A mais notável é que o sistema não é confiável, pode ficar lenta e apresentar muito ruído na conexão, de forma que não deve ser usada como um substituto para a telefonia convencional - pense em ligações de emergência. Embora alguns fornecedores tomem precauções, não há como garantir que você conseguirá chamar a pessoa que deseja.

A maior parte dos aplicativos não usa criptografia, então tenha consciência de que é fácil para alguém bisbilhotar suas conversas. Se segurança é uma de suas preocupações, leia a documentação que vem com o seu cliente VoIP. Além disso, se você estiver usando um firewall, ele deve ser configurado para permitir a chamada de conexões a partir de qualquer lugar, então o uso de VoIP também inclui assumir riscos no que atine à segurança de sites.

### O que você vai precisar?

#### Quanto ao servidor

Em primeiro lugar, é preciso ter um provedor que forneça este serviço. Este serviço pode integrar a telefonia tradicional e pode ou não ser gratuito. Dentre outros pode-se citar [SIPphone](#), [Vonage](#), [Lingo](#), [AOL TotalTalk](#) e muitos provedores locais que oferecem o chamado "serviço de telefonia completo". O serviço de telefonia pela internet é oferecido pelo [Skype](#), [SIP Broker](#), e pelo [Google](#) dentre outros.

Se você deseja configurar um servidor próprio, aconselhamos que consulte o [Asterisk](#).

#### Quanto ao cliente

No que atine ao cliente, os aplicativos que você pode usar dependem da sua configuração de rede. Se você possui uma conexão direta com a internet, não haverá problemas, desde que você saiba a qual servidor você pode se conectar, para o qual, via de regra, haverá também um nome de usuário e senha para o serviço.

Se você estiver atrás de um firewall que realiza o Network Address Translation (NAT), todavia, alguns serviços podem não funcionar, já que eles enxergarão somente o endereço de IP do seu firewall e não o da sua máquina, que muito provavelmente é roteado na internet. Por exemplo, quando você está usando a rede de uma empresa e o seu endereço de IP começa com 10., 192.168. ou outro prefixo não roteável. Isto depende do protocolo que é usado pelo aplicativo.

Outro fator impediante é a banda disponível: alguns aplicativos são otimizados para o baixo consumo de banda, enquanto outros podem exigir conexões de banda larga. Isto depende do codec que é usado pelo aplicativo.

Dentre os aplicativos mais comuns encontra-se o cliente Skype, que possui uma interface semelhante ao de mensageiros instantâneos, e o [X-Lite](#), a versão livre do XTen softphone, que se parece com um telefone celular. Todavia, embora estes programas estejam disponíveis gratuitamente na internet e sejam muito populares, eles não são livres no sentido de liberdade de expressão: eles usam protocolos proprietários e/ou são distribuídos apenas em pacotes binários, sem código-fonte.

Clientes VoIP livres e *abertos* são, por exemplo, [Gizmo](#), [Linphone](#), [GnomeMeeting](#) e o [KPhone](#).

## Hardware cliente

Embora o seu computador, especialmente os laptops, pode ter um microfone embutido, o resultado será muito melhor se você usar um headset. Se você tiver opção de escolher, opte por um headset USB, que funciona independentemente do hardware de áudio existente. Use o **alsamixer** para configurar a entrada e saída de som aos volumes desejados.

Aplicativos VoIP são definitivamente um mercado em expansão. Voluntários tentam documentar o estado atual no <http://www.voip-info.org/>.

## Resumo

A plataforma GNU/Linux é totalmente compatível com multimídia. Uma grande variedade de equipamentos, tais como cartões de som, tv, headsets, microfones, leitores de CD e DVD são suportados. A lista de aplicativos é quase infinita, motivo pelo qual nós precisamos encurtar a lista de novos comandos abaixo àqueles relativos ao áudio em geral.

**Table 11.1. Novos comandos no capítulo 11: Áudio**

Comando	Significado
<b>alsacnf</b>	Configura o sistema de som ALSA
<b>alsamixer</b>	Ajusta os níveis de saída do driver ALSA
<b>arecord</b>	Grava uma amostra de som.
<b>aumix</b>	Ferramenta de mistura de som
<b>cdp</b>	Toca um CD de áudio.
<b>cdparanoia</b>	Copia um CD de áudio.
<b>cdplay</b>	Toca um CD de áudio.
<b>gnome-alsamixer</b>	Gnome interface gráfica para o ALSA.
<b>gnome-cd</b>	Gnome interface gráfica para tocar CDs.
<b>gnome-sound-recorder</b>	Gnomeinterface gráfica para gravar arquivos de som.
<b>kaudiocreator</b>	Interface KDE para criar CDs de áudio.
<b>kmix</b>	Interface KDE para a configuração de som.
<b>krec</b>	Interface KDE para a gravação de amostras de som.
<b>mplayer</b>	Tocador multimídia.
<b>play</b>	Ferramenta da linha de comando para tocar amostras de som.

## Exercícios

1. A partir do menu do Gnome ou do KDE, abra o seu painel de configuração de som. Certifique-se de que as caixas de som ou fones de ouvido estão conectados ao seu sistema e encontre um volume que lhe seja confortável. Certifique-se, se o seu sistema for compatível com o ALSA, de que você está usando o painel apropriado.
2. Se você possui um microfone, tente gravar uma amostra de som de sua escolha. Certifique-se de que o volume da entrada de som não está muito alto, vez que isto causará tons muito agudos quando você se comunicar com outros. Na linha de comando, você pode tentar usar o **arecord** e o **aplay** para gravar e ouvir sons.
3. Localize os seus arquivos de som no sistema e tente tocá-los.
4. Insira um CD de áudio e tente tocá-lo.
5. Encontre alguém para conversar e configure um programa VoIP. (Pode ser que você precise antes instalar um).

6. Você consegue ouvir rádio pela internet?
7. Se você tiver um leitor de DVD e um filme nessa mídia, tente tocá-la.

## Appendix A. Para aonde ir a partir daqui?

### Table of Contents

#### [Livros úteis](#)

- [Linux em geral](#)
- [Editores](#)
- [Shells](#)
- [X Window](#)
- [Redes](#)

#### [Sites úteis](#)

- [Informações gerais](#)
- [Referências por arquitetura](#)
- [Distribuições](#)
- [Software](#)

### Abstract

Este documento fornece um panorama de livros e sites úteis.

## Livros úteis

### Linux em geral

- "Linux in a Nutshell", Ellen Siever, Jessica P. Hackman, Stephen Spainhour, Stephen Figgins, O'Reilly UK, ISBN 0596000251
- "Running Linux", Matt Welsh, Matthias Kalle Dalheimer, Lar Kaufman, O'Reilly UK, ISBN 156592469X
- "Linux Unleashed", Tim Parker, Bill Ball, David Pitts, Sams, ISBN 0672316889
- "When You Can't Find Your System Administrator", Linda Mui, O'Reilly UK, ISBN 1565921046
- Se você comprar uma distribuição, ela fornecerá um ótimo manual do usuário.

### Editores

- "Learning the Vi Editor", Linda Lamb e Arnold Robbins, O'Reilly UK, ISBN 1565924266
- "GNU Emacs Manual", Richard M. Stallman, iUniverse.Com Inc., ISBN 0595100333
- "Learning GNU Emacs", Debra Cameron, Bill Rosenblatt e Eric Raymond, O'Reilly UK, ISBN 1565921526
- "Perl Cookbook", Tom Christiansen e Nathan Torkington, O'Reilly UK, ISBN 1565922433

### Shells

- "Unix Shell Programming", Stephen G. Kochan e Patrick H. Wood, Sams Publishing, ISBN 067248448X
- "Learning the Bash Shell", Cameron Newham e Bill Rosenblatt, O'Reilly UK, ISBN 1565923472
- "The Complete Linux Shell Programming Training Course", Ellie Quigley e Scott Hawkins, Prentice Hall PTR, ISBN 0130406767
- "Linux and Unix Shell Programming", David Tansley, Addison Wesley Publishing Company, ISBN 0201674726
- "Unix C Shell Field Guide", Gail e Paul Anderson, Prentice Hall, ISBN 013937468X

### X Window

- "Gnome User's Guide", Comunidade Gnome, iUniverse.Com Inc., ISBN 0595132251
- "KDE Bible", por Dave Nash, Hungry Minds Inc., ISBN 0764546929
- "The Concise Guide to XFree86 for Linux", Aron HSiao, Que, ISBN 0789721821
- "The New XFree86", Bill Ball, Prima Publishing, ISBN 0761531521
- "Beginning GTK+ and Gnome", Peter Wright, Wrox Press, ISBN 1861003811



- "KDE 2.0 Development", David Sweet e Matthias Ettrich, Sams Publishing, ISBN 0672318911
- "GTK+/Gnome Application Development", Havoc Pennington, New Riders Publishing, ISBN 0735700788

## Redes

- "TCP/IP Illustrated, Volume I: The Protocols", W. Richard Stevens, Addison-Wesley Professional Computing Series, ISBN 0-201-63346-9
- "DNS and BIND", Paul Albitz, Cricket Liu, Mike Loukides e Deborah Russell, O'Reilly & Associates, ISBN 0596001584
- "The Concise Guide to DNS and BIND", Nicolai Langfeldt, Que, ISBN 0789722739
- "Implementing LDAP", Mark Wilcox, Wrox Press, ISBN 1861002211
- "Understanding and deploying LDAP directory services", Tim Howes e co., Sams, ISBN 0672323168
- "Sendmail", Brian Costales e Eric Allman, O'Reilly UK, ISBN 1565922220
- "Removing the Spam : Email Processing and Filtering", Geoff Mulligan, Addison Wesley Publishing Company, ISBN 0201379570
- "Managing IMAP", Dianna & Kevin Mullet, O'Reilly UK, ISBN 059600012X

## Sites úteis

### Informações gerais

- [The Linux documentation project](#): todos os docs, manpages, HOWTO e FAQs
- [LinuxQuestions.org](#): forum, downloads, docs e muito mais
- [Google para o Linux](#): a ferramenta de busca especializada do Google.
- [Google Groups](#): um arquivo de todos os posts dos grupos de notícias, incluindo a hierarquia comp.os.linux
- [Slashdot](#): notícias diárias
- <http://www.oreilly.com>: livros sobre administração de sistemas Linux e de redes, Perl, Java, ...
- [POSIX](#): o padrão
- [Linux HQ](#): Mantém uma base de dados completa dos códigos-fonte, correções e documentação das várias versões do kernel do Linux.

### Referências por arquitetura

- [AlphaLinux](#): Linux na arquitetura Alpha (p. ex. Digital Workstation)
- [Linux-MIPS](#): Linux no MIPS (p. ex. SGI Indy)
- [Linux on the Road](#): Orientações específicas para a instalação e uso do Linux em laptops, PDAs, telefones celulares e assim por diante. Arquivos de configuração para vários modelos.
- [MkLinux](#): Linux em um Apple

### Distribuições

- [O Projeto Fedora](#): A comunidade patrocinada pela RedHat para manter um sistema.
- [Mandriva](#)
- [Ubuntu](#)
- [Debian](#)
- [TurboLinux](#)
- [Slackware](#)
- [SuSE](#)
- [LinuxISO.org](#): Imagens de CDs para todas as distribuições.
- [Knoppix](#): distribuição que roda a partir de um CD, não é necessário instalar nada.
- [DistroWatch.com](#): procure um Linux que tenha a sua cara.

- ...

## Software

- [Freshmeat](#): novos softwares, e arquivos de softwares.
- [OpenSSH](#): site do Secure SHell
- [OpenOffice](#): suíte de escritório compatível com o MS Office.
- [KDE](#): site do KDE
- [GNU](#): GNU e softwares GNU
- [Gnome](#): O site oficial do Gnome
- [RPM Find](#): todos os pacotes RPM
- [Samba](#): serviços de arquivo e impressão MS Windows
- [Página do Projeto OpenLDAP](#): OpenLDAP server/clients/utilities, FAQ e outros documentos.
- [Página do Sendmail](#): Uma densa discussão técnica sobre os recursos do Sendmail, com exemplos de configuração.
- [Netfilter](#): contém diversas informações sobre iptables: HOWTOs, FAQs, guias, ...
- [site do GIMP](#): Tudo sobre o GNU Image Manipulation Program.
- [SourceForge.net](#): Site de desenvolvimento de softwares Open Source.
- [página do vlm](#)

## Appendix B. Comandos DOS vs comandos Linux

### Abstract

Neste anexo, combinamos os comandos DOS com os seus equivalentes no Linux.

Como uma forma extra de orientação para novos usuários com conhecimentos em Windows, a tabela abaixo lista comandos MS-DOS com suas contra-partes no Linux. Tenha em mente que os comandos Linux normalmente possuem diversas opções. Leia as páginas Info e man para saber mais sobre eles.

**Table B.1. Panorama dos comandos ODS/Linux**

Comandos DOS	Comando Linux
<command> /?	man <command> or <b>command --help</b>
cd	cd
chdir	pwd
cls	clear
copy	cp
date	date
del	rm
dir	ls
echo	echo
edit	vim (ou outro editor)
exit	exit
fc	diff
find	grep
format	mke2fs ou mformat
mem	free
mkdir	mkdir
more	more ou mesmo less
move	mv
ren	mv
time	date

## Appendix C. Recursos do Shell

### Table of Contents

## Abstract

Este documento fornece um panorama dos recursos comuns do shell (os mesmo em todos tipos de shell) e também alguns específicos de um shell.

## Recursos comuns

Os recursos a seguir são padrão em todos os shells. Note que os comandos stop, suspend, jobs, bg e fg estão presentes somente em sistemas que permitem o controle de tarefas.

**Table C.1. Recursos comuns do Shell**

Comando	Significado
>	Redirecionamento de saída
>>	Apensar a um arquivo
<	Redirecionamento de entrada
<<	"Aqui" documento (redirecionamento de entrada)
	Pipe da saída
&	Executar um processo em segundo plano
;	Separa comando na mesma linha
*	Coringa para quaisquer caracteres em um nome de arquivo
?	Coringa para um único caractere em um nome de arquivo
[ ]	Aponta ocorrências para os caracteres delimitados pelos colchetes
( )	Executa o comando em um subshell
` `	Substitui a saída do comando delimitado pelos acentos graves
" "	Aspas parciais (permitem a expansão de variáveis e comandos)
' '	Aspas plenas (sem expansão)
\	Retira o significado especial do caractere imediatamente posterior
\$var	Substitui pelo valor da variável
\$\$	Id do processo
\$0	Nome do comando
\$n	enésimo argumento (n entre 0 e 9)
\$*	Todos os argumentos como uma palavra simples
#	Inicia um comentário
bg	Execução em segundo plano
break	Interrompe loops de instruções
cd	muda de diretório
continue	Retorna a execução do loop de um programa
echo	Exibe saída
eval	Avalia argumentos
exec	Executa um novo shell
fg	Execução em primeiro plano
jobs	Exibe os trabalhos em execução
kill	Encerra trabalhos em execução
newgrp	Muda para um novo grupo
shift	Altera parâmetros de posição
stop	Suspende um trabalho em segundo plano
suspend	Suspende um trabalho em primeiro plano
time	Cronometra um comando
umask	Configura ou exibe as permissões de arquivo
unset	Apaga as definições de variáveis ou funções
wait	Espera o término de um trabalho em segundo plano

## Recursos específicos

A tabela abaixo mostra as principais diferenças entre o shell padrão (**sh**), o Bourne Again SHell (**bash**), o Korn shell (**ksh**) e o C shell (**cs**h).

### Compatibilidade entre shells

Uma vez que o Bourne Again SHell é um sucessor do **sh**, todos os comandos do **sh** também irão funcionar no **bash**, mas a recíproca não é verdadeira. O **bash** possui muitos recursos próprios e, como demonstra a tabela abaixo, muitos outros incorporados de outros shells.

Uma vez que o Turbo C shell é um sucessor do **csh**, todos os comandos do **csh** irão funcionar no **tcsh**, o inverso não é verdadeiro.

**Table C.2. Recursos específicos do shell**

sh	bash	ksh	csh	Significado/Ação
\$	\$	\$	%	Prompt de usuário padrão
	>	>	>!	Redirecionamento forçado
> file 2>&1	&> file OU > file 2>&1	> file 2>&1	>& file	Redireciona stdout e stderr para arquivo
	{ }		{ }	Expande elementos na lista
`comando`	`comando` ou \$(comando)	\$(comando)	`comando`	Substitui a saída do <b>comando</b>
\$HOME	\$HOME	\$HOME	\$home	Diretório home
	~	~	~	Símbolo do diretório home
	~+, ~-, dirs	~+, ~-	=-, =N	Acessa a stack do diretório
var=value	VAR=value	var=value	set var=value	Estipulação de valor para uma variável
export var	export VAR=value	export var=val	setenv var val	Configura uma variável do ambiente
	\${nnnn}	\${nn}		Mais de nove argumentos podem ser informados
"\$@"	"\$@"	"\$@"		Todos os argumentos como palavras separadas
\$#	\$#	\$#	\$#argv	Número de argumentos
\$?	\$?	\$?	\$status	Status de saída do comando mais recente
\$!	\$!	\$!		PID do mais recente processo em segundo plano
\$-	\$-	\$-		Opções Atuais
. file	source file OU . file	. file	source file	Lê os comandos em um arquivo
	alias x='y'	alias x=y	alias x y	Nome <b>x</b> no lugar do comando <b>y</b>
case	case	case	switch ou case	Escolhe alternativas
done	done	done	end	Encerra uma instrução
esac	esac	esac	endsw	End <b>case</b> ou <b>switch</b>
exit n	exit n	exit n	exit (expr)	Encerra com um status
for/do	for/do	for/do	foreach	Realiza um loop através das variáveis
	set -f , set -o nullglob dotglob nocaseglob noglob		noglob	Ignora os caracteres de substituição para a criação de nomes de arquivos
hash	hash	alias -t	hashstat	Exibe os comandos em hash (apelidos lembrados)
hash cmds	hash cmds	alias -t cmds	rehash	Relembra a localização dos comandos
hash -r	hash -r		unhash	Esquece as localizações dos comandos
	history	history	history	Lista os comandos anteriores
	ArrowUp+Enter ou !!	r	!!	Re-executa o comando anterior
	!str	r str	!str	Reinformatar o último comando iniciado por "str"
	!cmd:s/x/y/	r x=y cmd	!cmd:s/x/y/	Substituir "x" por "y" no comando mais recente iniciado por "cmd" e então executá-lo.
if [ \$i -eq 5 ]	if [ \$i -eq 5 ]	if ((i==5))	if (\$i==5)	Teste básico de condição
fi	fi	fi	endif	Termina a condição <b>if</b>

<b>ulimit</b>	<b>ulimit</b>	<b>ulimit</b>	<b>limit</b>	Configura os limites de recursos
<b>pwd</b>	<b>pwd</b>	<b>pwd</b>	<b>dirs</b>	Exibe o diretório atual
<b>read</b>	<b>read</b>	<b>read</b>	<b>\$&lt;</b>	Lê a partir do terminal
<b>trap 2</b>	<b>trap 2</b>	<b>trap 2</b>	<b>onintr</b>	Ignora interrupções
	<b>unalias</b>	<b>unalias</b>	<b>unalias</b>	Remove os aliases
<b>until</b>	<b>until</b>	<b>until</b>		Começa <b>until</b> loop
<b>while/do</b>	<b>while/do</b>	<b>while/do</b>	<b>while</b>	Começa <b>while</b> loop

O Bourne Again SHell possui muitos outros recursos que não foram listados aqui. Esta tabela serve apenas para dar uma idéia de como este shell incorpora todos os recursos úteis de outros shells: não existem espaços em branco na coluna do **bash**. Mais informações sobre os recursos que são exclusivos do Bash podem ser encontrados em suas páginas Info, na seção “Recursos do Bash”

Mais informações:

Você deve pelo menos ler um manual, sendo ele o do seu shell. A melhor escolha seria **info bash**, vez que o **bash** é o shell do GNU e o mais fácil para iniciantes. Imprima-o e leve-o para casa. Estude-o sempre que tiver algum tempo livre.

Leia o [Appendix B, Comandos DOS vs comandos Linux](#) se você estiver com dificuldades de assimilar os comandos do shell.

## Glossário

### Abstract

Esta seção contém uma lista em ordem alfabética dos comandos discutidos neste documento.

### A

a2ps

Formata arquivos para impressão em uma impressora PostScript, ver [the section called “Formatação”](#).

acroread

Visualizador PDF, ver [the section called “Exibindo arquivos pré-formatados”](#).

adduser

Cria uma nova conta de usuário ou atualiza as informações do novo usuário.

alias

Cria um alias no shell para um comando.

alsaconf

Configura o cartão de som usando o driver ALSA, ver [the section called “Driver e arquitetura”](#).

alsamixer

Ajusta o o equipamento de saída de som ALSA, ver [the section called “Controle de volume”](#).

anacron

Executa comandos periodicamente, não assume que a máquina esteja sempre ligada.

apropos

Pesquisa a base de dados whatis por ocorrências, ver o [the section called “Os comandos whatis e apropos”](#).

apt-get

Utilitário manipulador de pacotes APT, ver [the section called “APT”](#).

arecord

Grava uma amostra de som, ver [the section called “Gravação”](#).

aspell

Verificador ortográfico.

at, atq, atrm

Ordena, examina ou deleta trabalhos para execução posterior, ver [the section called “Processos automáticos”](#) e [the section called “O comando at”](#).

aumix

Ajusta o áudio, ver [the section called “Controle de volume”](#).

(g)awk

Linguagem para processamento de dados.

## **B**

bash

Bourne Again SHell, ver [the section called “O shell”](#) e [the section called “Scripts do shell”](#).

batch

Organiza, examina ou deleta trabalhos para posterior execução, ver [the section called “Processos automáticos”](#).

bg

Executa uma tarefa no plano de fundo, ver [the section called “Processos interativos”](#).

bitmap

Editor e conversor bitmap para o Sistema de janelas X.

bzip2

Um compressor de arquivos em blocos, ver [the section called “Compactando e descompactando com o gzip ou o bzip2”](#).

## **C**

cardctl

Gerencia cartões PCMCIA, ver [the section called “Comandos PCMCIA”](#).

cat

Concatena arquivos e imprime na saída padrão, ver [the section called “O básico do básico”](#) and [the section called “Os arquivos de configuração mais importantes”](#).

cd

Muda de diretório, ver [the section called “O básico do básico”](#).

cdp/cdplay

Um programa interativo no modo texto para controlar e tocar CDs de áudio no Linux, ver [the section called “Execução e cópia de CDs”](#).

cdparanoia

Um utilitário de leitura de CDs que inclui recursos adicionais de verificação de dados, ver [the section called “Execução e cópia de CDs”](#).

cdrecord

Grava um CD-R, ver [the section called “Criando uma cópia com o gravador de CDs”](#).

chattr

Muda os atributos dos arquivos.

chgrp

Muda o grupo de proprietários, ver [the section called “A máscara de arquivo”](#).

chkconfig

Atualiza ou consulta informações sobre o nível de execução de serviços do sistema, ver [the section called “Ferramentas”](#).

chmod

Muda as permissões de acesso de arquivos, ver [the section called “Direitos de acesso: A primeira linha de defesa do Linux”](#), [the section called “O comando chmod”](#) e [the section called “Alterando as titularidades de usuário e grupo”](#).

chown

Muda o proprietário e grupo de um arquivo, ver [the section called "A máscara de arquivo"](#).

compress

Comprime arquivos

cp

Copia arquivos e diretórios, ver [the section called "Criando e deletando arquivos e diretórios"](#).

crontab

Mantém arquivos crontab, ver [the section called "Cron e crontab"](#).

csch

Abre um shell C, ver [the section called "O shell"](#).

cut

Remove seções de cada linha do arquivo(s), ver [the section called "Alguns exemplos simples"](#).

## D

date

Exibe ou configura a data e hora do sistema.

dd

Converte e copia um arquivo (despeja o disco), ver [the section called "Usando o dd para copiar dados"](#).

df

Relata o uso do sistema de arquivos, ver [the section called "Pontos de montagem"](#).

dhcpd

Daemon cliente DHCP, ver [the section called "DHCP"](#).

diff

Encontra diferenças entre dois arquivos.

dig

Envia pacotes de consulta de nome de domínio para servidores de nomes [the section called "O comando host"](#).

dmesg

Exibe ou controla o buffer do kernel.

du

Faz uma estimativa do espaço em disco.

dump

Cria um backup do sistema de arquivos, ver [the section called "Ferramentas para a sua distribuição"](#).

## E

echo

Exibe uma linha de texto, ver [the section called "O caminho \(path\)"](#).

ediff

Diff para tradutor para o Inglês.

egrep

Grep extenso

eject

Desmonta e ejeta mídias removíveis, ver [the section called "Usando o CD"](#).

emacs

Inicia o editor Emacs, ver [the section called "GNU Emacs"](#).

exec

Invoca subprocesso(s), ver [the section called "Criação de processos"](#).

exit

Sai do shell, ver [the section called "O básico do básico"](#).

export

Adiciona função(ões) ao ambiente shell, ver [the section called "O caminho \(path\)"](#), [the section called "Exportando variáveis"](#) e [the section called "Alguns exemplos"](#).

## F

fax2ps

Converte um fax TIFF para Postscript, ver [the section called "Formatação"](#).

fdformat

Formata um disquete, ver [the section called "Formatando o disquete"](#).

fdisk

Manipulador da tabela de partições para o Linux, ver [the section called "Disposição e tipos de partições"](#).

fetchmail

Recupera emails de um servidor POP, IMAP, ETRN ou ODMR, ver [the section called "Gerenciadores de emails"](#).

fg

Traz um trabalho para o primeiro plano, ver [the section called "Processos interativos"](#).

file

Determina o tipo de arquivo, ver [the section called "Mais ferramentas"](#).

find

Localiza arquivos, ver [the section called "Find e locate"](#).

firefox

Navegador web, ver [the section called "Navegadores web"](#).

fork

Cria um novo processo, ver [the section called "Criação de processos"](#).

formail

(Re)formatador de emails, ver [the section called "Gerenciadores de emails"](#).

fortune

Imprime um, talvez interessante, adágio aleatório.

ftp

Serviços de transferência de arquivos (inseguro a não ser em caso de conta anônima!) ver [the section called "Clientes FTP"](#).

## G

galeon

Navegador web gráfico.

gdm

Gerenciados de display Gnome, ver [the section called "Init"](#).

gedit

Editor gráfico, ver [the section called "Mas eu quero um editor de textos gráfico!"](#).



(min/a)getty

Controle de equipamentos de console.

gimp

Programa de manipulação de imagens.

gpg

Criptografa, verifica e descriptografa arquivos, ver [the section called "GNU Privacy Guard"](#).

grep

Imprime linhas em que ocorre uma dada seqüência, ver [the section called "O comando grep"](#) e [the section called "Mais sobre o grep"](#).

groff

Emula o comando nroff com o groff, ver [the section called "Formatação"](#).

grub

O shell do grup, ver [the section called "Recursos do GRUB"](#) e [the section called "Atualizando o seu kernel"](#).

gv

Um visualizador PostScript e PDF, ver [the section called "Exibindo arquivos pré-formatados"](#).

gvim

Versão gráfica do editor vim, ver [the section called "Mas eu quero um editor de textos gráfico!"](#).

gzip

Comprime ou expande arquivos, ver [the section called "Compactando e descompactando com o gzip ou o bzip2"](#).

## H

halt

Finaliza o sistema, ver [the section called "Desligamento"](#).

head

Exibe o início de arquivos, ver [the section called "Os comandos head e tail"](#).

help

Exibe ajuda em um comando interno do shell.

host

Utilitário de busca DNS, ver [the section called "O comando host"](#).

httpd

Protocolo de transferência hipertexto Apache, ver [the section called "O comando ip"](#).

## I

id

Exibe UIDs e GIDs reais e efetivos, ver [the section called "Direitos de acesso: A primeira linha de defesa do Linux"](#).

ifconfig

Configura a interface de rede ou exibe a configuração, ver [the section called "PPP, SLIP, PLIP, PPPOE"](#).

info

Lê documentos Info, ver [the section called "As páginas Info"](#).

init

Inicialização de processos de controle, ver [the section called "Criação de processos"](#), [the section called "Init"](#) e [the section called "Níveis de execução do init"](#).

insserv

Gerencia scripts do init, ver [the section called “Ferramentas”](#).

iostat

Exibe estatísticas de E/S, ver [the section called “Recursos de Entrada/Saída”](#).

ip

Exibe/altera o status da interface de rede, ver [the section called “PPP, SLIP, PLIP, PPPOE”](#).

ipchains

Administração de firewall de IP, ver [the section called “Redirecionamento de X11 e TCP”](#).

iptables

Administração de filtros de pacotes IP, ver [the section called “Redirecionamento de X11 e TCP”](#).

## J

jar

Ferramenta de ficheiros Java, ver [the section called “Ficheiros Java”](#).

jobs

Lista as tarefas do plano de fundo.

## K

kdm

Gerente de desktop para o KDE, ver [the section called “Init”](#).

kedit

Editor gráfico do KDE, ver [the section called “Mas eu quero um editor de textos gráfico!”](#).

kill(all)

Encerra processo(s), ver [the section called “Processos interativos”](#).

konqueror

Gerenciador de arquivos, navegador de ajuda, ver [the section called “Fazer uma bagunça...”](#).

ksh

Abre um shell Korn, ver [the section called “O shell”](#).

kwrite

Editor gráfico do KDE, ver [the section called “Mas eu quero um editor de textos gráfico!”](#).

## L

less

**more** com mais recursos, ver [the section called ““less is more””](#).

lilo

Inicializador Linux, ver [the section called “Processo de inicialização, iniciar e desligar”](#).

links

Navegador WWW no modo texto, ver [the section called “O comando ifconfig”](#).

ln

Cria links entre arquivos, ver [the section called “Interligando arquivos”](#).

loadkeys

Carrega as tabelas de tradução do teclado, ver [the section called “Configuração de teclado”](#).

locate

Encontra arquivos, ver [the section called “Find e locate”](#) e [the section called “Cron e crontab”](#).

logout

Fecha o shell atual, ver [the section called “Modo Texto”](#).

lp

Envia pedidos ao serviço de impressão LP, ver [the section called “Imprimir arquivos”](#).

lpc

Programa de controle da impressora na linha de comando, ver [the section called “Imprimir arquivos”](#).

lpq

Programa que informa o status da fila de impressão, ver [the section called “Imprimir arquivos”](#).

lpr

Impressão offline, ver [the section called “Imprimir arquivos”](#).

lprm

Remove pedidos de impressão, ver [the section called “Imprimir arquivos”](#).

ls

Lista o conteúdo de um diretório, ver [the section called “O básico do básico”](#), [the section called “Tipos de arquivos”](#) e [the section called “Mais sobre o ls”](#).

lynx

Navegador WWW no modo texto, ver [the section called “O comando ifconfig”](#).

## **M**

mail

Envia e recebe emails, ver [the section called “Gerenciadores de emails”](#).

man

Leia as páginas man, ver [the section called “As páginas man”](#).

mc

Gerente de arquivos, Midnight Commander, ver [the section called “Fazer uma bagunça...”](#).

mcopy

Copia arquivos MSDOS de/para o Unix.

mdir

Exibe um diretório MSDOS

memusage

Exibe o uso de memória, ver [the section called “Recursos de memória”](#).

memusagestat

Exibe as estatísticas de consumo de memória, ver [the section called “Recursos de memória”](#).

mesg

Controla o acesso de gravação ao seu terminal, ver [the section called “SUID and SGID”](#).

mformat

Adiciona um sistema de arquivos MSDOS a um disquete formatado em baixo nível [the section called “Formatando o disquete”](#).

mkbootdisk

Cria um disquete de inicialização para o sistema em execução.

mkdir

Cria um diretório, ver [the section called “Criando e deletando arquivos e diretórios”](#).

mkisofs

Cria um sistema de arquivos híbrido ISO9660, ver [the section called “Criando uma cópia com o gravador de CDs”](#).

mplayer

Tocador/decodificador de filmes para o Linux, ver [the section called "Tocando arquivos de música"](#) e [the section called "Execução de vídeo, fluxos e televisão"](#).

more

Filtro para exibição de texto uma tela por vez, ver [the section called "'less is more'"](#).

mount

Monta um sistema de arquivos ou exibe informações sobre um sistema de arquivos montado, ver [the section called "Montando um CD"](#).

mozilla

Navegador, ver [the section called "O comando ifconfig"](#).

mt

Controla a operação do drive de fita magnética.

mtr

Ferramenta para diagnóstico de rede.

mv

Renomeia arquivos, [the section called "Criando e deletando arquivos e diretórios"](#).

## N

named

Servidor de nomes de domínio internet, ver [the section called "O Domain Name System"](#).

nautilus

Gerente de arquivos, ver [the section called "Fazer uma bagunça..."](#).

ncftp

Programa navegador de serviços ftp (inseguro!), ver [the section called "Clientes FTP"](#).

netstat

Exibe as conexões de rede, tabelas de roteamento, estatísticas de interface, conexões mascaradas e filiação por difusão seletiva, ver [the section called "AppleTalk"](#) e [the section called "Rsh, rlogin e telnet"](#).

newgrp

Efetua o login em outro grupo, ver [the section called "Efetuando o login em outro grupo"](#).

nfsstat

Exibe estatísticas sobre sistemas de arquivos em rede.

nice

Executa um programa com um cronograma modificado, ver [the section called "Prioridade"](#).

nmap

Ferramenta de exploração de rede e scanner de segurança.

ntpd

Network Time Protocol Daemon, ver [the section called "Data e Fuso Horário"](#).

ntpdate

Configura a data e hora através de um servidor NTP, ver [the section called "Data e Fuso Horário"](#).

ntsysv

Interface simples para configurar níveis de execução, ver [the section called "Ferramentas"](#).

## O

ogle

Tocador de DCD com suporte para menus de DVDs, ver [the section called "Execução de vídeo,](#)

[fluxos e televisão](#)".

## P

passwd

Altera a senha, ver [the section called "O básico do básico"](#) e [the section called "SUID and SGID"](#).

pccardctl

Gerencia cartões PCMCIA, ver [the section called "Comandos PCMCIA"](#).

pdf2ps

Traduz Ghostscript PDF para PostScript, ver [the section called "Formatação"](#)

perl

Practical Extraction e Report Language.

pg

Divide a saída de texto em páginas, [the section called ""less is more""](#).

pgrep

Procura por processos de acordo com os nomes ou outros atributos, ver [the section called "Exibindo informações de processos"](#).

ping

Envia um pedido echo para uma máquina, ver [the section called "O comando ping"](#).

play

Toca uma amostra de som, ver [the section called "Gravação"](#).

pr

Converte arquivos de texto para impressão.

printenv

Exibe todo o ambiente ou parte dele, ver [the section called "Variáveis do ambiente"](#).

procmail

Processador de emails autônomo, ver [the section called "Gerenciadores de emails"](#).

ps

Relata o status de um processo, ver [the section called "Exibindo informações de processos"](#) e [the section called "Recursos de Entrada/Saída"](#).

pstree

Exibe uma árvore de processos, ver [the section called "Exibindo informações de processos"](#).

pwd

Informa qual o diretório atual, ver [the section called "O básico do básico"](#).

## Q

quota

Exibe a capacidade do disco e seus limites, ver [the section called "Sua pasta home"](#).

## R

rcp

Cópia remota (inseguro!)

rdesktop

Cliente Remote Desktop Protocol, ver [the section called "O protocolo rdesktop"](#).

reboot

Finaliza o sistema, ver [the section called "Desligamento"](#).

recode

Converte arquivos para outro conjunto de caracteres, ver [the section called “Idioma”](#).

renice

Altera a prioridade de um processo em execução, ver [the section called “Prioridade”](#).

restore

Recupera backups criados com o **dump**, ver [the section called “Ferramentas para a sua distribuição”](#).

rlogin

Login remoto (telnet, inseguro!), ver [the section called “Rsh, rlogin e telnet”](#) e [the section called “Serviços”](#).

rm

Remove um arquivo, ver [the section called “Criando e deletando arquivos e diretórios”](#).

rmdir

Remove um diretório, ver [the section called “As ferramentas”](#).

roff

Uma pesquisa do sistema roff da configuração de teclado, ver [the section called “Formatação”](#).

rpm

Gerente de Pacotes RPM, ver [the section called “Pacotes RPM”](#).

rsh

Shell remoto (inseguro!), ver [the section called “Rsh, rlogin e telnet”](#).

rsync

Sincroniza dois diretórios, ver [the section called “Usando o rsync”](#).

## S

scp

Cópia remota segura, ver [the section called “Introdução”](#).

screen

Gerente de tela com o emulação VT100, ver [the section called “Processos interativos”](#).

set

Exibir, criar ou alterar uma variável.

setterm

Configura os atributos do terminal.

sftp

FTP Seguro (criptografado), ver [the section called “Introdução”](#).

sh

Abre um shell padrão, ver [the section called “O shell”](#).

shutdown

Encerra o sistema, ver [the section called “Desligamento”](#).

sleep

Espera por um dado período, ver [the section called “Use aquele tempo ocioso!”](#).

slocate

Versão aperfeiçoada com foco em segurança do GNU Locate, ver [the section called “Find e locate”](#).

slrnn

Cliente Usenet no modo texto, ver [the section called “Outras máquinas”](#).

snort

Ferramenta para detecção de invasores na rede.

sort

Organiza linhas de arquivos texto, ver [the section called “Filtrando a saída”](#).

spell

Verificador ortográfico, ver [the section called “Combinando redirecionamentos”](#).

ssh

Shell seguro, ver [the section called “Introdução”](#).

ssh-keygen

Cria, gerencia e converte chaves de autenticação, ver [the section called “Chaves de autenticação”](#).

stty

Muda e exibe as configurações do terminal.

su

Muda de usuário, ver [the section called “O caminho \(path\)”](#), [the section called “APT”](#) e [the section called “O protocolo rdesktop”](#).

**T**

tac

Concatena e imprime arquivos ao contrário, ver [cat](#).

tail

Imprime a última parte de um arquivo, veja [the section called “Os comandos head e tail”](#).

talk

Ferramenta para comunicar-se com outros usuários.

tar

Ficheiro, ver [the section called “Arquivando com o tar”](#).

tcsh

Abre um shell Turbo C, ver [the section called “O shell”](#).

telinit

Inicialização de processos de controle, ver [the section called “Níveis de execução do init”](#).

telnet

Interface para o protocolo TELNET (inseguro), ver [the section called “Rsh, rlogin e telnet”](#).

tex

Formatação de texto e tipografia, ver [the section called “Formatação”](#).

time

Cronometra uma comando ou informa o consumo de recursos, ver [the section called “Quanto tempo leva?”](#).

tin

Programa para leitura de notícias, ver [the section called “Outras máquinas”](#).

top

Exibe uma lista dos processos com maior consumo de CPU, ver [the section called “Exibindo informações de processos”](#), [the section called “Recursos de memória”](#) e [the section called “Recursos de Entrada/Saída”](#).

touch

Altera os carimbos de data/hora de um arquivo, ver [the section called “Crie espaço”](#).

traceroute

Exibe a rota que os pacotes seguem na rede até uma máquina, ver [the section called “O comando](#)

[traceroute](#)".

tripwire

Um verificador de integridade de arquivos para sistemas UNIX, ver [the section called "VNC"](#).

troff

Formata documentos, ver [the section called "Formatação"](#).

tvime

Um programa para assistir televisão de alta qualidade.

twm

Tab Window Manager para o Sistema de Janelas X.

## U

ulimit

Controla recursos, ver [the section called "Limite o tamanho dos arquivos"](#).

umask

Exibe a máscara de criação de arquivos, [the section called "Efetuando o login em outro grupo"](#).

umount

Desmonta sistemas de arquivos.

uncompress

Descomprime arquivos comprimidos.

uniq

Remove linhas em duplicidade de um arquivo, ver [the section called "Filtrando a saída"](#).

up2date

Atualiza pacotes RPM, ver [the section called "Sistemas que usam pacotes RPM"](#).

update

Daemon do kernel para expulsar buffers sujos de volta para o disco.

update-rc.d

Configura os scripts do init [the section called "Ferramentas"](#).

uptime

Exibe o tempo de funcionamento ininterrupto do sistema e o seu consumo médio, ver [the section called "Exibindo informações de processos"](#) e [the section called "Recursos de CPU"](#).

urpmi

Atualiza pacotes RPM, ver [the section called "Sistemas que usam pacotes RPM"](#).

userdel

Deleta uma conta de usuário e arquivos relacionados.

## V

vi(m)

Inicia o editor vi (aprimorado), ver [the section called "Vi\(m\)"](#).

vimtutor

O tutor para o Vim

vmstat

Relata as estatísticas de memória virtual, ver [the section called "Recursos de Entrada/Saída"](#).

## W

w



Mostra quem está logado e o que estão fazendo.

wall

Envia uma mensagem ao terminal de todos, ver [the section called "SUID and SGID"](#).

wc

Exibe o número de bytes, palavras e linhas em arquivos, ver [the section called "O caminho \(path\)"](#).

which

Mostra o caminho completo para um comando (do shell), ver [the section called "O caminho \(path\)"](#) e [the section called "Which"](#).

who

Mostra quem está logado, ver [the section called "SUID and SGID"](#).

who am i

Exibe o ID efetivo do usuário.

whois

Consulta as bases de dados do whois ou nickname, [the section called "O comando whois"](#).

write

Envia uma mensagem para outro usuário, ver [the section called "SUID and SGID"](#).

**X**

xargs

Cria e executa linhas de comando a partir da entrada padrão, ver [the section called "Find e locate"](#).

xauth

Utilitário de arquivos X authority

xawtv

Um programa X11 para assistir TV.

xcdroast

Interface gráfica para o cdrecord, ver [the section called "Criando uma cópia com o gravador de CDs"](#).

xclock

Relógio analógico/digital para o X.

xconsole

Monitora mensagens de console do sistema com o X.

xdm

X Display Manager com suporte para XDMCP, para escolha de máquinas, ver [the section called "Init"](#) e [the section called "O Sistema de Janelas X"](#).

xdvi

Visualizador DVI, ver [the section called "Exibindo arquivos pré-formatados"](#).

xedit

Editor gráfico do X, ver [the section called "Mas eu quero um editor de textos gráfico!"](#).

xf86-config

Servidor de fontes do X.

xhost

Programa de controle de acesso a servidores para o X, ver [the section called "Telnet e X"](#).

xine

Um tocador de vídeos livre, ver [the section called "Execução de vídeo, fluxos e televisão"](#).

xinetd

O daemon extenso para serviços de internet, ver [the section called "\(x\)inetd"](#).

xload

Mostrador do X da carga média do sistema, ver [the section called "Ferramentas gráficas"](#).

xlsfonts

Mostrador do X para o servidor de fontes.

xmms

Tocador de áudio para o X, ver [the section called "Arquivos mp3"](#).

xpdf

Visualizador PDF, ver [the section called "Exibindo arquivos pré-formatados"](#).

xterm

Emulador de terminal do X.

## Y

yast

Ferramenta de administração do sistema no Novel SuSE Linux.

yum

Atualiza pacotes RPM, ver [the section called "Sistemas que usam pacotes RPM"](#).

## Z

zapping

Um visualizador de TV do ambiente Gnome.

zcat

Comprime ou expande arquivos.

zgrep

Busca expressões no interior de arquivos comprimidos.

zmore

Filtro para a visualização de texto comprimido.

## Index

### Symbols

.bashrc, [Redirecionamento de saída com > e |](#), [Arquivos de configuração do shell](#), [Alguns exemplos](#), [Configuração de proxy](#)

.bash\_login

example, [Exemplo de .bash\\_login](#)

.bash\_logout, [Arquivos de configuração do shell](#)

example, [Exemplo de .bash\\_logout](#)

.bash\_profile

example, [Exemplo de .bash\\_profile](#)

.forward, [Email](#)

/etc/fstab, [Montando um CD](#)

, [Usando os recursos Bash](#), [Visual](#)

overview, [Os arquivos de variáveis mais comuns](#)

## A

absolute paths, [Caminhos absolutos e relativos](#)

acoread, [Exibindo arquivos pré-formatados](#)

aliases, [Os arquivos de configuração mais importantes](#)  
alsamixer, [Controle de volume](#)  
apache, [Os arquivos de configuração mais importantes](#)  
Apache, [O servidor web Apache](#)  
apropos, [Os comandos](#), [As páginas man](#)

example, [Os comandos whatis e apropos](#)

apt-get, [O que são pacotes Debian?](#), [APT](#)

installing packages, [APT](#)  
update package list, [APT](#)  
upgrade system, [APT](#)

aptitude, [O que são pacotes Debian?](#)

archiving

burning a CD, [Criando uma cópia com o gravador de CDs](#)  
CD-writer, [Criando uma cópia com o gravador de CDs](#)  
dd, [Usando o dd para copiar dados](#)  
floppies, [Formatando o disquete](#)  
Java archives, [Ficheiros Java](#)  
tapes, [Fazendo backup de dados com um dispositivo de fita](#)  
tools, [Arquivando com o tar](#)  
USB devices, [Backups em/a partir de jazz drives, dispositivos USB e outros removíveis](#)

arecord, [Gravação](#)

arrows, [Usando os recursos Bash](#)

at, [Use aquele tempo ocioso!](#)

example, [O comando at](#)

audio

ALSA, [Driver e arquitetura](#)  
applications, [Outros formatos](#)  
architecture, [Driver e arquitetura](#)  
drivers, [Instalação](#)  
OSS, [Driver e arquitetura](#)  
recording, [Gravação](#)  
volume control, [Controle de volume](#)

aumix, [Controle de volume](#)

automount, [Montando um CD](#)

## B

backslash, [O caminho \(path\)](#), [Caracteres especiais](#)

backups

rsync, [Introdução](#)  
tape device, [Arquivando com o tar](#)  
tar, [Arquivando com o tar](#)  
tools, [Ferramentas para a sua distribuição](#)

Bash

command name completion, [Usando os recursos Bash](#)  
features, [Usando os recursos Bash](#)  
file name completion, [Usando os recursos Bash](#)  
job control, [Processos interativos](#)  
noclobber, [Redirecionamento de saída com > e |](#)

bash, [Tipos de shell](#)

bashrc, [Os arquivos de configuração mais importantes](#), [A máscara de arquivo](#), [Arquivos de configuração do shell](#)

example, [Exemplo de /etc/bashrc](#)

batch, [Processos automáticos](#)

bg, [Processos interativos](#)

bifurcar-e-executar, [Criação de processos](#)

BIOS, [O processo de inicialização](#)

boot partition, [Disposição e tipos de partições](#)

bootstrapping, [Criação de processos](#)

Bourne Again SHell, [Tipos de shell](#)

Bourne Shell, [Tipos de shell](#)

bzip2, [Compactando e descompactando com o gzip ou o bzip2](#)

## C

C shell, [Tipos de shell](#)  
camera

mounting, [Backups em/a partir de jazz drives, dispositivos USB e outros removíveis](#)

cancel, [Removendo trabalhos da fila de impressão](#)

cardctl, [Comandos PCMCIA](#)

cat, [Os comandos, "less is more"](#)

syntax, [Os arquivos de configuração mais importantes](#)

cd, [Os comandos](#)

CD

audio CD playing, [Execução e cópia de CDs](#)

audio CD ripping, [Execução e cópia de CDs](#)

eject, [Ejetando o CD](#)

using, [Usando o CD](#)

CD burning

command line tools, [Criando uma cópia com o gravador de CDs](#)

graphical tools, [Criando uma cópia com o gravador de CDs](#)

CD images

mkisofs example, [Criando uma cópia com o gravador de CDs](#)

cdp, [Execução e cópia de CDs](#)

cdparanoia, [Execução e cópia de CDs](#)

cdplay, [Execução e cópia de CDs](#)

cdrecord, [Criando uma cópia com o gravador de CDs](#)

cdrom, [Os dispositivos mais comuns](#)

character encoding, [Idioma](#)

chgrp, [Alterando as titularidades de usuário e grupo](#)

example, [Alterando as titularidades de usuário e grupo](#)

chkconfig, [Ferramentas](#)

chmod, [O comando chmod](#)

example, [O comando chmod](#)

chown, [Efetuando o login em outro grupo](#), [Alterando as titularidades de usuário e grupo](#)

restrictions, [Alterando as titularidades de usuário e grupo](#)

syntax, [Alterando as titularidades de usuário e grupo](#)

coloured ls output, [Tipos de arquivos](#)

command name completion

, [Usando os recursos Bash](#)

commands

aliases, [Which](#)

arguments, [Observações](#)

basics, [Os comandos](#)

built-ins, [Exceções](#)

entering, [Observações](#)

finding, [Which](#)

multiple man pages, [As páginas man](#)

not found, [Which](#)

options, [Observações](#)

remote execution, [Introdução](#)

synopsis, [As páginas man](#)

configuration files

, [Os arquivos de configuração mais importantes](#)

console, [Os dispositivos mais comuns](#)

controle de trabalho, [Processos interativos](#)

copy interactively, [Removendo arquivos](#)

coreutils, [Mais sobre o ls](#)

cp, [Copiando arquivos](#)

cron, [Use aquele tempo ocioso!](#)

crontab syntax, [Cron e crontab](#)

edit crontab, [Cron e crontab](#)

run-parts, [Cron e crontab](#)

Vixie cron, [Cron e crontab](#)

cron.\*, [Os arquivos de configuração mais importantes](#)

cron.d, [Cron e crontab](#)

crontab, [Os arquivos de configuração mais importantes](#), [Cron e crontab](#)

example, [Cron e crontab](#)

ssh, [Tipos de shell](#)

Ctrl, [Usando os recursos Bash](#)

cua, [Os dispositivos mais comuns](#)

CUPS, [Geral](#)

Cygwin, [Cygwin](#)

## D

daemons

cron, [Cron e crontab](#)

definição, [Daemons](#)

inetd, [\(x\)inetd](#)

dash, [Observações](#), [Tipos de arquivos](#)

data partition, [Disposição e tipos de partições](#)

data partitions, [Disposição e tipos de partições](#)

date

configuration, [Data e Fuso Horário](#)

dd, [Usando o dd para copiar dados](#)

default, [Os arquivos de configuração mais importantes](#), [Init](#)

delete interactively, [Removendo arquivos](#)

desktop

development, [Linux para usuários inexperientes](#)

look and feel, [Linux para usuários inexperientes](#)

menus, [Modo Gráfico](#)

office, [Dez anos de experiência a seu dispor](#)

types, [Qual distribuição devo instalar?](#)

desligamento, [Introdução](#)

dev, [Os dispositivos mais comuns](#)

devices

entries, [Os dispositivos mais comuns](#)

df, [Pontos de montagem](#), [Criando uma cópia com o gravador de CDs](#)

example, [Pontos de montagem](#), [Visual](#)

DHCP, [DHCP](#)

dhcpcd, [DHCP](#)

dict, [Filtro de linha genérico](#)

dig, [O comando host](#)

directories

creation, [Criando diretórios](#)

definition, [Tipos de arquivos](#)

removing, [Removendo arquivos](#)

directory tree, [Visual](#)

DISPLAY, [Variáveis reservadas](#), [Nomes de display](#), [Telnet e X](#), [Redirecionamento de X11 e TCP](#)

distributions

how to chose?, [Qual distribuição devo instalar?](#)

ISO images, [Qual distribuição devo instalar?](#)

mainstream, [Qual distribuição devo instalar?](#)

dmesg, [Backups em/a partir de jazz drives, dispositivos USB e outros removíveis](#)

DOMAIN, [Variáveis reservadas](#)

dots, [Observações](#), [Mais sobre o ls](#)

DrakConf, [Ferramentas](#)

dselect, [O que são pacotes Debian?](#)

dsp, [Os dispositivos mais comuns](#)

dump, [Ferramentas para a sua distribuição](#)

## E

echo, [O caminho \(path\)](#), [Exportando variáveis](#)

example, [Sua pasta home](#)

ed, [Ed](#)  
EDITOR, [Variáveis reservadas](#)  
editors

emacs, [GNU Emacs](#)  
graphical editors, [Mas eu quero um editor de textos gráfico!](#)  
text editor overview, [Qual editor devo usar?](#)  
vim, [Vi\(m\)](#)  
why?, [Por que eu deveria usar um editor?](#)

EGID, [Atributos de processos](#)  
emacs, [GNU Emacs](#)  
encryption

creating encrypted files, [Criptografando seus dados](#)  
decrypting, [Descriptografando arquivos](#)  
export public key, [Sobre a sua chave](#)  
fingerprint, [Sobre a sua chave](#)  
generate keys, [Gerando uma chave](#)  
GnuPG, [GNU Privacy Guard](#)  
key ID, [Sobre a sua chave](#)  
searching keys, [Sobre a sua chave](#)  
why?, [Por que criptografar dados?](#)

environment

graphical, [Introdução](#)  
organization, [Introdução](#)  
show variables, [Geral](#)

escape characters, [Caracteres especiais](#)  
etc, [Os arquivos de configuração mais importantes](#)  
EUID, [Atributos de processos](#)

exemplo, [Atributos de processos](#)

exec, [Serviços](#)  
exit, [Os comandos](#)  
export, [Exportando variáveis](#)

## F

fd, [Os dispositivos mais comuns](#)  
fdformat, [Formatando o disquete](#)  
fdisk, [Disposição e tipos de partições](#)  
fetchmail, [Gerenciadores de emails](#)  
fg, [Processos interativos](#)  
file, [Os comandos](#), [Mais ferramentas](#), [Geral](#)

example, [Mais ferramentas](#)

file descriptors

examples, [Uso dos descritores de arquivo](#), [Analisando erros](#)  
overview, [Uso dos descritores de arquivo](#)

file managers

overview, [Fazer uma bagunça...](#)

file name completion, [Usando os recursos Bash](#)

file permissions

codes, [Direitos de acesso: A primeira linha de defesa do Linux](#)  
display, [Direitos de acesso: A primeira linha de defesa do Linux](#)  
overview, [O comando chmod](#)  
setting, [O comando chmod](#)  
SGID, [Modos especiais](#)  
special modes, [Modos especiais](#)  
SUID, [Modos especiais](#)

file system

hierarchy, [Visual](#)  
layout, [Visual](#)  
orientation, [O caminho \(path\)](#)

file types

colours, [Mais sobre o ls](#)  
determining, [Mais ferramentas](#)  
suffixes, [Mais sobre o ls](#)

## files

abstraction, [Geral](#)  
change ownership recursively, [Alterando as titularidades de usuário e grupo](#)  
changing ownership, [Alterando as titularidades de usuário e grupo](#)  
compressing, [Compactando e descompactando com o gzip ou o bzip2](#)  
concept, [Geral](#)  
copying, [Copiando arquivos](#)  
core dumps, [Limite o tamanho dos arquivos](#)  
decrypting, [Descriptografando arquivos](#)  
display permissions, [Direitos de acesso: A primeira linha de defesa do Linux](#)  
emptying, [Esvaziando arquivos](#)  
encrypting, [Criptografando seus dados](#)  
limit size, [Limite o tamanho dos arquivos](#)  
linking, [Tipos de links](#)  
list of types, [Tipos de arquivos](#)  
logging, [Arquivos de log](#)  
logs, [Mais sobre os arquivos de log](#)  
moving, [Movendo arquivos](#)  
music files, [Arquivos mp3](#)  
naming conventions, [Criando diretórios](#)  
properties, [Mais sobre o ls](#)  
removing, [Removendo arquivos](#)  
renaming, [Movendo arquivos](#)  
searching, [Find e locate](#)  
searching content, [Filtro de linha genérico](#)  
security, [Direitos de acesso: A primeira linha de defesa do Linux](#)  
sharing in a group, [Modos especiais](#)  
size, [Find e locate](#), [Filtrando a saída](#)  
structure, [Geral](#)  
types, [Tipos de arquivos](#)  
uncompressing, [Compactando e descompactando com o gzip ou o bzip2](#)

filesystems, [Os arquivos de configuração mais importantes](#)

## filtering

grep, [Mais sobre o grep](#)  
sort, [Filtrando a saída](#)

find, [Find e locate](#)

example, [Find e locate](#)

## finding files

find, [Find e locate](#)  
locate, [Find e locate](#)  
using the shell, [Usando os recursos do shell](#)

fingerprint, [Sobre a sua chave](#)

## fonts

configuration, [Fontes](#)

formail, [Gerenciadores de emails](#)

fstab, [Pontos de montagem](#), [Os arquivos de configuração mais importantes](#), [Backups em/a partir de jazz drives, dispositivos USB e outros removíveis](#)

## FTP

clients, [Clientes FTP](#)  
secure FTP, [Clientes FTP](#)  
servers, [Servidores FTP](#)

ftp, [Clientes FTP](#)

ftp\*, [Os arquivos de configuração mais importantes](#)

ftp\_proxy, [Configuração de proxy](#)

## G

gdm, [Init](#)  
getty, [Init](#)  
gimp, [Geral](#)  
Gnome System Monitor, [Ferramentas gráficas](#)  
gnome-alsamixer, [Controle de volume](#)  
gnome-cd, [Execução e cópia de CDs](#)  
gnome-sound-recorder, [Gravação](#)

## GNU, [O Linux e o GNU](#)

examples, [O Linux e o GNU](#)  
GNU/Linux, [O GNU/Linux](#)  
public license, [Os prós do Linux](#)  
tools, [O Linux e o GNU](#)

gpdf, [Exibindo arquivos pré-formatados](#)

gpg, [Gerando uma chave](#)

GPL, [Os prós do Linux](#)

greater-than, [Redirecionamento de saída com > e |](#)

grep, [Filtro de linha genérico](#)

example, [Filtro de linha genérico](#)  
I/O redirection, [Mais sobre o grep](#)  
special characters, [Caracteres especiais](#)

groff, [Ferramentas e Idiomas](#)

group, [Os arquivos de configuração mais importantes](#), [Efetuando o login em outro grupo](#)

groups

changing ownership, [Alterando as titularidades de usuário e grupo](#)  
display, [Efetuando o login em outro grupo](#)  
primary group, [Efetuando o login em outro grupo](#)  
switching, [Efetuando o login em outro grupo](#)  
user private groups, [Efetuando o login em outro grupo](#)

## GRUB

, [O processo de inicialização](#)  
features, [Recursos do GRUB](#)

grub.conf

example, [Atualizando o seu kernel](#)

GUI, [Linux para usuários inexperientes](#)

gv, [Geral](#), [Exibindo arquivos pré-formatados](#)

gzip, [Compactando e descompactando com o gzip ou o bzip2](#)

## H

halt, [Desligamento](#)

hd, [Os dispositivos mais comuns](#)

head, [Os comandos head e tail](#)

help

/usr/share/doc, [Exceções](#)  
example, [A opção --help](#)  
Gnome Help, [Ajuda gráfica](#)  
graphical, [Ajuda gráfica](#)  
option, [A opção --help](#)

HISTSIZE, [Variáveis reservadas](#)

home, [Sua pasta home](#)

HOME, [Sua pasta home](#), [Variáveis reservadas](#)

home directory

definition, [Sua pasta home](#)  
example, [Fazer uma bagunça...](#)  
limitations, [Sua pasta home](#)  
path, [Sua pasta home](#)  
usage, [Sua pasta home](#)

host, [O comando host](#)

HOSTNAME, [Variáveis reservadas](#)

hosts, [Os arquivos de configuração mais importantes](#), [/etc/hosts](#)

http\_proxy, [Configuração de proxy](#)

## I

I/O redirection

append output, [O operador >>](#)  
combining, [Combinando redirecionamentos](#)  
emptying files, [Redirecionamento de saída com > e |](#)  
null command, [Esvaziando arquivos](#)  
prevent overwrite, [Redirecionamento de saída com > e |](#)  
printing, [Encaminhando o arquivo à impressora](#)  
saving output in a file, [Redirecionamento de saída com > e |](#)



splitting stdout and stderr, [Separando a saída padrão do erro padrão](#)  
write to stdout and file, [Escrevendo para a saída e arquivos simultaneamente](#)

id, [Direitos de acesso: A primeira linha de defesa do Linux](#), [Efetuando o login em outro grupo](#)  
ifconfig, [O comando ip](#)  
IMAP, [Servidores de email remotos](#)  
inetd, [\(x\)inetd](#)  
inetd.conf, [Os arquivos de configuração mais importantes](#), [\(x\)inetd](#)  
info, [Os comandos](#), [As páginas Info](#)

example, [As páginas Info](#)  
info info, [As páginas Info](#)

Info pages, [As páginas Info](#)  
info pages

browsing, [As páginas Info](#)

inicialização, [Introdução](#)

BIOS, [O processo de inicialização](#)  
carregador de inicialização, [O processo de inicialização](#)  
carregamento direto, [Recursos do GRUB](#)  
GRUB, [Introdução](#)  
init, [Init](#)  
LILO, [Introdução](#)  
MBR, [O processo de inicialização](#)

init, [Init](#)

kill scripts, [Init](#)  
run levels, [Níveis de execução do init](#)  
start scripts, [Init](#)  
SysV/BSD, [Init](#)

init scripts

administration, [Ferramentas](#)

init.d, [Ferramentas](#)  
inittab, [Os arquivos de configuração mais importantes](#), [Init](#)  
inode

definition, [O sistema de arquivos na realidade](#)  
stored information, [O sistema de arquivos na realidade](#)

input

default input, [O que são entrada e saída padrão?](#)  
redirection, [Redirecionamento de entrada](#)  
standard input, [O que são entrada e saída padrão?](#)

INPUTRC, [Variáveis reservadas](#)

inputrc, [Fontes](#)  
insserv, [Ferramentas](#)  
installing software

package managers, [O Linux e o GNU](#)

Internet Protocol, [TCP/IP](#)

iostat, [Recursos de Entrada/Saída](#)  
ip, [O comando ip](#)  
ir, [Os dispositivos mais comuns](#)  
ISC, [O Domain Name System](#)  
isdn, [Os dispositivos mais comuns](#)  
issue, [Os arquivos de configuração mais importantes](#)

## J

jar, [Ficheiros Java](#)  
jobs, [Processos interativos](#)  
journaled file system, [Por que particionar?](#)  
js, [Os dispositivos mais comuns](#)

## K

kaudiocreator, [Execução e cópia de CDs](#)  
kdm, [Init](#)  
kdvi, [Exibindo arquivos pré-formatados](#)  
kernel

functions, [O kernel](#)  
upgrade, [Atualizando o seu kernel](#)

key combinations, [Usando os recursos Bash](#)  
keyboard

layout, [Configuração de teclado](#)  
XkbdLayout, [Configuração de teclado](#)

KeyBoard

switching layouts, [Configuração de teclado](#)

kill, [Processos interativos](#), [Sinais](#)

example, [Interrompendo processos](#)

kmem, [SUID and SGID](#)  
kmix, [Controle de volume](#)  
Konqueror

, [Ajuda gráfica](#)

konqueror, [Fazer uma bagunça...](#)

Korn shell, [Tipos de shell](#)

kpdf, [Exibindo arquivos pré-formatados](#)

krec, [Gravação](#)

ksh, [Tipos de shell](#)

## L

LANG, [Variáveis reservadas](#), [Idioma](#)

LANGUAGE, [Idioma](#)

LBA, [Recursos do GRUB](#)

ld.so.conf, [Os arquivos de configuração mais importantes](#)

LDAP, [LDAP](#)

LD\_LIBRARY\_PATH, [Variáveis reservadas](#)

LEIAME, [Compactando e descompactando com o gzip ou o bzip2](#)

less, ["less is more"](#)

pager for man, [As páginas man](#)

less-than, [Redirecionamento de entrada](#)

LILO, [Introdução](#)

lilo.conf, [Os arquivos de configuração mais importantes](#)

links

creating, [Criando links simbólicos](#)

definition, [Tipos de arquivos](#)

hard links, [Tipos de links](#)

inodes, [Tipos de links](#)

mechanism, [Tipos de links](#)

saving space, [Economize espaço com links](#)

symbolic links, [Tipos de links](#)

types, [Tipos de links](#)

usage, [Criando links simbólicos](#)

user-space link, [Tipos de links](#)

Linus

creator, [Linus e Linux](#)

Linux

advantages, [Os prós do Linux](#)

applications, [O atual uso dos sistemas Linux](#)

bugs, [Os prós do Linux](#)

disadvantages, [Os contras do Linux](#)

distributions, [O Linux e o GNU](#)

ease of use, [Linux para usuários inexperientes](#)

free, [Os prós do Linux](#)

goals, [Linus e Linux](#)

origin, [UNIX](#)

portability, [Os prós do Linux](#)

properties, [Os prós do Linux](#)

robustness, [Os prós do Linux](#)

scalability, [Os prós do Linux](#)

security, [Os prós do Linux](#)

user interface, [O Linux é difícil?](#)

In, [Criando links simbólicos](#)  
load, [Carga](#)  
loadkeys, [Configuração de teclado](#)  
localhost, [/etc/hosts](#)  
locate, [Find e locate](#)  
log files

syslog, [Os arquivos de variáveis mais comuns](#)

login, [Arquivos de configuração do shell](#)

authentication, [Introdução](#)  
graphical, [Modo Gráfico](#)  
language configuration, [Idioma](#)  
root, [Modo Gráfico](#)  
text mode, [Modo Texto](#)

LOGNAME, [Variáveis reservadas](#)

logout, [Modo Texto](#)

graphical, [Modo Gráfico](#)

logrotate.\*, [Os arquivos de configuração mais importantes](#)

lp, [Os dispositivos mais comuns](#), [Encaminhando o arquivo à impressora](#)

lpq, [Status das suas impressões](#)

lpr, [Encaminhando o arquivo à impressora](#)

lprm, [Removendo trabalhos da fila de impressão](#)

lpstat, [Status das suas impressões](#)

ls, [Os comandos](#), [Observações](#), [Tipos de arquivos](#), [Mais sobre o ls](#), [Direitos de acesso: A primeira linha de defesa do Linux](#)

coloured output, [Mais sobre o ls](#)  
example, [Mais sobre o ls](#)

## M

mail, [Os arquivos de configuração mais importantes](#)

anti-spam/anti-vírus, [Servidores](#)  
client applications, [Gerenciadores de emails](#)  
mailing lists, [Servidores](#)  
protocols, [Servidores de email remotos](#)  
servers overview, [Servidores](#)

MAIL, [Variáveis reservadas](#)

man, [Os comandos](#)

example, [As páginas man](#)

man pages

browsing, [As páginas man](#)  
graphical, [Ajuda gráfica](#)  
man man, [As páginas man](#)  
printing, [Ferramentas e Idiomas](#)  
sections, [As páginas man](#)  
structure, [As páginas man](#)

MANPATH, [Exportando variáveis](#), [Variáveis reservadas](#)

mask, [A máscara de arquivo](#)

MBR, [O processo de inicialização](#)

mc, [Fazer uma bagunça...](#)

mem, [Os dispositivos mais comuns](#), [SUID and SGID](#)

messages

language, [Idioma](#)

mformat, [Formatando o disquete](#)

midi, [Os dispositivos mais comuns](#)

mixer, [Os dispositivos mais comuns](#)

mkbootdisk, [Formatando o disquete](#)

mkdir, [Criando diretórios](#)

mkisofs, [Criando uma cópia com o gravador de CDs](#)

modem, [Os dispositivos mais comuns](#)

modo com um único usuário, [Init](#)

modules.conf, [Os arquivos de configuração mais importantes](#), [Backups em/a partir de jazz drives](#), [dispositivos USB e outros removíveis](#)

Monitor de Sistemas do Gnome, [Exibindo informações de processos](#)

monitoring, [Ferramentas gráficas](#)

more, ["less is more"](#)

motd, [Os arquivos de configuração mais importantes](#)

mount, [Montando um CD](#)

example, [Montando um CD](#)

mount a CD, [Montando um CD](#)

mount point, [Pontos de montagem](#)

example, [Pontos de montagem](#)

mouse, [Os dispositivos mais comuns](#)

mplayer, [Arquivos mp3](#)

mt, [Fazendo backup de dados com um dispositivo de fita](#)

mtab, [Os arquivos de configuração mais importantes](#)

music, [Os dispositivos mais comuns](#)

mv, [Movendo arquivos](#)

## N

named, [O Domain Name System](#)

named pipes

definition, [Tipos de arquivos](#)

nameserver, [O Domain Name System](#)

nautilus, [Ajuda gráfica](#), [Fazer uma bagunça...](#)

ncftp, [Clientes FTP](#)

netstat, [Recursos de Entrada/Saída](#), [Serviços](#)

network configuration

files, [Arquivos de configuração de rede](#)

graphical tools, [Configuração de interfaces de rede](#)

networking

chatting and conferencing, [Conferências e Bate-papo](#)

check other hosts, [O comando ping](#)

device names, [O comando ip](#), [Nomes das interfaces de rede](#)

display IP address, [O comando ip](#)

display routes, [O comando ip](#), [Verificando a configuração da máquina com o netstat](#)

host lookup, [O comando host](#)

IETF, [TCP/IP](#)

info on hosts, [O comando whois](#)

local loop, [O comando ip](#)

OSI Model, [O modelos OSI](#)

PCMCIA, [Comandos PCMCIA](#)

protocol overview, [Alguns protocolos populares de rede](#)

protocols, [O modelos OSI](#)

remote execution, [Introdução](#)

Samba, [SMB/NMB](#)

show IP address, [O comando ifconfig](#)

trace hosts, [O comando traceroute](#)

newgrp, [Efetuando o login em outro grupo](#)

nice, [Prioridade](#)

nice number, [Prioridade](#)

nível de execução, [Init](#)

exibir, [Init](#)

noclobber, [Redirecionamento de saída com > e |](#)

nsswitch.conf, [Os arquivos de configuração mais importantes](#), [/etc/nsswitch.conf](#)

NTP, [Data e Fuso Horário](#)

ntpdate, [Data e Fuso Horário](#)

ntsysv, [Ferramentas](#)

null, [Os dispositivos mais comuns](#)

## O

office suits

best practice, [Uso geral dos documentos de escritório](#)

history, [História](#)

Open Source

achievements, [Open Source](#)

community, [Dez anos de experiência a seu dispor](#)

other projects, [Dez anos de experiência a seu dispor](#)

reliability, [Os contras do Linux](#)

what is it?, [Open Source](#)

OpenOffice.org, [História](#)  
options

--help, [A opção --help](#)  
long options, [Observações](#)

OS, [Geral](#), [Variáveis reservadas](#)  
OSI Model, [O modelos OSI](#)  
OSTYPE, [Variáveis reservadas](#)  
output

default output, [O que são entrada e saída padrão?](#)  
redirection with >, [Redirecionamento de saída com > e |](#)  
redirection with |, [Redirecionamento de saída com > e |](#)  
standard output, [O que são entrada e saída padrão?](#)

## P

packages

APT, [APT](#)  
DEB, [O que são pacotes Debian?](#)  
install from CD, [Montando um CD](#)  
install RPM, [Exemplos de RPM](#)  
kernel, [Atualizando o seu kernel](#)  
RPM, [O que é um RPM?](#)  
source, [Pacotes fonte](#)  
Update Agent, [Sistemas que usam pacotes RPM](#)  
updates, [Observações gerais](#)  
URPMI, [Sistemas que usam pacotes RPM](#)  
YaST, [Sistemas que usam pacotes RPM](#)  
YOU, [Sistemas que usam pacotes RPM](#)  
yum, [Sistemas que usam pacotes RPM](#)

PAGER, [Variáveis reservadas](#)

PAM, [PAM](#)

pam.d, [Os arquivos de configuração mais importantes](#)

par, [Os dispositivos mais comuns](#)

partitions

creation, [Disposição e tipos de partições](#)  
file system, [O sistema de arquivos na realidade](#)  
layout, [Disposição e tipos de partições](#)  
mount points, [Pontos de montagem](#)  
root partition, [Disposição e tipos de partições](#), [Pontos de montagem](#)  
server layout, [Disposição e tipos de partições](#)  
swap, [Disposição e tipos de partições](#)  
types, [Disposição e tipos de partições](#)  
var partition, [Os arquivos de variáveis mais comuns](#)  
why?, [Por que particionar?](#)  
workstation layout, [Disposição e tipos de partições](#)

passwd, [Os comandos](#), [Tipos de shell](#), [Os arquivos de configuração mais importantes](#), [Efetuando o login em outro grupo](#), [SUID and SGID](#), [Tradicionais](#)

PATH, [O caminho \(path\)](#), [Variáveis reservadas](#)

configuration, [O caminho \(path\)](#), [Which](#)  
example, [O caminho \(path\)](#)

paths

examples, [Caminhos absolutos e relativos](#)  
types, [Caminhos absolutos e relativos](#)

pccardctl, [Comandos PCMCIA](#)

performance, [Performance](#)

permissões de arquivo

SGID, [SUID and SGID](#)  
SUID, [SUID and SGID](#)

pg, ["less is more"](#)

pgrep, [Exibindo informações de processos](#)

PID, [Atributos de processos](#)

ping, [O comando ping](#)

pipe, [Redirecionamento de saída com > e |](#)

piping

examples, [Redirecionamento de saída com > e |](#)

POP3, [Servidores de email remotos](#)

Postfix, [Servidores](#)

power button, [Modo Texto](#)

power off, [Modo Texto](#)

PPID, [Atributos de processos](#)

prefdm, [Init](#)

printcap, [Os arquivos de configuração mais importantes](#)

printenv, [Geral](#)

printers

choosing, [Comprando uma impressora para o Linux](#)

PostScript, [Comprando uma impressora para o Linux](#)

printer configuration, [Configuração gráfica da impressora](#)

printing

CUPS, [Geral](#)

formatting tools, [Ferramentas e Idiomas](#)

languages overview, [Ferramentas e Idiomas](#)

lp/lpr, [Encaminhando o arquivo à impressora](#)

man pages, [Ferramentas e Idiomas](#)

preview, [Exibindo arquivos pré-formatados](#)

remove jobs, [Removendo trabalhos da fila de impressão](#)

status, [Status das suas impressões](#), [Status da sua impressora](#)

troubleshooting, [Arquivo errado](#)

proc, [Exibindo informações de processos](#)

processes

automatic, [Processos automáticos](#)

changing priority, [Interrompendo processos](#)

init scripts, [Init](#)

job control, [Processos interativos](#)

job control overview, [Processos interativos](#)

manage load, [Interrompendo processos](#)

managing, [Trabalho para o administrador do sistema](#)

priority, [Prioridade](#)

scheduling, [Use aquele tempo ocioso!](#)

timing, [Quanto tempo leva?](#)

processo

finalizando, [Finalizando processos](#)

processos

bifurcar, [Criação de processos](#)

códigos de retorno, [Finalizando processos](#)

criação, [Criação de processos](#)

daemons, [Daemons](#)

exec, [Criação de processos](#)

exibição contínua, [Exibindo informações de processos](#)

exibindo, [Atributos de processos](#)

interativos, [Processos interativos](#)

numero nice, [Atributos de processos](#)

parando, [Sinais](#)

plano de fundo, [Processos interativos](#)

primeiro plano, [Processos interativos](#)

propriedade, [Atributos de processos](#)

relações, [Exibindo informações de processos](#)

sinal, [Sinais](#)

transformar em daemon, [Criação de processos](#)

visualisar, [Exibindo informações de processos](#)

zumbi, [Criação de processos](#)

procmail, [Gerenciadores de emails](#)

profile, [Os arquivos de configuração mais importantes](#), [A máscara de arquivo](#)

example, [Exemplo /etc/profile](#)

profile.d, [O diretório profile.d](#)

prompt

backup, [Introdução](#)

configuration, [Introdução](#)

examples, [Alguns exemplos](#)

protocols

AppleTalk, [AppleTalk](#)  
DHCP, [DHCP](#)  
IPv6, [TCP/IPv6](#)  
ISDN, [ISDN](#)  
LDAP, [LDAP](#)  
PPP, [PPP](#), [SLIP](#), [PLIP](#), [PPPOE](#)  
RDP, [O protocolo rdesktop](#)  
SMB/NMB, [SMB/NMB](#)  
TCP/IP, [TCP/IP](#)

proxy

configuration, [Configuração de proxy](#)

ps, [Recursos de Entrada/Saída](#)

example with options, [Exibindo informações de processos](#)  
exemplo simples, [Exibindo informações de processos](#)

PS1, [Variáveis reservadas](#)

PS2, [Variáveis reservadas](#)

pstree, [Exibindo informações de processos](#)

pty, [Os dispositivos mais comuns](#)

pwd, [Os comandos](#)

PWD, [Variáveis reservadas](#)

## Q

Qmail, [Servidores](#)

quota, [Sua pasta home](#)

## R

radio, [Os dispositivos mais comuns](#)

ram, [Os dispositivos mais comuns](#)

ranges

substitution, [Usando os recursos do shell](#)

rc\*, [Os arquivos de configuração mais importantes](#)

rc\*.d, [Init](#)

rdesktop, [O protocolo rdesktop](#)

README, [Geral](#)

reboot, [Desligamento](#)

recode, [Idioma](#)

regular expressions

getting started, [Mais sobre o grep](#)

regular files

, [Tipos de arquivos](#)

relative paths, [Caminhos absolutos e relativos](#)

remote execution

Cygwin, [Cygwin](#)

RDP, [O protocolo rdesktop](#)

security, [Rsh, rlogin e telnet](#)

VNC, [VNC](#)

X Window, [Telnet e X](#)

remove interactively, [Removendo arquivos](#)

rename interactively, [Removendo arquivos](#)

renice, [Prioridade](#)

resolv.conf, [Os arquivos de configuração mais importantes](#), [/etc/resolv.conf](#), [O Domain Name System](#)

restore, [Ferramentas para a sua distribuição](#)

RGID, [Atributos de processos](#)

rlogin, [Rsh, rlogin e telnet](#), [Serviços](#)

rm, [Removendo arquivos](#)

rmdir

, [Removendo arquivos](#)

root

file creation mask, [A máscara de arquivo](#)

login, [Modo Texto](#)

root directory, [Visual](#)

example, [Visual](#)

root login, [Observações](#)  
root partition, [Disposição e tipos de partições](#)  
route, [O comando ip](#)  
RPM

installation, [Exemplos de RPM](#)  
remove package, [Exemplos de RPM](#)  
search files, [Exemplos de RPM](#)  
search packages, [Exemplos de RPM](#)

rpm, [Exemplos de RPM](#)  
rsh, [Rsh, rlogin e telnet](#), [Serviços](#)  
rsync, [Introdução](#)  
RUID, [Atributos de processos](#)  
run level

rc files, [Init](#)

run levels

switching, [Níveis de execução do init](#)

run-parts, [Cron e crontab](#)

## S

scheduling

at, [O comando at](#)  
batch, [O comando at](#)  
cron, [Cron e crontab](#)  
execution time, [O comando at](#)  
sleep, [O comando sleep](#)  
types, [Use aquele tempo ocioso!](#)

scp, [Cópia remota segura](#)

example, [Cópia remota segura](#)

screen, [Processos interativos](#)

script

example, [Alguns exemplos simples](#)  
first line, [O que são scripts?](#)  
init script example, [Alguns exemplos simples](#)

scripts

chmod, [O que são scripts?](#)

sd, [Os dispositivos mais comuns](#)

Secure SHell

, [Introdução](#)  
authentication keys, [Chaves de autenticação](#)  
example, [Introdução](#)  
secure copy, [Cópia remota segura](#)  
server authentication, [Autenticação com o servidor](#)  
X11Forwarding, [Redirecionamento de X11 e TCP](#)

security

check open ports, [Serviços](#)  
considerations, [Introdução](#)  
disable services, [Serviços](#)  
file permissions, [Direitos de acesso: A primeira linha de defesa do Linux](#)  
firewall, [O que é um firewall?](#)  
intrusion detection, [Detecção de invasões](#), [Eu fui invadido?](#)  
log files, [Os arquivos de variáveis mais comuns](#)  
logs, [Arquivos de log](#)  
packet filters, [Filtros de pacotes](#)  
proxies, [Proxies](#)  
recovery, [Recuperando-se de uma invasão](#)  
remote commands, [Rsh, rlogin e telnet](#)  
Secure SHell, [Introdução](#)  
TCP wrappers, [TCP wrappers](#)  
tips, [Mais dicas](#)



updates, [Atualize o sistema com regularidade](#)

Sendmail, [Servidores](#)

sendmail.cf, [Os arquivos de configuração mais importantes](#)

sequencer, [Os dispositivos mais comuns](#)

server types

managed services, [\(x\)inetd](#)

standalone, [Servidores autônomos](#)

servers

authentication, [Tradicionais](#)

DHCP, [DHCP](#)

DNS, [O Domain Name System](#)

FTP, [Servidores FTP](#)

mail, [Servidores](#)

news, [Serviços de notícias](#)

proxy, [O que é um servidor proxy?](#), [Proxies](#)

web, [O servidor web Apache](#)

services, [Os arquivos de configuração mais importantes](#)

setfont, [Fontes](#)

sftp, [Clientes FTP](#)

example, [Cópia remota segura](#)

SGID, [Modos especiais](#), [SUID and SGID](#)

SGML, [Ferramentas e Idiomas](#)

sh, [Tipos de shell](#)

shadow, [Tradicionais](#)

shell

definition, [O que é um shell?](#)

history, [Filtro de linha genérico](#)

scripts, [O que são scripts?](#)

substitution, [Usando os recursos do shell](#)

switching, [Tipos de shell](#)

types, [Tipos de shell](#)

SHELL, [Qual shell estou usando?](#), [Geral](#), [Variáveis reservadas](#)

shells, [Tipos de shell](#)

Shift, [Usando os recursos Bash](#)

shutdown, [Init](#), [Desligamento](#)

SIGHUP, [Sinais](#)

SIGINT, [Sinais](#), [Interrompendo processos](#)

SIGKILL, [Sinais](#), [Interrompendo processos](#)

signals

visão geral, [Sinais](#)

SIGTERM, [Sinais](#), [Interrompendo processos](#)

slapd, [LDAP](#)

slash, [Visual](#)

backslash, [O caminho \(path\)](#)

forward, [Observações](#), [Disposição e tipos de partições](#)

sleep, [Use aquele tempo ocioso!](#)

examples, [O comando sleep](#)

smb.conf, [SMB/NMB](#)

sndconfig, [Os arquivos de configuração mais importantes](#)

sockets

definition, [Tipos de arquivos](#)

software

installing, [Geral](#)

sort, [Filtrando a saída](#)

file size, [Filtrando a saída](#)

unique, [Filtrando a saída](#)

sound, [Os arquivos de configuração mais importantes](#)

source, [Exemplo de .bash\\_profile](#)

special files

definition, [Tipos de arquivos](#)

spool, [Os arquivos de variáveis mais comuns](#)

square braces, [As páginas man](#), [Usando os recursos do shell](#)

ssh, [Os arquivos de configuração mais importantes](#), [Rsh, rlogin e telnet](#), [Introdução](#)

ssh-keygen, [Chaves de autenticação](#)

sshd\_config, [Introdução](#)

standard error, [Uso dos descritores de arquivo](#)

status de finalização, [Finalizando processos](#)

stdin, [O que são entrada e saída padrão?](#)

stdout, [O que são entrada e saída padrão?](#)

sticky bit, [Modos especiais](#)

su, [O caminho \(path\)](#)

sudo, [Modo Texto](#)

SUID, [Modos especiais](#), [SUID and SGID](#)

swap partition, [Disposição e tipos de partições](#)

example, [Disposição e tipos de partições](#)

swapping, [Recursos de memória](#)

symbolic links, [Criando links simbólicos](#)

sysconfig, [Os arquivos de configuração mais importantes](#), [Init](#)

system

CPU resources, [Recursos de CPU](#)

I/O resources, [Recursos de Entrada/Saída](#)

load, [Carga](#)

memory resources, [Recursos de memória](#)

monitoring, [Ferramentas gráficas](#)

performance, [Performance](#)

updates, [Observações gerais](#)

## T

tail, [Os comandos head e tail](#), [Esvaziando arquivos](#)

tar, [Arquivando com o tar](#)

compressed files, [Compactando e descompactando com o gzip ou o bzip2](#)

example, [Arquivando com o tar](#)

GNU tar, [Arquivando com o tar](#)

incremental backups, [Adição ao backup com o tar](#)

options, [Arquivando com o tar](#)

tarballs, [Arquivando com o tar](#)

tcsh, [Tipos de shell](#)

tee, [Escrevendo para a saída e arquivos simultaneamente](#)

telinit, [Níveis de execução do init](#)

telnet, [Rsh, rlogin e telnet](#), [Serviços](#)

examples, [Rsh, rlogin e telnet](#)

temporary files, [Os arquivos de variáveis mais comuns](#)

TERM, [Geral](#), [Variáveis reservadas](#)

terminal

send a message, [SUID and SGID](#)

starting up, [Modo Gráfico](#)

view, [Modo Gráfico](#)

TeX, [Ferramentas e Idiomas](#)

text mode

logout, [Modo Texto](#)

view, [Modo Texto](#)

tilde, [Sua pasta home](#)

time, [Quanto tempo leva?](#)

time zone

configuration, [Data e Fuso Horário](#)

timezone, [Data e Fuso Horário](#)

tmp, [Os arquivos de variáveis mais comuns](#)

to chmod, [O comando chmod](#)

top, [Exibindo informações de processos](#), [Recursos de memória](#), [Recursos de Entrada/Saída](#)

changing process priority, [Interrompendo processos](#)

touch, [Esvaziando arquivos](#)

traceroute, [O comando traceroute](#)

Transport Control Protocol, [TCP/IP](#)  
troff, [Ferramentas e Idiomas](#)  
truncating, [Redirecionamento de saída com > e |](#), [Esvaziando arquivos](#)  
tty, [Os dispositivos mais comuns](#)  
TTY, [Atributos de processos](#)  
tzconfig, [Data e Fuso Horário](#)

## U

UID, [Variáveis reservadas](#)  
ulimit, [Limite o tamanho dos arquivos](#)  
umask, [A máscara de arquivo](#)  
UNIX

origin, [UNIX](#)

up2date, [Sistemas que usam pacotes RPM](#)  
update-rc.d, [Ferramentas](#)  
uptime, [Exibindo informações de processos](#), [Recursos de CPU](#)  
usb, [Os dispositivos mais comuns](#)  
USB

dmesg, [Backups em/a partir de jazz drives, dispositivos USB e outros removíveis](#)  
mount device, [Um exemplo: rsync em um dispositivo de armazenamento USB](#)

user name

display, [Direitos de acesso: A primeira linha de defesa do Linux](#)

USER(NAME), [Variáveis reservadas](#)

users

classification, [Usuários](#)

UTF-8, [Idioma](#)

## V

var, [Os arquivos de variáveis mais comuns](#)

variable files

mail, [Os arquivos de variáveis mais comuns](#)  
spool directories, [Os arquivos de variáveis mais comuns](#)  
web server data, [Os arquivos de variáveis mais comuns](#)

variables

export, [Exportando variáveis](#)  
reserved variables, [Variáveis reservadas](#)  
show content, [Exportando variáveis](#)

vi, [Vi\(m\)](#)

insert mode, [Comandos que alteram o editor para o modo de inserção.](#)  
modes, [Dois modos](#)  
navigation, [Movendo-se através do texto](#)

video, [Os dispositivos mais comuns](#)

applications, [Execução de vídeo, fluxos e televisão](#)

vim, [Vi\(m\)](#)

gvim, [Vi\(m\)](#)  
tutor, [O modo fácil](#)

vimtutor, [O modo fácil](#)

VISUAL, [Variáveis reservadas](#)

vmstat, [Recursos de Entrada/Saída](#)

VNC, [VNC](#)

VoIP

clients, [Quanto ao cliente](#)  
service providers, [Quanto ao servidor](#)  
what?, [O que é?](#)

## W

wall, [SUID and SGID](#)  
wc, [O caminho \(path\)](#)

web browsers

overview, [Navegadores web](#)

whatis

example, [Os comandos whatis e apropos](#)

which, [O caminho \(path\)](#), [Which](#)

who, [Init](#)

example, [SUID and SGID](#)

whois, [O comando whois](#)

window manager, [Gerenciadores de janela e desktop](#)

write

, [SUID and SGID](#)

## X

X Window

application overview, [O Sistema de Janelas X](#)

clients, [O Sistema de Janelas X](#)

configuration, [Configuração do servidor X](#)

definition, [O Sistema de Janelas X](#)

display names, [Nomes de display](#)

display number, [Nomes de display](#)

features, [Recursos do X](#)

hostname, [Nomes de display](#)

screen number, [Nomes de display](#)

X.Org, [Configuração do servidor X](#)

X11, [Os arquivos de configuração mais importantes](#)

xargs, [Find e locate](#)

Xauthority, [Redirecionamento de X11 e TCP](#)

xcdroast, [Criando uma cópia com o gravador de CDs](#)

xdvi, [Exibindo arquivos pré-formatados](#)

XENVIRONMENT, [Variáveis reservadas](#)

XF86Config, [Configuração do servidor X](#)

KeyBoard, [Configuração de teclado](#)

XFILESEARCHPATH, [Variáveis reservadas](#)

xhost, [Telnet e X](#)

xinetd, [\(x\)inetd](#)

xinetd.\*, [Os arquivos de configuração mais importantes](#)

xinetd.d, [\(x\)inetd](#)

xkill, [Interrompendo processos](#)

xload, [Ferramentas gráficas](#)

XML, [Ferramentas e Idiomas](#)

xorg.conf, [Configuração do servidor X](#)

xpdf, [Geral](#), [Exibindo arquivos pré-formatados](#)

## Y

yast, [Ferramentas](#)

yum, [Sistemas que usam pacotes RPM](#)