
Administrando o Linux
MEL COM AÇÚCAR!
WWW.TUX4.COM.BR

Sumário

Sobre o E-book

Este livro tem como objetivo capacitar o leitor na utilização e administração de uma distribuição GNU/Linux. O livro não aborda em particular nenhuma distro[1], porém, os conceitos tratados são abrangentes de modo que você poderá ter a liberdade de escolher qual delas utilizar. O livro também aborda conteúdo relacionado ao ***exame 101 da certificação LPI*** (Linux Professional Institute).

O conteúdo do livro foi enxugado de modo que ele vai direto ao ponto, sem rodeios. Esta compactação é fruto da minha experiência de mais de dez anos atuando como professor do treinamento de LPIC-1 e 2 e coordenando pedagogicamente cursos de Administração de sistemas em escolas particulares e Univesidades.

Referências:

Distro[1] – Abreviatura para Distribuição de GNU/Linux
Tux4you[2] – Rede social para o compartilhamento do conhecimento humano
www.tux4.com.br/networking

Sobre o autor por ele mesmo

Meu nome é Juliano Ramos, sou consultor de Servidores Linux há mais de dez anos, certificado LPIC-3, Novell Linux e Ubuntu. Atuei na prefeitura de São Paulo coordenando trinta unidades de Telecentros (Centros de inclusão digital) que eram implantados em bairros de baixo IDH. Após cinco anos à frente deste trabalho, segui para uma empresa privada atuando como coordenador pedagógico em projetos para o terceiro setor e hoje atuo em minha própria empresa como consultor de TI e professor. Sou ativista do movimento Open-Source e palestrante em eventos como o Campus Party, mas minha grande inspiração na vida é ser o mantenedor da mídia social Tux4you a qual sou também o criador.

Acesse:
www.tux4.com.br/networking

Créditos

A Deus, que se mostrou criador, que foi criativo. Seu fôlego de vida em mim
Me foi sustento e me deu coragem para questionar realidades
E propor sempre um novo mundo de
Possibilidades.

À minha amada esposa, por sua capacidade de acreditar
Em mim, mesmo quando ninguém mais o fez.

À minha pequena luz e fonte de inspiração
Obrigado pelo seu sorriso diário
Giovanna, minha filha.

Capítulo 1 – A origem do GNU/Linux

Neste capítulo:

- ✔ História do Multics – Unix – Minix – GNU e Linux
- ✔ Simbiose: Linux e GNU
- ✔ Software Livre
- ✔ Simbiose GNU e Linux

No ano de 1964 o MIT (Instituto de Tecnologia de Massachusetts) junto com a divisão de computadores da companhia General eletric (GE) e os laboratórios Bell de Telefonia iniciaram um projeto que estava muito adiante do seu tempo. Tratava-se da criação de um grande sistema operacional de "Tempo Compartilhado". Em um sistema de tempo compartilhado a capacidade e o processamento da máquina é dividido entre múltiplos usuários, que acessam através de terminais. O usuário dirige a sua tarefa, seus comandos são interpretados e executados em seguida (processamento online). Segundo a visão dos seus criadores, haveria imensos computadores, poderosos e inderrubáveis, rodando Multics, sendo acessados por milhares de pessoas em "terminais" espalhados por todo o planeta. Esses usuários pagariam suas "contas do computador" como hoje pagamos luz, água e TV a cabo. No ano de 1969 os laboratórios Bell abandonaram o projeto, sua divisão de computadores foi comprada pela Honeywell em 1970 que tentou usar de forma comercial o MULTICS sem sucesso. Com duras críticas o projeto ficou abandonado e teve sua última instalação operacional desligada em 31 de outubro de 2000. Apesar do insucesso deste projeto, o MULTICS foi o início de uma revolução no modo de se fazer e usar sistemas operacionais.

UNIX

Ken Thompson, trabalhava para a GE desenvolvendo o MULTICS. Quando a empresa abandonou o projeto em 1969 ele começou a reescrever o sistema num conceito menos ambicioso, batizando-o de Unics e usando a linguagem de montagem (assembly). Mas tarde, o sistema foi rebatizado de Unix pôr Brian Kernighan que foi pioneiro no desenvolvimento das linguagens de programação

AWK e AMPL.
Um marco importante no mundo UNIX aconteceu em 1973 quando todo o sistema foi portado de Assembly (linguagem de máquina) para a “Linguagem C” . A linguagem de programação C foi criada pôr Dennis Ritchie, sem sombra de dúvidas, grande parte do sucesso do Unix e seus derivados é por causa desta linguagem de programação. A linguagem C foi influência para as linguagens de programação recentes, tais como: C++,Java, C#, PHP e Javascript. Ritchie morreu em 12 de outubro de 2011, uma semana após a morte de Steve Jobs. Sua morte só foi noticiada em alguns portais especializados, bem diferente do que aconteceu com Jobs.

MINIX

“Uma opção livre, de código fonte aberto, para se aprender UNIX”.

Andrew S. Tanenbaum criou o sistema operacional MINIX para explicar o funcionamento dos sistemas operacionais. Tanenbaum lançou um livro em 1987 chamado "Operating System Design and Implementation" que continha 12.000 linhas de código do kernel, gerenciador de memória e sistemas de arquivos. Este sistema era compatível com a sétima edição do UNIX, porém, era de código aberto e disponível. Os cursos universitários de Tecnologia, começaram a usar o MINIX que rodava em computadores modestos; para explicar o funcionamento do sistema operacional UNIX, proprietário, caríssimo e que exigia um super computador.

GNU

“O início do conceito de Software Livre”

Richard Matthew Stallman, ou RMS nasceu em 16 de março de 1953, fundador do projeto GNU e da fundação do software livre é um dos ativistas mais atuantes em favor do software livre no mundo. Seu primeiro contato com um computador aconteceu no ano de 1969 no primeiro ano do ensino médio (High School), Stallman passou o verão escrevendo seu primeiro programa - um pré-processador para a linguagem de programação PL/1 no IBM 360. "Eu escrevi primeiro em PL/1, passando então para a linguagem de máquina (assembly) quando o programa PL/1 tornou-se grande demais para caber no computador". Contou Stallman, anos depois (Williams 2002, Capítulo 3).
Nos anos de 1980 aconteceu o “O declínio da cultura hacker” que até então dominava a vida de Stallman. A portabilidade dos softwares tornou-se um problema para os fabricantes de computadores, que começaram a não divulgar o código fonte dos seus softwares de modo que os concorrentes não poderiam utilizá-lo. Quando esta nova cultura de softwares proprietários atingiu o MIT onde RMS trabalhava, ele a rejeitou. Negou-se a assinar acordos de não-divulgação de informação. Ele escolheu, ao contrário, compartilhar seu trabalho com os outros, o que considerou como um espírito clássico de colaboração científica. No ano de 1984 Stallman parou seu trabalho no MIT para dedicar-se integralmente ao seu projeto GNU, anunciado em setembro de 1983.
No ano de 1985 RMS publicou o Manifesto GNU[1].

Referência:
http://www.gnu.org/gnu/manifesto.pt-br.html

GNU e o Kernel

O GNU é um sistema operacional que foi criado nos padrões UNIX porém com um código próprio que é livre para ser distribuído, usado, melhorado e copiado. Porém de 1985 quando foi lançado o manifesto GNU até 1992 o sistema operacional GNU não estava pronto para ser distribuído, faltava-lhe um kernel que fosse realmente utilizável.
Kernel (Núcleo) é o software que faz a gestão do Hardware do seu computador, ele serve de ponte entre os aplicativos e o processamento real de dados feito a nível de Hardware. O sistema GNU iniciou o desenvolvimento de um kernel chamado HURD em 1990 porém por um erro de desenvolvimento este projeto acabou ficando estagnado. De acordo com Thomas Bushnell, o arquiteto inicial do Hurd, o plano inicial era adaptar o kernel 4.4BSD-lite (E isto provavelmente daria certo de acordo com Bushnell) para o sistema GNU, porém, Stallman propôs que eles usassem o Mach microkernel que estava sendo desenvolvido na universidade Carnegie Mellon (CMU). Esta escolha atrasou o desenvolvimento em três anos, devido a incerteza se a CMU iria liberar o código do Mach em uma licença de acordo com o GNU. Neste intervalo de incertezas algo surpreendente aconteceu.

O Kernel Linux

O projeto GNU seguia firme com vários desenvolvedores abraçando a causa, foram criadas ferramentas indispensáveis para um sistema operacional, como um compilador para a linguagem C, editores de texto entre outras aplicações. Em 1992 o sistema já estava "quase" pronto, mas ainda faltava um kernel funcional. Neste ano um jovem Finlandês chamado Linus Torvalds, mudou a licença de um kernel que tinha desenvolvido a partir do Minix, de acordo com suas palavras (Era um Minix melhor que o Minix) para a licença livre GPL do projeto GNU. Então, não demorou muito para os desenvolvedores empacotarem o Kernel Linux, com as dezenas de softwares GNU já disponíveis criando um sistema operacional completo.

A Simbiose

Simbiose é uma relação mutualmente vantajosa entre dois ou mais organismos vivos de espécies diferentes. Explicar o que é "simbiose" foi a melhor maneira que encontrei para explicar o sucesso do GNU/Linux, um dependia do outro. Sem a licença livre do projeto GNU chamada de GPL (GNU General Public License) o Linux não teria alcançado seus primeiros desenvolvedores voluntários que tornaram o código utilizável em larga escala, além disto, para o usuário final, foram os softwares aplicativos (GNU) que tornaram o sistema operacional realmente operacional.

Free Software

A FSF - Free Software Foundation em português (Fundação do Software Livre) é uma organização sem fins lucrativos, fundada em 04 de outubro de 1985 por Richard Stallman e que se dedica a eliminação de restrições sobre a cópia, redistribuição, estudo e modificação de programas de computador. No ano de 1989 Richard Stallman lançou pela FSF a licença GPL (Gnu Public License) - Licença Pública Geral que garantia aos desenvolvedores do projeto GNU que um software livre seria sempre livre. Stallman, compreendeu, que seria muito difícil eliminar as leis de Copyright (direitos autorais) então, ele usou esta lei a favor de suas ideias, criando o conceito de Copyleft (Esquerdos autorais).
A GPL é uma licença com Copyleft, isto significa que ela é "viral", ou seja, quando um

programador usa um código GPL e altera/modifica este código ele também deve deixar estas modificações sobre GPL. Mas quais liberdades a GPL garante? São quatro liberdades básicas:

- A Liberdade para usar o software
- A Liberdade de estudar o software
- A Liberdade de copiar e compartilhar o software com outros
- A Liberdade para modificar o trabalho e também para distribuir os trabalhos modificados e derivados

Com a garantia destas liberdades, a GPL permite que os programas sejam Administrando o linuxdistribuídos e reaproveitados, mantendo, porém, os direitos do autor por forma a não permitir que essa informação seja usada de uma maneira que limite as liberdades originais. A licença não permite, por exemplo, que o código seja aponderado por outra pessoa, ou que sejam impostos sobre ele restrições que impeçam que seja distribuído da mesma maneira que foi adquirido. Com o passar dos anos, outras licenças livres foram desenvolvidas, algumas mais outras menos permissivas, a Fundação do software livre considera livre uma licença que atenda pelo menos as quatro liberdades. A GPL ganhou uma segunda versão em 1992 (GPLV2) e a uma terceira versão em junho de 2007 (GPLv3) que é a versão atual.

Distribuição GNU/Linux

Você pode usar o sistema operacional GNU com o kernel HURD (Não estável), BSD ou com o Linux (Mais comum e recomendado). Neste caso, é comum chamarmos de distribuição GNU/Linux. Uma distribuição GNU/Linux (distro) é um sistema operacional completo, isto inclui o núcleo (linux), um conjunto de pacotes (softwares), um sistema de gestão de pacotes e um repositório. Algumas distribuições foram criadas e são mantidas por pessoas e/ou comunidades de voluntários, outras porém, são mantidas por empresas. A primeira distribuição de GNU/linux foi lançada em 1992 chamada de MCC Interim Linux, em seguida surgiu a SLS (Softlanding Linux System) e ainda que não tenha durado muito, foi a base para a criação da distro Slackware (Que até hoje está em evidência). Em 1993 Ian Murdoch lançou o Debian Linux, e no ano seguinte vieram o Red Hat Linux e o SuSe Linux.

Capítulo 2 – Sobre as distribuições de GNU/Linux

Neste capítulo

- ✔ Conheça as principais distribuições de GNU/Linux
- ✔ Debian, Slackware, Fedora, Ubuntu, Arch Linux, Trisquel Linux

Debian

O Debian foi lançado em 16 de Agosto de 1993 por Ian Murdock, ao tempo estudante universitário, que escreveu o *Manifesto Debian* que apelava à criação de uma distribuição Linux a ser mantida de uma maneira aberta, segundo o espírito de Linux e do GNU. O projeto Debian cresceu vagarosamente até 1995 quando o projeto *dpkg* ganhou notoriedade.

O *dpkg* é a base de gerenciamento de pacotes da distribuição linux Debian. Foi inicialmente criado por Matt Welsh, Carl Streeter e Ian Murdock como uma aplicação *Perl*, sendo posteriormente

reescrito, em sua maior parte, para a linguagem C (Linguagem de programação) por Ian Jackson em 1993. O *dpkg* é um software para instalar, remover e atualizar programas. A primeira versão 1.x do Debian aconteceu em 1996.

O ciclo de desenvolvimento das versões do Debian passa três fases:

- "*Unstable*" - Instável
- "*Testing*" - Teste
- "*Stable*" - Estável

Quando as versões estão na fase "*testing*" elas são identificadas por codinomes tirados dos personagens do filme "Toy Story". Ao se tornarem "*Stable"* as versões recebem um número de versão (ex: 5.0).
A versão *"Testing"* atual é *"Stretch"*. A versão *"Unstable"* terá sempre o nome Sid, personagem que costumava a quebrar os brinquedos, em alusão aos *bugs* que podem ocorrer nesta versão.

Slackware Linux

O Slackware é um sistema operacional baseado em projetos oficiais de software livre, desenvolvido por pessoas espalhadas no mundo organizadas em comunidades e instituições, sendo a principal delas a FSF (Free Software Foundation) com seus projetos e licenciamentos GNU LGPL de software livre. Utiliza o Kernel oficial da Linux Foundation, Linux. O nome "Slackware" teve sugestiva origem da "*The Church Of the SubGenius"* (Igreja do Subgênio) que é uma organização religiosa originalmente baseada em Dallas, no estado estadunidense do Texas, a igreja do Subgênio ganhou destaque nas décadas de 1980 e 1990 e mantêm uma presença ativa na internet. A igreja ganhou espaço com a popularização da internet em 1990 com um site elaboradamente decorado e dois "*newgrous Usenet"* (alt.slack e alt.binaries.slack) e pelo programa de radio semanal (*Hour of slack*).

Patrick Volkerding criador do Slackware incorporou da Igreja do Subgênio este nome, de onde idealiza-se o termo "SLACK" que, satírica e ironicamente, incorpora-se o "Senso de liberdade, independência e originalidade para alcançar suas metas pessoais". Como a tradução literal de "SLACK" é "Preguiça" e "WARE" é "Produto" muitos associam o nome como "Produto preguiçoso" principalmente por que o Slackware não usa softwares de lançamento, somente versões estáveis, além disto, todas as configurações do sistema são feitas diretamente nos documentos de texto de configuração e não com utilitários gráficos.
O Slackware Linux (ou simplesmente "Slack") tem como objetivo manter-se fiel ao padrões UNIX, mantendo-se bem estruturada e organizada para administradores e usuários, profissionais e acadêmicos, rejeitando ferramentas de configuração que escondam o real funcionamento do sistema adotando o princípio KISS (Acrônimo em inglês de: *Keep It simple, stupid* – Faça isto simples, estúpido) de produção. A primeira versão do Slackware, o 1.0.0, foi lançada em 16 ou 17 de julho de 1993 por Patrick Volkerding, fundador e programador líder do projeto até os dias atuais.
O Slackware possui seu próprio gerenciador de pacotes, o pkgtool, e traz os comandos de gerenciamento: installpkg, upgradepkg, removepkg, explodepkg, makepkg; todos sem gerenciamento de dependências. O formato dos pacotes .tgz/.txz são bastante similar a um .tar.gz, contendo apenas os arquivos a serem instalados em suas respectivas pastas em relação à pasta raiz do sistema, além de um script com comandos complementares para a instalação.

Fedora

Fedora antigamente chamado de **Fedora Core** é um sistema operacional que tem por base o linux, a distribuição linux é completamente livre de custos para poder usufruir e partilhar. Foi criada pela Red Hat. Atualmente mantida pelo projeto Fedora (Fedora Project) e patrocinado pela Red Hat.

O gerenciador de pacotes do Fedora é o **RPM** (Red Hat Package Manager), o RPM serve para instalar, atualizar, desinstalar, verificar e procurar softwares. Até a versão 17 o instalador era semelhante ao da distribuição Red Hat 9, na versão 18 o sistema de instalação foi totalmente reformulado. Novas versões do Fedora são lançadas aproximadamente a cada seis meses. Além das versões oficiais, o Fedora possui também uma versão instável, o Rawhide, que serve como um campo de provas para todas as atualizações e mudanças que farão parte da próxima versão.

O Fedora distribui software absolutamente livre que tem um instalador gráfico completo, vem com ferramentas desktop e de administração fáceis de usar. Fedora é o nome de um clássico chapéu que surgiu na década de 20.

Ubuntu

Ubuntu é uma distribuição baseada no Debian patrocinada pela Canonical Ltd. O Ubuntu diferencia-se do Debian por ter versões lançadas semestralmente, por disponibilizar suporte técnico nos 9 meses seguintes ao lançamento de cada versão (as versões LTS – *Long Term Support* – Para desktop e servidor recebem 5 anos de suporte) e pela sua filosofia em torno de sua concepção. A proposta do Ubuntu é oferecer um sistema que qualquer pessoa possa utilizar sem dificuldades, independentemente de nacionalidade, nível de conhecimento ou limitações físicas. O sistema dever ser constituído principalmente por Software Livre. Deve também ser isento de qualquer taxa.

O nome “Ubuntu” - [u'buntu] deriva do conceito sul africano de mesmo nome, diretamente traduzido como “Humanidade com os outros” ou “sou o que sou pelo que nós somos”. O ubuntu assim como o Debian utiliza pacotes no formato .DEB que podem ser instalados pelo utilitário *dpkg* ou *apt.* Um grande diferencial do Ubuntu em relação as outras distribuições de GNU/Linux é a sua interface gráfica Unity. O Unity foi desenvolvido pela comunidade Ayatana e adaptado pela Canonical Ltd. Sua primeira aparição foi na versão 10.10 para netbooks, ele foi desenhado inicialmente para fazer um uso mais eficiente do espaço das telas limitadas dos netbooks, devido ao seu sucesso, tornou-se padrão na versão 11.04 que ainda incluía o GNOME como opção. Diferente do GNOME, KDE, XFCE e LXDE o Unity não inclui aplicações, já que foi feito para usar programas GTK+ já existentes. A partir da versão 11.10 do Ubuntu o Unity passou a ser a única interface padrão.

Opensuse

Após adquirir o Suse linux em janeiro de 2004, a Novell, uma empresa Americana que na década de 1980 ficou famosa por seu sistema operacional de rede (Netware) incentivou o desenvolvimento de uma comunidade para o desenvolvimento de uma distribuição GNU/Linux. A comunidade OpenSuse Project é patrocinada pela Novell e conta com voluntários de todo o mundo.

OpenSuse é completamente livre e disponível para download, eles também vendem um Dvd-Box

para o público em geral que deseja colaborar com o projeto. O OpenSuse possui edições para arquiteturas x86 e x86-64. O OpenSuse é compatível com o RPM (Red Hat Package) e possui seu próprio gerenciador de pacotes Zypper que funciona de forma similar ao apt-get do Debian.

Arch Linux

O Arch assim como o Slackware é uma distribuição KISS. O sistema é baseado em torno de arquivos binários, que são classificados como i686 e x86_64, sendo um sistema parecido com os ports/ebuild está disponivel para a compilação automática dos pacotes, trata-se do Arch Build Systema.

O Arch Linux é considerado uma das distribuições mais difíceis de se instalar, isto por quê, não existe um programa de instalação. Você vai realizando tarefas através de scripts que simplificam o processo. O gerenciamento de pacotes é feito pelo *pacman* que conduz a instalação, atualização e remoção e atualização de pacotes.

Trisquel Linux

O projeto nasceu em 2004 com o patrocínio da Universidade de Vigo, e foi oficialmente apresentado em Abril de 2005 com Richard Stallman, fundador do projeto GNU. O projeto hospeda seus próprios repositórios, que derivam do “main” e “universe” do Ubuntu, porém todo o software proprietário ou com licenças não livres são eliminados. O Trisquel está na lista de distribuições livres da FSF.

Capítulo 3 – Instalando o Linux

Neste capítulo:

- ✔ Passos sobre a instalação
- ✔ Analisando o Hardware
- ✔ Usando um Live-CD
- ✔ Instalando o linux pelo pendrive
- ✔ Particionando o Disco rígido

Quando você compra um computador, normalmente ele possui o Microsoft Windows pré-instalado. Se você pretende usar o Linux, deverá realizar a sua instalação. O processo de instalação de uma distribuição GNU/Linux costuma ser fácil. A parte que normalmente causa fobia nos novatos é o particionamento do disco, o resto da instalação é rotina, apenas uma questão de seguir instruções. Se você ainda não se sente confortável para instalar o Linux, tudo bem, você poderá realizar o download de uma distribuição Live-CD e executá-la sem precisar alterar nenhuma modificação no seu disco rígido. Todo o sistema neste caso funciona pelo CD ou DVD, quando você reinicia o computador e remove o CD do drive, todas as suas coisas estão lá, intactas.Porém, quando você executa o sistema pelo CD/DVD e até mesmo pelo pendrive, não terá o mesmo desempenho e prazer que é usar o linux nativamente em seu computador.

Passos da instalação do Linux

Instalar qualquer distribuição de Linux envolve uma série de passos. Vou orientá-lo sobre estas etapas de modo que você poderá instalar qualquer uma seguindo as orientações dos instaladores. Algumas distribuições de Linux exigem que você tenha um pouco de informação sobre o hardware do seu PC. Felizmente a grande maioria das distribuições instala de forma automatizada todos os seus periféricos, exemplo: Ubuntu Linux, OpenSuse, Fedora e Linux Mint.

1) O primeiro passo é você realizar o download de sua distribuição. No site oficial é comum o link para a seção download com diferentes arquiteturas. Escolha 32 Bits se seu computador tem menos de 4GB de memória RAM e um processador mais antigo, opte pela versão 64 Bits se seu processador é mais moderno e você possui 4GB ou mais de memória RAM.

É padrão que a distribuição seja disponibilizada no formato (ISO) – Imagem de disco. No Windows, clique com o botão direito do mouse sobre a imagem da distro que você realizou download e escolha a opção “gravador de imagens de disco do windows”.

2) O Segundo passo é configurar no Setup do seu PC a inicialização pelo CD/DVD. As etapas específicas para entrar na configuração do Setup e definir o dispositivo de boot varia de computador para computador. Normalmente você acessa o Setup com uma tecla (F2 ou Delete) após ligar o PC, movimentando-se com as setas do teclado, você localiza a opção (Boot) e coloca o CD/DVD em primeiro lugar da ordem de boot. Após isto é necessário salvar o Setup, fechar e iniciar o computador com o CD/DVD de sua distro no drive.

3) Caso sua distribuição seja Live, você poder escolher a opção de testar a distribuição. Neste caso, após reiniciar o computador remova o CD/DVD. Nenhuma alteração será realizada no seu disco rígido.

4) Se você pretende instalar o Linux, antes de qualquer coisa, faça um backup de seus arquivos (em outro HD, Disco Rígido externo, em algum HD virtual como o dropbox ou google drive). Se você pretende instalar junto com o Windows (Dual Boot) de modo que possa escolher qual usar na inicialização do computador. Instale primeiro o Windows e reserve um espaço (pelo menos 20GB) para o Linux. Instaladores como os do Ubuntu, OpenSuse, Fedora, já vão detectar este espaço e ofertá-lo para você instalar o sistema. Caso o instalador pergunte sobre em qual local instalar o gerenciador de boot (grub ou lilo) selecione MBR.

5) Seleção de softwares. A grande maioria das distribuições já instalam um conjunto de pacotes (softwares) de modo automático. Se você é iniciantes sempre escolha as opções (default) dos instaladores.

6) No final da instalação você vai reiniciar o seu computador. Lembre-se de remover o CD/DVD para não iniciar novamente o processo de instalação. Distribuições como o Fedora, OpenSuse e Open Mandriva, oferecem pacotes adicionais no primeiro boot.

Analisando o seu Hardware

Se você está preocupado se seu PC pode ou não rodar o Linux, aqui está uma lista das principais considerações antes de começar o processo de instalação.

- **Unidade de DVD**: Você deve ter uma unidade de DVD e que seu computador seja capaz de iniciar por esta unidade. O modelo exato não importa. Se você adicionar uma unidade de DVD externa a uma porta USB funcionará sem problemas no linux.
- **Disco Rígido**: Qualquer unidade de disco IDE/SCSI funciona no linux. Para usar as distribuições atuais confortavelmente, tenha pelo menos 20GB de espaço.
- **Teclado:** Todos os teclados vão funcionar no Linux.
- **Monitor:** O tipo de monitor não é particularmente crítico, exceto que ele deve ser capaz de exibir as resoluções de tela que a placa de video usa. Se o instalador não detectar o seu monitor você pode escolher um modelo genérico.
- **Mouse**: O programa de instalação pode detectar o mouse. Todos os tipos de mouse (como PS/2 ou USB) vão funcionar no ambiente gráfico do Linux (X Window System).
- **Placa de rede**: Normalmente o instalador irá detectar e tentar configurar (DHCP) sua placa de rede. Se você tiver problemas, tente encontrar na placa a marca e o modelo, procure nos buscadores informação de como proceder para instalar na sua distro especifica.
- **Placa de som:** Se o seu PC tiver uma placa de som, faça uma pesquisa com a marca e modelo na WEB para ter certeza se ela é compatível com o linux.
- **Placa de Vídeo**: Linux funciona bem com todas as placas de vídeo (Também conhecidos como adaptadores de vídeo), alguns modelos porém não possuem suporte 3D e você pode ter problemas com altas resoluções, destaco aqui, os adaptadores SYS que estão em Notebooks mais antigos, da linha Positivo. Se você usa um adaptador: Intel, Nvidia ou ATI fique tranquilo com o Linux.
- **Impressora**: Este é um hardware que você deve pesquisar marca e modelo para saber se é compatível com o Linux. Equipamentos que possuem um bom suporte (tratando-se de impressoras) são os fabricados pela HP.

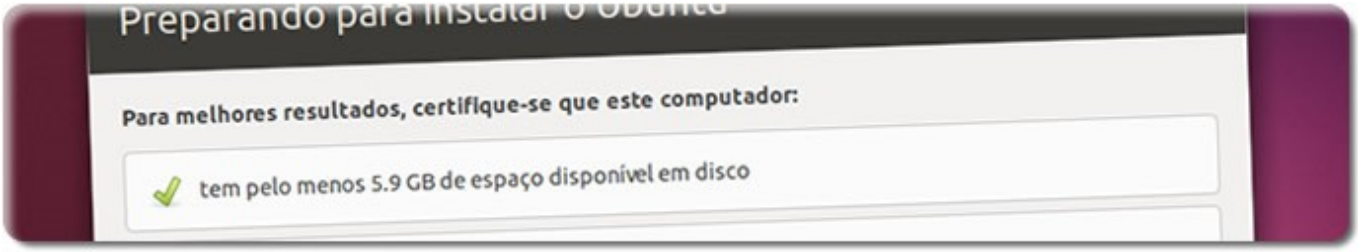
Você pode visualizar uma lista de Hardware compatível com o linux, neste endereço:
http://www.tldp.org/HOWTO/Hardware-HOWTO/

Instalação do Ubuntu Linux

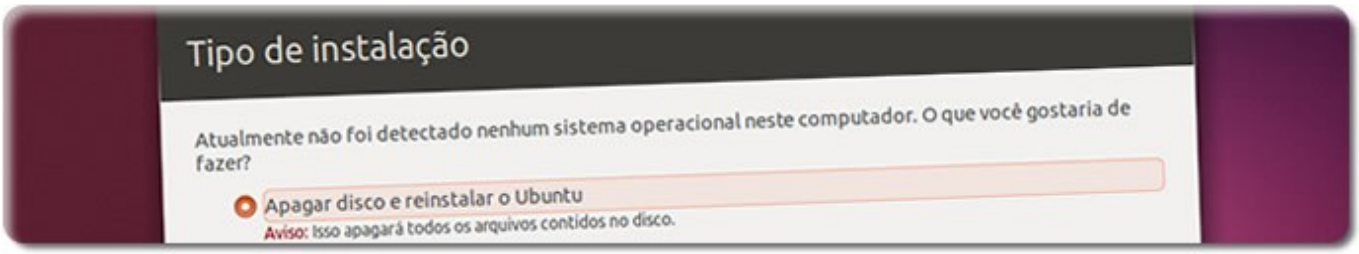
Este livro não aborda em particular nenhuma distribuição de Linux, no entanto, abordaremos o processo de instalação do Ubuntu Linux que considero o mais simples e fácil para usuários iniciantes. Ligue o computador, coloque o CD/DVD do Ubuntu. Deixe o computador prosseguir com a inicialização do instalador. Hoje em dia a grande maioria dos computadores novos já saem de fábrica com a configuração para dar boot pela unidade de CD/DVD-ROM, portanto, se o seu computador não estiver com esta configuração definida, você deverá entrar no setup da sua máquina para setar essa opção.

A figura abaixo mostra a tela de configuração da linguagem a ser utilizada. Vamos escolher a linguagem Português do Brasil e dar um clique no botão Instalar o Ubuntu:

Na tela que aparecer em seguida, Preparando para instalar Ubuntu, selecione a opção **Instalar esse programa de terceiro** e então clique em **Continuar.**



Caso seu computador não possua nenhum sistema operacional, a tela ***Tipo de Instalação***, irá aparecer conforme imagem abaixo. Dê um clique na primeira opção (Apagar disco e reinstalar Ubuntu) e depois um clique no botão **Instalar Agora**.



Caso você tenha algum sistema operacional, você terá a opção (Apagar o seu sistema operacional e instalar o Ubuntu).

Na próxima tela, **Onde você está?**, dê um clique na região do mapa onde você se encontra e clique no botão **Continuar:**

Na tela Disposição do teclado, veja que automaticamente o instalador já marca exatamente o tipo de layout do seu teclado. Se você estiver em dúvida se o layout do seu teclado foi configurado corretamente, basta você digitar algumas palavras acentuadas no campo "**Digite aqui para testar o seu teclado**". Se a configuração estiver incorreta, você poderá dar um clique no botão **Detectar a disposição do teclado**. Quando terminar e o teclado estiver corretamente configurado, clique em Continuar:

Agora nesta próxima tela (**Quem é você?**), você deverá fornecer algumas informações necessárias ao instalador. Veja no exemplo abaixo como fizemos o preenchimento e ao término clique em **Continuar:**

Ao término da cópia dos arquivos, clique no botão **reiniciar agora** lembre-se de remover o CD/DVD do drive.

Capítulo 4 – Noções fundamentais

Neste capítulo:

- ✔ Noções fundamentais sobre o Shell
- ✔ Histórico de comandos
- ✔ Gerenciamento de Arquivos
- ✔ Filtros de Texto

O Shell

O Shell é um interpretador de comandos que analisa o texto digitado na linha de comandos e os executa produzindo algum resultado. Se você está usando o modo gráfico, aperte as teclas: ctrl+alt+f1 para acessar o modo-texto.

Os principais tipos de Shell:
bash, sh, csh, tcsh, ksh e zsh, sendo o bash, o shell padrão na maioria das distribuições de Linux.

Prompt do Shell
O símbolo $ (dólar) identifica o prompt de comandos do shell. Algumas variações deste símbolo são permitidas, como o nome de usuário, do computador, diretório corrente entre outras opções.

Simbologia
O prompt do shell pode variar, dependendo do usuário que está utilizando o sistema no momento. O sinal “$” significa que um usuário comum é que está usando a máquina. O sinal “#” significa que o super-usuário está logado no sistema.

Root – Também conhecido como Super Usuário
O super-usuário é o administrador do sistema Linux. Ele tem poderes para fazer absolutamente tudo no sistema. Ele é conhecido como usuário “root” ou “raíz” traduzindo do inglês.

Variáveis Ambientais
Durante a execução do bash algumas variáveis são carregadas, elas também, podem ser configuradas manualmente.

Prompt Sting 1

A primeira variável que abordaremos é a *“prompt string 1”* ou simplesmente **PS1**, esta variável guarda o conteúdo do prompt de comandos do bash quando ele está pronto para receber comandos. O conteúdo de qualquer variável poderá ser visualizado utilizando o comando **echo** e o nome da variável:

echo $PS1

Geralmente o prompt é exibido da seguinte maneira:
juliano@trisquel:/home$

Que representa:
nomedeusuário@nomedocomputador:/diretório

Na variável é configurado da seguinte maneira:
\u@\h:\W\$

Vejamos a seguir uma tabela com os caracteres da **PS1:**

\]
Termina

\t
Exibe a hora

\h
Exibe o hostname (nome do computador)

\s
Exibe o shell

\u
Especifica o nome do usuário

\w
Especifica o diretório corrente

Exemplo de configuração da variável PS1, abra seu terminal de comandos e digite:

PS1=” \u@\h:\W:”

Prompt String 2

A variável PS2 armazena o conteúdo do prompt de comandos quando é necessário múltiplas linhas de comandos para completar um comando. Esta variável é o simbolo que aparece para o usuário quando um comando necessita seguir para uma segunda linha.

Variáveis Global e Local
Existem dois tipos de variáveis de ambiente: Global e local.

Variáveis de ambiente local:
São as variáveis disponíveis pelo shell corrente, e que não estão sendo acessadas por subprocessos do sistema. Para visualizar as variáveis de ambiente locais, execute os comandos env ou printenv. As variáveis globais podem ser acessadas por qualquer shell.

Comandos para manipular as variáveis de ambiente:

echo

Exibe o valor de uma variável de ambiente, exemplo:
echo $USER

Especificar um intervalo de tempo (Em segundos) antes de executar o próximo comando:

sleep

Exemplo de uso do comando sleep:
sleep 5

Para tornar global uma variavel de ambiente use o comando **export** como no exemplo:

VALOR=10
export VALOR

Executar e exportar comandos contidos em um arquivo no shell corrente:

alias

Exemplo do comando:

alias cor='ls –color'

No exemplo acima, criamos um comando chamado (cor) que ao ser utilizado, executa o comando **ls –color**. Se você desejar que este comando exista após reiniciar o seu computador deverá criar ou editar (caso já exista) o arquivo **.bashrc** localizado no seu diretório de usuário. Observe que na frente do nome do arquivo existe um ponto (.bashrc) no ambiente Unix, isto representa um arquivo oculto. Se você está em um terminal de comandos pelo modo gráfico, pode executar o seu editor de texto já abrindo o arquivo, neste caso, digite o comando:

gedit /home/seu_usuario/.bashrc

Se você não tem o editor gedit, tente com (kedit, xedit, pluma). Você também pode abrir um editor em modo-texto, o mais simples é o **nano** neste caso use:

nano /home/seu_usuario/.bashrc

Coloque no arquivo a linha:

alias cor='ls –color'

No caso do editor nano, salve o documento com as teclas: **<ctrl+o>** e saia do editor com as teclas: **<ctrl + x>**

Para excluir o valor de uma variável de ambiente, execute o comando **unset**.
Exemplo:

unset VALOR

Variáveis de ambiente do bash

Vejamos agora, algumas variáveis de grande utilização do shell bash:

MANPATH
Exibe os diretórios onde o comando man encontra suas páginas de manual.
echo $MANPATH

DISPLAY
Exibe o terminal do ambiente gráfico atualmente usado, exemplo:
echo $DISPLAY

HOME
Exibe o diretório home do usuário
echo $HOME

TERM
Exibe o terminal atualmente utilizado;
echo $TERM

LOGNAME
Exibe o login do usuário
echo $LOGNAME

USER
Exibe o nome de usuário
echo $USER

LANG
Exibe o idioma do sistema
echo $LANG

HISTSIZE
Define o valor máximo de comandos do history.
echo $HISTSIZE

OSTYPE
Exibe a arquitetura do sistema.
echo $OSTYPE

Histórico de Comandos

O Shell mantém um histórico dos últimos comandos digitados pelo usuário, podendo ser visualizado pelo caractere **!** De 4 formas diferentes:

!!
Visualizar o último comando digitados
!n
Onde **n** é o número que você deve especificar que está no histórico de comandos. A cada comando que você digita ele ficar armazenado em uma lista.

!string
Comandos que iniciam com o **string** que você especificar. (String: é uma sequência de caracteres, normalmente palavras).

!-n
Onde **n** é o numero que você deve especificar (Ele mostra a partir do último comando do histórico).

history

Exibe a lista de comandos digitados. Opções mais utilizadas:

history -r
Usa como histórico o arquivo (/home/usuario/.bash_history) ao invés de usar o histórico de comandos global.

history -w
Reescreve o arquivo (/home/usuario/.bash_history)

history -c
Limpa o histórico de comandos.

Aliases (Apelidos de comandos)

Usamos o comando **alias** para criar apelido que são como atalhos para comandos. O seu arquivo de configuração fica localizado no arquivo (**/home/usuario/.bashrc**).

Exemplo:

alias cor=`ls –color'

Para apagar um apelido de comando, usamos o comando **unalias:**

unalias cor

Entrando com comandos no shell

- Cada comando Unix/Linux possui uma sintaxe única e pode haver variações dependendo de sua distribuição de Linux.
- Alguns comandos podem requerer opções, geralmente precedidos pelo simbolo “-” ou “--” e por argumentos.
- O comando precisa ser válido e estar nos diretórios listados da variável PATH ou com sua localização absoluta.

Ao abrir o terminal de comandos digite **ls**, exemplo:

ls

Observe que ele não necessita de nenhum argumento ou opção para ser executado.
No caso deste comando, as opções podem ser configuradas separadamente ou combinadas:

ls -a
Mostra todos os arquivos do seu diretório incluindo os ocultos (que iniciam com ponto).

-t
Mostra os arquivos ordenados pela última data de modificação.

Para alguns comandos as opções têm de ser precedidas com dois traços “--” ao invés de um traço.

OBS: Alguns comandos oferecem formas alternativas de indicar uma mesma opção. No caso do “ls” as opções “-a” e “--all” produzem o mesmo efeito.

Man, Help e Info

Obviamente você não aprenderá ou decorará todas as opções dos comandos Unix/Linux de uma noite para o dia. Talvez nunca consiga e isto, não é realmente necessário. Para a salvação de nós meros mortais, temos os comandos: man, help e info.

Sintaxe do comando man:
man nome_do_comando

Sintaxe do comando info:
info nome_do_comando

Sintaxe do comando help
nome_do_comando –help

Gerenciamento de Arquivos

O **Filesystem Hierarchy Standard** (Padrão para sistemas de arquivos hierárquicos), ou FHS, define os principais diretórios, e o seu conteúdo, em um sistema operacional Linux ou do tipo Unix. A versão atual é a 2.3, anunciada em 29 de janeiro de 2004. O FHS é mantido pelo Free Standards Group, uma organização sem fins lucrativos formada por importantes empresas de hardware e software, como HP, Red Hat, IBM e Dell. Ainda hoje, a grande maioria das distribuições de GNU/Linux, incluindo membros da Free Standards Group, não adotam fielmente o padrão proposto. Em particular, diretórios (paths) criados pela FHS, como o /srv/, não foram adotados em grande escala. Alguns sistemas Unix e Linux rompem com o padrão FHS, como o GoboLinux. O Mac OS x utiliza uma estrutura com nomes legíveis para humanos em conjunto com um sistema baseado em FHS.

Para visualizar toda a estrutura de arquivos de sua distribuição

/bin/
Comandos binários essenciais para todos os usuários.

/boot/
Arquivos do Boot Loader

/dev/
Dispositivos, exemplo: /dev/null

/etc/
Arquivos de configuração específicos do computador.

/etc/opt/
Arquivos de configuração para o /opt/.

/etc/X11/
Arquivos de configuração para o X Window Systema, Versão 1.1

/etc/sgml/
Arquivos de configuração para SGML

/etc/xml/
Arquivos de configuração para XML

/home/
Diretórios de usuários.(Opcional)

/lib/
Diretório com as bibliotecas essenciais para os arquivos binários contidos nos diretórios /bin/ e /sbin/.

/mnt/
Sistemas de arquivos “montados” temporariamente.

/media/
Pontos de "montagem" para mídia removível, como CD-ROMs (surgiram na versão 2.3 do FHS).

/opt/
Pacotes estáticos de aplicações.

/proc/
Sistemas de arquivo virtual, que possui o estado do núcleo e processos do sistema; a maioria dos arquivos é baseada no formato texto (ex: tempo de execução, rede).

/root/
Diretório home para o super usuário (root).(Opcional)

/sbin/
Arquivos binários para propósito de administração do sistema.

/tmp/
Arquivos temporários. (Ver também /var/tmp).

/srv/
Dados específicos que são servidos pelo sistema.

/usr/
Hierarquia secundária para dados compartilhados de usuários, cujo acesso é restrito apenas para leitura.

/usr/bin/
O mesmo que a hierarquia do topo (/bin), mas contém apenas arquivos não essenciais (que não são necessários para que o sistema funcione ou para a sua recuperação).

/usr/include/
Diretório padrão para arquivos do tipo header.

/usr/lib/
O mesmo que a hierarquia do topo (/lib).

/usr/sbin/
O mesmo que a hierarquia do topo, mas contém apenas arquivos não essenciais (ex: daemon e serviços de rede)

/usr/share/
Dados compartilhados que são independentes da arquitetura do computador..

/usr/src/
Código fonte (ex: código fonte do núcleo do sistema)

/usr/X11R6/
X Window System, Versão 11 R6.

/usr/local/
Hierarquia terciária com dados locais, específicos deste host.

/var/
Arquivos "variáveis", como logs, base de dados, páginas Web e arquivos de e-mail.

/var/lock/
Arquivos de lock. Utilizados para manter o controle sobre recursos em uso.

/var/log/
Arquivos para log. Utilizado para log de dados em geral.

/var/mail/
Caixas de email dos usuários do sistema.

/var/run/
Contém informação sobre a execução do sistema desde a sua última inicialização. (ex: usuários e daemons em execução).

/var/spool/
Spool para tarefas em espera para execução. (ex: filas de impressão e emails ainda não lidos).

/var/spool/mail/
Local para caixas de correio dos usuários. Não deve ser mais utilizada, existe apenas para compatibilidade retroativa.

/var/tmp/
Arquivos temporários. Quando em modo multi-usuário, preferível em relação ao /tmp.

Comandos de gerenciamento de arquivos

O comando cp copia arquivos e diretórios. Ele pode ser utilizado para copiar apenas um arquivo ou múltiplos arquivos.

Exemplo:

cp arquivo1 arquivo2

No exemplo acima, copiamos o **arquivo1** para **arquivo2** criando este arquivo.

cp arquivo1 /home/juliano/documentos/arquivo2

No exemplo acima, copiar o **aquivo1** para um diretório específico (/home/juliano/documentos)

As opções mais frequentes do comando **cp,** são:

cp -d
Preserva os links ao copiar os arquivos

cp -p
Preserva todas as informações dos atributos do arquivo, como dono do arquivo, grupo, permissões e data

cp -R
Copia arquivos recursivamente, ou seja, copia diretórios com conteúdo.

cp -f
Força uma cópia, sobrescrevendo se necessários

cp -v
Mostra o nome de cada arquivo copiado.

Exemplo da copia de um diretório completo:

cp -R /home/juliano /home/backup

No exemplo acima, todo o conteúdo do diretório (/home/juliano) irá para (/home/backup).

Comando: mv

O **mv** move ou renomeia arquivos e diretórios. Ele não altera os atributos dos arquivos ou diretórios movidos se a transferência for o mesmo sistema de arquivos. Se o destino para onde os arquivos ou diretórios forem movidos não existir, o comando renomeia a origem, senão os dados serão gravados por cima.

No exemplo abaixo, renomeia-se o arquivo1 para arquivo2:
mv arquivo1 arquivo2

No exemplo abaixo, move-se o arquivo1 para o diretório /tmp.
mv arquivo1 /tmp

Comando: rm

O comando **rm** é utilizado para remover arquivos. Você só pode remover arquivos que você tenha permissão de gravação. O super usuário (root) pode remover arquivos e diretórios dos outros usuários do sistema.

As opções mais utilizadas:

- **-f** Força a remoção dos arquivos sem perguntar ao usuários
- **-R** Remove um diretório e todo o seu conteúdo

Exemplos, removendo o arquivo **texto.txt**:
rm texto.txt

Para remover um diretório com conteúdo:
rm -R diretório

Exemplo:
rm -R /opt/agenda

No exemplo acima, o diretório (agenda) será removido.

Você também pode forçar a remoção com o comando:
rm -Rf diretório

Comando mkdir

O comando **mkdir** é utilizado para criar um diretório. Você precisa ter permissão de escrita no diretório corrente para executar o mkdir, exemplo:

mkdir musicas

No exemplo acima, foi criado o diretório “musicas” no diretório corrente. Para criar um diretório especificando sua localização, faça:

mkdir /home/juliano/documento/administrativo

No exemplo acima, foi criado o diretório **administrativo** dentro do diretório /home/juliano/documento/

As opções mais frequentes, são:

mkdir -p

Cria o diretório especificado, mesmo que o diretório “pai” não existe. Neste caso ele cria também o diretório “pai”. Exemplo:

mkdir -p correios/sedex

No exemplo acima foi criado o diretório **correio** que chamamos de pai e o subdiretório **sedex**, caso você apague o diretório pai “correio” você também irá apagar seus subdiretórios (filhos), neste caso “sedex”.

Comando rmdir

O comando **rmdir,** remove um ou mais diretórios do sistema. O diretório precisa estar vazio.

Exemplo:
rmdir /home/documentos

Remove o diretório documentos dentro do diretório home.

Para remover diretórios com conteúdo, use o comando **rm**, exemplo: **rm -R /home/juliano/documentos**

Comando: touch

O comando **touch** muda a data e hora de acesso e/ou modificação dos arquivos.
As opções mais utilizadas são:

touch -a arquivo
Muda a data e hora para sua hora atual.

touch -m arquivo
Muda somente a data e hora de modificação para a atual

touch -t 201503112100 arquivo
No exemplo acima, estou configurando a data de criação do arquivo para: ANO (2015), MÊS (03), DIA (11), HORA (21:00). Neste caso: 2015 03 11 2100.

Filtros de Texto

Veremos agora alguns comandos de filtro de texto que estão disponíveis em praticamente todas as distribuições de GNU/Linux.

O comando cat

Usando o comando **cat** para criar arquivos de texto puro:

cat > texto.txt

No exemplo acima você irá digitar conteúdo de texto que será armazenado no arquivo (texto.txt). Ao final da sua digitação, pressione as teclas **ctrl + d** em uma linha vazia. Isto irá salvar o arquivo (texto.txt).

Para visualizar um arquivo de texto, use o comando:

cat texto.txt

Para concatenar arquivos, execute o comando:

cat texto1 > texto2

No exemplo acima o conteúdo de **texto2** é substituído pelo conteúdo de **texto1.** Para acrescentar conteúdo sem substituir o conteúdo já existente utilize >> ao invés de >, como no exemplo:

cat texto1 >> texto2

Neste exemplo o conteúdo de **texto1** é adicionado ao texto já existente no arquivo **texto2.**

O comando cut

O comando cut (cortar) lê o conteúdo de um ou mais arquivo recortando verticalmente este texto.

Exemplo:
Crie em texto chamado (planilha) com o seguinte conteúdo:

coluna1, coluna2, coluna3
10, 2, 8
12,9, 10
15,30,25
90,120,245

Observe que o delimitador de cada (coluna) é o sinal de vírgula. Em alguns caso poderia ser (:) dois pontos ou (;) ponto e vírgula. Para visualizarmos somente o conteúdo da segunda coluna do arquivo planilha executamos o comando:

cut -d, -f2 planilha

Entendendo o comando acima:
-d delimitador.

Configura o delimitador que separa uma coluna de outra. O padrão é “tab”, no caso do exemplo

(planilha) usamos (virgula).

-f Número.
Defina o número da coluna que será exibida, neste caso foi o 2 (segunda coluna).

O comando expand

O comando expand troca o tab (tabulação) dentro do texto para um número de espaços determinado pelo usuário. Exemplo:

Crie um arquivo chamado **tabulacao.txt** com o conteúdo abaixo:

Fusca azul
Gol Vermelho
Santana Preto

No exemplo acima utilizei três vezes a tecla <tab>. Visualize primeiramente o comando com o cat.

cat tabulacao.txt

Agora execute o comando **expand** especificando o número de espaços que substituirão a tabulação, exemplo:

expand -t1 tabulacao.txt

Tente, -t2, -t3, etc...

Comando: fmt

O comando **fmt** formata um texto com uma largura específica. Ele pode remover espaços ou adicionar espaços conforme a largura desejada, o padrão é 75 caracteres.

A opção frequentemente utilizada é:

fmt -w 50 leiame.txt

Neste exemplo, definimos 50 como largura desejada para o arquivo (leiame.txt).

Comando head

O comando **head** mostra as primeiras 10 linhas do inicio do texto. A opção frequentemente utilizada é:

head -n 50 leiame.txt

Neste exemplo as 50 primeiras linhas do arquivo (leiame.txt) serão mostradas.

Comando: Join

O comando **join** une as linhas de dois arquivos que possuam índice comum, exemplo:

Crie o arquivo **fruta.txt** e coloque o seguinte conteúdo:

1 maça
2 uva
3 melancia

Agora, crie o arquivo **preco.txt** e coloque o seguinte conteúdo:

1 R$ 2,00 Reais
2 R$ 4,00 Reais
3 R$ 5,00 Reais

Execute agora o comando **join** para concatenar os dois arquivos:

join fruta.txt preco.txt

O resultado será:

1 maça R$ 2,00 Reais
2 uva R$ 4,00 Reais
3 melancia R$ 5,00 Reais

O comando pr

Prepara um arquivo de texto para impressão. Exemplo:

pr -l 70 -o 30 fruta.txt

Onde:

-l
Especifica o número de caracteres de largura de página. O padrão é 66 caracteres.

-o
Especifica o número de espaços de margem esquerda.

O comando tac

O comando **tac** é o inverso do comando cat, ele mostra o conteúdo de um arquivo de texto de trás para frente. Exemplo: **tac arquivo.txt**

O comando tail

O comando **tail** visualiza as 10 últimas linhas de um arquivo. Funciona como o oposto do comando head.
As duas opções mais utilizadas do comando tail são:

-*n*
Especifica o número de linhas finais que o tail irá mostrar de um arquivo

-f
Mostra as últimas linhas de um arquivo continuamente. Útil para arquivo de LOG.

Exemplo:
tail -n 50 /var/log/meulog
tail -f /var/log/meulog

O comando wc

O comando wc conta as linhas, palavras e caracteres de um arquivo. Exemplo:

wc teste.txt

O comando xargs

O comando xargs utiliza a saída padrão de um comando como argumento para outro comando, exemplo:

cat texto.txt | xargs echo

Neste exemplo, cada linha do arquivo **texto.txt** é passada como argumento para o comando **echo**.

Capítulo 5 – Editores de Texto em modo-texto

Neste capítulo:
- ✔ Editor de texto nano
- ✔ Editor de texto mcedit
- ✔ Editor de texto VI e VIM

Editor nano

O Editor nano é um editor de texto puro muito utilizado no Unix e Linux. Para instalar no Debian e similares | Red Hat e similares use os comandos:

Debian/Ubuntu
apt-get install nano

Red Hat/Fedora:
yum install nano

Para abrir um arquivo utilize:
nano nome_do_arquivo

Neste editor você realiza as opções como Salvar, Copiar, Cortar e Colar através de teclas de atalhos. Vejamos:

ctrl+g
Exibe a ajuda

ctrl+b
Move o cursor um caractere atrás

ctrl+n
Move o cursor uma linha abaixo

ctrl+f
Move o cursor um caractere à frente

ctrl+p
Move uma linha acima

ctrl+d
Recorta o caractere

ctrl+k
Recorta a linha

ctrl+c
Exibe a posição do cursor

ctrl+A
Vai para o inicio da linha

ctrl+E
Vai para o fim da linha

ctrl+j
Justifica o parágrafo atual

ctrl+V
Avança uma tela

ctrl+y
Retrocede uma tela

ctrl+u
Cola texto ou desfaz justificação

ctrl+w
Localiza string do texto

ctrl+t
Verifica ortográfia

ctrl+r
Abre um arquivo

ctrl+o
Salva o arquivo com outro nome

ctrl+x
Sai

Editor mcedit

O editor mcedit ou mc é bastante fácil de ser utilizado. Inclusive com o uso do mouse mesmo no modo-texto. Para instalar no Debian e derivados | Red Hat e derivados, execute:

Debian e derivados:
apt-get install mc

Red Hat e derivados:
yum install mc

Abrindo um arquivo:
mcedit ou mc -e nome_do_arquivo

A seguir uma lista com as teclas de atalho do mcedit:

F1
Exibe a ajuda

F2
Salva o arquivo corrente

F3
Marca o inicio do bloco

F4
Substitui texto

F5

Copia o bloco marcado com f3

F6
Move bloco marcado com f3

F7
Procure texto

F8
Apaga linha corrente

F9
Abre o menu

F10
Sai

F12
Salvar como

Editor de texto VI e VIM

O editor VI tanto no UNIX como no Linux serve para criar arquivos de texto e scripts shells, assim como editá-los. O editor VI possui dois modos: Edição, quando você aperta a tecla (I) ou (insert). E o modo de comando, quando você aperta a tecla (ESC).

Iniciando o VI
Digite no bash:
vi ou **vi nome_de_arquivo** aperte **i** para escrever e **esc** para sair e executar os comandos:

:q (Enter)
Sair do arquivo sem salvar

:q! (Enter)
Sair sem salvar, forçando

:wq (Enter)
Salva o arquivo atual e sai do vi

:w (Enter)
Salva o arquivo atual

:w arquivo1 (Enter)
Salva o arquivo atual como arquivo1

:e arquivo1 (Enter)
Abre o arquivo1

:r arquivo1 (Enter)

Insere o arquivo no ponto onde está o cursor.

:u (Enter)
Desfaz a última ação, similiar a ctrl+z

:d ou dd (Enter)
Apaga a linha corrente

:yy
Copia a linha onde está o cursor

:p
Cola o texto

:dd
Apaga a linha onde está o cursor

:i
Inserir um texto

:s/velho/novo
Substitui a palavra velho pela palavra novo.

% s/velho/novo/g
substitua em todo o arquivo (%), todas (g) as ocorrências de velho por novo

Além dos comandos de edição do texto como já vimos, existem comandos para configurar o Editor VIM, são eles:

:set aw
Salva a cada alteração

:set nu
Mostra o número das linhas

:set ff
Converte o arquivo para DOS

:set et
Troca tab por espaços

Capítulo 6 – Permissões de Arquivos

Neste capítulo:

- ✔ Visualizando permissões
- ✔ Configurando permissões em arquivos e diretórios

Permissões de acesso a arquivos e diretórios
Neste capítulo estudaremos o sistema de permissões do linux. O sistema de permissões permite ao administrador do sistema (root) definir o nível de acesso ao usuários, grupos e outros arquivos e diretórios além de programas executáveis.

As permissões de acesso protege o sistema de algumas invasões e respectivos programas não autorizados. Por exemplo, no linux, podemos impedir que algum programa malicioso instale-se na máquina, delete algum arquivo ou que seja transferido para outras pessoas via rede, de modo a invadir outros sistemas.

3 níveis de permissão
O linux gerencia o sistema de privilégios em 3 tipos:

- 1. Privilégio do dono
- 2. Privilégio do Grupo
- 3. Privilégio de outros

Cada um destes tipos são divididos em 3 níveis de permissões de acesso:

- 1. Permissão de leitura (r)
- 2. Permissão de escrita (w)
- 3. Permissão de execução (x)

As definições de propriedade do usuário e do grupo fazem parte do **inode** e ambas são atribuídas quando um arquivo é criado. Geralmente, o proprietário é o usuário que criou o arquivo. O grupo do arquivo normalmente é definido como padrão do seu criador. A propriedade de grupo adiciona flexibilidade em situações nas quais uma equipe compartilha arquivos. Os “outros” usuários são aqueles que não são membros do grupo do arquivo e também não são os proprietários. Para cada uma dessas três classes de usuários, o modo de acesso define três tipos de permissões, que se aplicam diferentemente para arquivos e diretórios. As permissões encontram-se listadas na lista abaixo:

Examinar o conteúdo de um arquivo e poder listá-lo:
r

Escrever ou modificar um arquivo. Criar ou remover arquivos de um diretório:
w

Executar um arquivo como um programa. Acessar o diretório.
x

Essas três permissões se aplicam as três classes diferentes de usuários: **usuários, grupo e outros**. Cada uma tem permissão de leitura, escrita e execução.

Visualizando privilégios e permissões

Você pode visualizar as permissões de arquivos e diretórios listando-os com o comando:

ls -l /home/usuario
(Visualiza privilégios e permissões de arquivos)

ls -ld /home/usuario
(Visualiza privilégios e permissões de diretórios)

Ao executar o comando **ls -l** você terá como retorno, algo semelhante a :

drwx------ ... 2 wester ..... 512 Jan ... 29 23:30 .. Arquivos/
-rw-rw-r-- ... 1 wester ..... 280232 Dec .. 16 22:41... notas.txt

O primeiro caractere das linhas acima indicam o tipo de arquivo:

d	Diretório
b	Arquivo de bloco
c	Arquivo especial de caractere
p	Canal
s	Socket
-	Arquivo “normal”.

Repare agora que no restante da string ainda há 9 caracteres. Que são os 3 níveis de permissões (Dono, grupo e outros). usando o exemplo da segunda linha, arquivo (**notas.txt**):

rw-	Permissões do proprietários
rw-	Permissão do grupo
r--	Permissão para outros

Lista de permissões comuns:

Nenhuma permissão

r--
Permissão de leitura

r-x
Leitura e execução

rw-
Leitura e gravação

rwx
Leitura, gravação e execução

Para configurar um arquivo usando as permissões do modo textual (que vimos até o momento) usamos o comando:

chmod u+rw, g+w, o-rwx arquivo.txt

Onde:

U – Representa o usuário
G – Representa o grupo
O – Representa outros

Permissões em sistema Octal

Além de podemos configurar as permissões do modo textual. Podemos usar o modo octal, neste caso:

r = 4
w = 2
x = 1

Se você somar os 3 números, vai obter 7, que significa permissão total, ou seja, **rwx.** Exemplo:

chmod 764 arquivo.txt

Os respectivos valores são:

rwx = 7 (Dono – Acesso total)
rw- = 6 (Grupo – Leitura e Gravação)
r-- = 4 (Outros – Acesso a leitura)

Capítulo 7 – Usuários e Grupos

Neste capítulo:

- ✔ Gerenciamento de usuários
- ✔ Gerenciamento de grupos
- ✔ Adicionando e Removendo usuários
- ✔ Arquivos de configuração dos usuários

Para administrar usuários no Gnu/Linux é necessário conhecer os comandos *useradd* e *adduser*, a diferença entre eles é que o *useradd* é um binário e o *adduser* é um script.

Você pode verificar esta informação executando o comando:

file /usr/sbin/useradd

O comando “file” verifica o tipo do arquivo.

Qual dos comandos eu devo utilizar?

Para personalizar uma conta de usuário no momento de sua criação o comando ideal é o *useradd* pois nele é possível passar várias opções de parâmetros no momento da criação. O comando *adduser* é um script em Perl que utiliza a base do *useradd*, com ele a criação é automatizada, apenas respondendo-se perguntas como: Nome, Senha, Telefone, E-mail e etc…

Criando um novo usuário com useradd

useradd -m -c “Juliano Ramos” -s /bin/bash juliano

Opções utilizadas no comando de exemplo:

-m Cria o diretório home do usuário no momento da criação (/home/usuario) o diretório é criado com o mesmo nome do usuário.

-c Define o comentário, como o nome completo do usuário, por exemplo.

-s Define o shell padrão deste usuário, no exemplo: (/bin/bash)

O comando *useradd* cria o usuário, ele não define sua senha. Para isto, você deve executar o comando *passwd*. Exemplo:

passwd juliano

Criando um novo usuário com adduser

O utilitário de criação de usuários *adduser* é mais comulmente utilizado, pelo fato de por padrão já criar o diretório /home e definir a senha do usuário.

adduser juliano

Para verficar informações sobre o usuário, você poderá usar o comando *finger*, em muitos casos deverá realizar sua instalação. No caso do Debian e derivados:

apt-get install finger

Adicionando novos grupos

Para adicionar novos grupos no linux, utilize os comandos *groupadd* e *addgroup*, a diferença entre eles é que *groupadd* é um binário e *addgroup* um link simbólico para o script *adduser.*

Lembrando que você pode verificar isto com o comando **file.**

file /usr/bin/groupadd

Criar um novo grupo com addgroup

O comando para a criação de novos grupos é:

addgroup nome_do_grupo

Agora que temos usuários e grupos em nosso sistema, vamos gerenciá-los através de alguns comandos. É possível por exemplo, bloquear a conta de um usuário e forçá-lo a trocar sua senha em seu primeiro login, para isto usamos o comando **chage**:

chage -d 0 juliano

Para trocar a senha de um usuário usamos o comando:

passwd juliano

Para bloquear o login de um usuário, usamos:

usermod -L juliano

Para desbloquear o login deste usuário, usamos:

usermod -U juliano

Para modificar a descrição de um usuário, usamos:

usermod -c “Juliano Oliveira” juliano

Mudando o login de um usuário

O usuário “Juliano” agora deverá ser “Grubelilo”. O primeiro passo é mudar o nome do usuário:

usermod -l grubelilo juliano

Agora que já mudamos o nome do login, vamos alterar o seu diretório /home, primeiro vamos criá-lo:

mkdir /home/grubelilo

Adicione agora um novo grupo com o nome do usuário:

addgroup grubelilo

Mude o dono do grupo para o novo diretório:

chown grubelilo.grubelilo /home/grubelilo

Altere o grupo principal do usuário:

usermod -g grubelilo grubelilo

Altere o diretório padrão do usuário:

usermod -d /home/grubelilo grubelilo

Gerenciando grupos do sistema

Agora que já adicionamos novos grupos e usuários ao sistema, vamos aprender como gerenciá-los através de comandos. É possível por exemplo, alterar o grupo principal de um usuário, tornar um usuário administrador de um grupo entre outras opções.

Adicionando um usuário ao grupo
gpasswd -a aluno tux4you

Removendo um usuário de um grupo
gpasswd -d aluno tux4you

Tornando um usuário administrador do grupo
gpasswd -A aluno tux4you

Quando um usuário se torna administrador de um grupo ele tem permissão para adicionar e remover usuários do grupo, adicionar e remover senhas do grupo.

Adicionando senha ao grupo com usuário comum:
gpasswd logistica

Trabalhando com arquivos de usuários e grupos

Até aqui realizamos as configurações dos usuários e grupos através de comandos, mas, é possível o gerenciamento usando arquivos do sistema.

Bloqueio de usuários
Para bloquear o login de um usuário, acesse o arquivo: /etc/passwd, adicione o sinal de **!** antes do nome do usuário:

nano /etc/passwd

!aluno:x:1004:1007:Aluno Tux;;;:/home/aluno:/bin/bash

Login sem senha:
Abra o arquivo **/etc/passwd** e apague o **x** entre os dois : (dois ponto) ao lado do nome do usuário, exemplo da linha sem o **x:**

juliano::1004:1007:Juliano Ramos;;;:/home/juliano:/bin/bash

Sem dúvida está não é uma boa prática de segurança. Para voltar a digitar senha, coloque novamente o “x” entre os dois pontos.

Esqueceu a senha do root?

No exemplo anterior observamos como posso remover a senha de um usuário pelo arquivo (/etc/passwd). Se você não consegue mais acessar o seu sistema como root, também não poderá alterar este arquivo, neste caso. Use um LIVE-CD como o do Ubuntu e inicie o sistema.

Abra o seu HD pelo gerenciador de arquivos do Ubuntu e navegue até o diretório **/etc** abra agora o arquivo **shadow** e remova o x que está entre os dois : (dois pontos) do lado do nome root. Reinicie o computador e remova o live-cd. Seu root vai estar sem senha.

Capítulo 8 – Gerenciamento de pacotes

Neste capítulo:

- ✔ Instalação de pacotes Debian
- ✔ Instalação de pacotes Red Hat
- ✔ Instalação de pacotes Slackware
- ✔ Gerenciamento de pacotes pelo código fonte

O dpkg é o comando básico para lidar com pacotes Debian no sistema. Se você tem pacotes .deb, é com o dpkg que você instala ou analisa seu conteúdo. Mas este programa tem apenas uma visão parcial do universo Debian: ele sabe o que está instalado no sistema, e o que for dado na linha de comando, mas não sabe nada dos outros pacotes disponíveis. Assim, ele vai falhar se uma dependência não for satisfeita. Ferramentas como o apt-get, ao contrário, criará uma lista de dependências para instalar tudo o mais automaticamente possível.

Instalando pacotes
dpkg -i nomedopacote.deb

Remoção de pacotes
dpkg -r nomedopacote.deb

Arquivos de Log do dpkg
Dpkg mantém um log de todas as suas ações em **/var/log/dpkg.log**. Este log é extremamente detalhado, pois detalha todos os passos da manipulação de pacotes pelo dpkg.

Instalando pacotes pelo apt-get

APT é um projeto amplo, cujos planos originais incluem uma interface gráfica. Ele é baseado numa biblioteca que contém as aplicações principais, e o apt-get é o primeiro front-end em linha de comando, que foi desenvolvido dentro do projeto.

Instalando um pacote
apt-get install gnome-shell

Instalando um pacote pelo aptitude
aptitude install gnome-shell

Removendo um pacote
aptitude remove gnome-shell
apt-get remove gnome-shell

Quando você remove um programa, nem sempre, remove os arquivos de configuração. Exemplo:

Você instala o servidor de impressão CUPS e cria uma configuração personalizada. Ao remover com "**apt-get remove cups**" estas configurações ainda continuam presentes no **/etc/**, caso você venha a reinstalar este pacote estas configurações serão mantidas. Para remover um pacote e também estes
arquivos de configuração, utilize o comando:

Removendo com as configurações
apt-get purge gnome-shell

Instalando vários programas ao mesmo tempo
apt-get install pacote1 pacote2 pacote3

Removendo vários programas ao mesmo tempo
apt-get remove pacote1 pacote2 pacote3

Reinstalando pacotes

O sistema pode, às vezes, ser danificado com a remoção ou modificação de arquivos num pacote. A forma mais fácil de recuperar estes arquivos é reinstalar o pacote afetado. Infelizmente, o sistema de empacotamento nota que o pacote está instalado e se recusa educadamente a reinstalá-lo; para evitar isto, use a opção: --reinstall do comando apt-get. O comando abaixo reinstala o postifix mesmo quando ele já está presente:

apt-get –reinstall install gnome-shell

O cache dos arquivos .deb
Apt mantém uma cópia de cada arquivo .deb no diretório /**var/cache/apt/archives** para remover você pode usar o comando:

apt-get clean

Atualizando o sistema
apt-get upgrade

Pesquisando pacotes
apt-cache search nomedopacote

Removendo pacotes (Automático)
apt-get autoremove

Quando você usa o aptitude ou o apt-get ele procura os softwares nos servidores especificados no arquivo /etc/apt/sources.list. Vejamos um exemplo deste arquivo (ubuntu 15.04):

deb cdrom:[Ubuntu-MATE 15.04 _Vivid Vervet_ - Release amd64 (20150422.1)]/ vivid main multiverse restricted universe

See http://help.ubuntu.com/community/UpgradeNotes for how to upgrade to
newer versions of the distribution.
deb http://br.archive.ubuntu.com/ubuntu/ vivid main restricted
deb-src http://br.archive.ubuntu.com/ubuntu/ vivid main restricted

Major bug fix updates produced after the final release of the
distribution.
deb http://br.archive.ubuntu.com/ubuntu/ vivid-updates main restricted
deb-src http://br.archive.ubuntu.com/ubuntu/ vivid-updates main restricted

N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
team. Also, please note that software in universe WILL NOT receive any
review or updates from the Ubuntu security team.
deb http://br.archive.ubuntu.com/ubuntu/ vivid universe
deb-src http://br.archive.ubuntu.com/ubuntu/ vivid universe

O arquivo era maior, mas com isto já podemos ver como ele funciona. Quando você vai instalar algum programa em específico, pode ser necessário instalar um servidor antes neste arquivo. Sempre que você adiciona um novo servidor, deve executar o comando: **apt-get update**

Pacotes RPM

O Red Hat Package Manager – RPM é um sistema de gerenciamento de pacotes para sistemas GNU/Linux baseados em Red Hat. Ele instala, atualiza, desinstala e verifica softwares.

Uma vantagem que o RPM possui em relação ao DEB é que possui ferramentas de verificação criptográfica com GPG e o Md5, além de verificação de integridade dos arquivos já instalados.

Verificar quais pacotes estão instalados no sistema
rpm -qa

Verificar se um pacote específico está instalado:

rpm -q nomedopacote.rpm

Verificar o que será instalado com um pacote específico:
rpm -qp nomedopacote.rpm

Verificar se a instalação irá ocorrer corretamente:
rpm -ih –test –percent nomedopacote.rpm

As opções -h e –percent servem para mostrar uma barra de progresso e a porcentagem concluída.

Instalar um programa
rpm -ih –percent nomedoprograma.deb

Verifique o que será efetuado ao removermos um pacote:
rpm -e –test nomedopacote

Removendo o pacote:
rpm -e nomedopacote.rpm

Para realizar uma atualização de versão de um determinado programa, podemos usar:
rpm -Uh nomedopacote.rpm

Uma funcionalidade muito boa do RPM é a capacidade de realizar verificações de integridade dos pacotes instalados. Periodicamente, você pode verificar se ocorreu alguma alteração no seu sistema sem você saber ou se sua máquina foi invadida, pode-se tentar identificar o que foi mexido nela.

Verificar a integridade de todos os pacotes:
rpm -Va

Pacotes Slackware

Os pacotes slackware possuem a extensão (tgz).

Instalando pacotes
Installpkg pacote.tgz

Removendo pacotes
Removepkg pacote

Conversor de pacotes Alien

O alien é um conversor de pacotes. Isto significa que você consegue transformar um pacote RPM em DEB e vice versa.

No debian, instale com o comando:

apt-get install alien

Para converter um pacote para **rpm** para **deb** executamos:

alien -d pacote.rpm

Para coverter um pacote **deb** para **rpm**, executamos:

alien -r pacote.deb

Para conver um pacote para tgz, executamos:

alien -t pacote.deb

Não será sempre que o alien será capaz de fazer um pacote compatível com seu sistema. Sempre procure pelo pacote nativo, em último recurso, use o alien.

Instalando pacotes pelo código fonte

Para instalar pacotes pelo código fonte é necessário que você tenha as ferramentas necessárias para compilá-los. No Debian e derivados, instale estes pacotes com:

apt-get install build-essential

Normalmente os pacotes de programas distribuídos através do código fonte são compactados e empacotes nos formatos (tar.gz ou tar.bz2). Para descompactar estes pacotes execute:

tar -xvzf pacote.tar.gz
Para pacotes no formato .tar.gz

tar -jxvf pacate.tar.bz2
Para pacotes no formato .tar.bz2

Após descompactar, acesse o diretório e execute os comandos:

./configure
make
make install

O ./configure executa um script (dentro da pasta do programa), que verifica o sistema em busca de componentes que ele precisa. Ele avisa se algo estiver faltando, por isto, você deve ficar atento a todas as mensagens. O “make” cuida do trabalho pesado, fazendo a compilação. Ele procura os componentes registrados pelo ./configure. O “make install” instala o programa, copiando os arquivos gerados pelo make para as pastas corretas do sistema.

Capítulo 9 – Compactadores e empacotadores

Neste capítulo:

- ✔ Diferença entre empacotar e compactar
- ✔ Usando o tar
- ✔ Usando o gzip/bzip2

No mundo GNU/Linux é muito comum vermos arquivos com a extensão .**tar.gz** ou **tar.bz2** estas extensões significam que além de o criador ter realizado um pacote ele os compactou.

O bzip2 (bz2) faz uma compactação mais eficiente, ou seja, ele reduz mais o tamanho do arquivo do que o gzip (gz), em compensação, ele demora mais para realizar a compactação. Podemos classificar assim:

gzip Rápido, porém não muito eficiente para redução de tamanho
bzip2 Lento, porém muito eficiente para redução de tamanho

Criando um pacote com o TAR

O **tar** é um empacotador, que serve para juntar vários arquivos em um só; imagine a situação cotidiana:
Você precisa enviar algumas fotos através do anexo do e-mail. Você pode fazer uma a uma o que levaria algum tempo, ou empacotá-las criando um único arquivo que você enviará.

Uma única observação: O tar não compacta ele somente empacota. Isto significa, que os arquivos não vão diminuir o tamanho.

Criando um pacote com o tar
Vamos criar um pacote de todo o diretório **/etc,** para isto fazemos:

tar -cvf bkp_etc.tar /etc

Onde:

-c Para criar um backup
-v (verbose) Para mostrar os detalhes na hora da criação
-f Para indicar o nome do arquivo. Essa opção sempre vem por último, pois é ela quem define o nome do arquivo.

Você pode verificar que o tamanho do diretório **/etc** e do arquivo **bkp_etc.tar** é o mesmo, com o comando:

du -h /etc
du -h bkp_etc.tar

Para informar o comando **tar** que você também deseja compactar, use os complementos:

-z Para compactar com o gzip

-j Para compactar com o bzip2

Então o comando mudaria para:

tar -cvzf bkp_etc.tar /etc

ebooklinuxNeste caso para o formato Gzip, ou :

tar -cvjf bkp_etc.tar /etc

Para usarmos o formato bzip2. Agora que empacotamos e compactamos um diretório, vamos aprender como extrai-los. Neste caso usamos o **x** de *extract.*

tar -xvf bkp_etc.tar
Para pacotes (.tar)

tar -xvzf bkp_etc.tar.gz
Para pacotes compactados Gzip (tar.gz)

tar -xvjf bkp_etc.tar.gz
Para pacotes compactador Bzip2 (tar.bz2)

Capítulo 10 – Agendamento de tarefas no Linux

Neste capítulo:

- ✔ Agendador at
- ✔ Agendador crontab

Usamos o **crontab** para agendar comandos para serem executados periodicamente. Já o **at** usamos para agendar execução de programas pontualmente.

Agendador at

Agendaremos que as 19:30 do dia 24 de Junho de 2015 será executado o comando **ls -lha** no diretório **/etc** e o resultado será impresso em um arquivo chamado **listagem.txt** no meu diretório **/tmp.**

Neste caso tenho que abrir o **at** com a sintaxe: **at HH:mm MM/DD/YYYY**

Onde (HH:MM – representa horas e minutos – 19:30), MM (Mês – 06), DD (Dia: 24), YYYY (Ano: 2015). Então o comando seria:

at 19:30 06/24/2015
ls -lha /etc > /home/tux/listagem.txt
<ctrl + d>

Quando você terminar de digitar o seus comandos dentro do **at** é necessário sair gravando com as teclas de atalho: **ctrl + d**

Você pode visualizar os agendamentos com o comando:

atq

Você pode remover um agendamento com o comando **atrm** mais o número do agendamento, que você obtém com o comando **atq.** Por exemplo, ao executar o **atq** tive como retorno:

3 Wed Jun 24 19:40:00 2015 a root

Neste caso, para encerrar este processo eu uso: **atrm 3**

Crontab

Qualquer usuário pode agendar comandos no **crontab** porém ele terá que ter permissão para executar o que está agendando. Para editar o crontab, digite:

crontabe -e

Onde **-e** é de *edit*. Este comando irá abrir um arquivo em branco, caso não tenha feito nenhuma tarefa agendada ainda, para que você possa adicionar novas tarefas. Neste arquivo em branco, temos que adicionar uma nova tarefa seguindo a estrutura:

*** * * * * comando/tarefa**

A estrutura representa:

Minuto	0 – 59
Hora	0 – 23
Dia do Mês	1 – 31
Mês	1 – 12
Dia da Semana	0 – 7 (o “0” e “7” é domingo, “1” segunda”)

Exemplo:

00 14 25 12 0 echo “Hoje é domingo” > arquivo.txt

A tarefa acima vai criar o **arquivo.txt** no dia 25 de Dezembro as 14:00h e este dia precisará ser domingo, caso contrário, isto não será executado.

Para visualizar as tarefas agendadas pelos usuários, como root digite:

crontab -l -u tux

Onde:

-l Para listar

-u Para especificar o usuários

Quando mandar salvar a regra, ou tarefa agendada, a mesma irá para um arquivo com o nome do seu usuário:

ls /var/spool/cron/crontabs/tux

Você pode executar o comando **cat** no seu arquivo para visualizar o que foi agendado.

Digamos que você tenha criado um script que realiza backup dos arquivos dos usuários para um servidor de backup. De segunda a sexta-feira ele deve ser executado as 22 horas. Neste caso execute:

crontab -e

00 22 * * 1-5 /opt/script/scriptbackup.sh

Sempre será necessário colocar o caminho completo (/opt/script/scriptbackup.sh) do seu script.

Entendendo o formato dos operadores de agendamento:

,
Especifica uma lista de valores, por exemplo: “1,3,4,7,8”

-
Especifica um intervalo de valores, por exemplo: 1-5 (De 1 a 5)

*
Especifica todos os valores possíveis

/
Especifica pulos de valores, por exemplo: Se no campo (Hora) usarmos : */3 o comando será executado às: 0, 3, 6, 9, 12, 15, 18, 21

Observamos que os arquivos de crontab dos usuários ficam em:

/var/spool/cron/crontabs

Já a crontab do sistema é encontrada em /etc/crontab e já possui agendamento para realizar as tarefas que se encontram nos diretórios:

/etc/cron.hourly, /etc/cron.daily, /etc/cron.weekly e /etc/cron.monthly

Se você possui um script e deseja que ele seja executado diariamente, faça uma cópia dele para o diretório: **/etc/cron.daily**

cp /opt/script/scriptbackup.sh /etc/cron.daily/

Para que seja semanalmente, copie-o para o diretório **/etc/cron.weekly** e para que seja executado mensalmente copie-o para o diretório **/etc/cron.monthly**

Depois de adicionar os scripts dentro do diretório do cron, será necessário reiniciar o serviço:

/etc/init.d/cron stop
/etc/init.d/cron start

Capitulo 11 – Processos

Neste capítulo:

- ✔ Diretório /proc
- ✔ Funcionamento dos processos pelo linux
- ✔ Monitorando um processo
- ✔ Finalizando um processo

Processo é uma tarefa em execução controlada pelo kernel ocupando uma área na memória (RAM), no qual o Linux usa o mecanismo de identificá-los por números. Ou seja, tudo que estiver rodando no sistema será um processo. É o processo que utiliza os recursos de um computador, como memória e processamento. Um simples comando é um processo, browser aberto é um processo. É responsabilidade do Kernel gerenciar os processos tentando otimizar a performance da CPU – Unidade central de processamento. O primeiro processo do sistema chama-se **init** ele é o pai de todos os outros processos.

Cada processo tem um número diferente de identificação (**PID**). Dessa forma podemos manipulá-los e controlá-los. Controlar é poder matar, reiniciar e pausar um processo.

No diretório **/proc** você verá vários arquivos com informações gerais que o kernel nos fornece, por exemplo os arquivos: cpuinfo, ioports, meminfo, interrupts, mounts, swaps e muitos outros. Todo processo cria temporariamente um diretório em **/proc** com o seu número de PID.

Podemos visualizar o número dos processos executados pelo nosso usuário com o comando:

ps -x

A primeira coluna de retorno deste comando é o número do PID. Neste momento estou com o Libreoffice aberto no meu computador e o número de seu PID é o 4741. Eu posso obter algumas informações acessando o seu diretório:

cd /proc/4741/

Em seguida visualizo o arquivo *cmdline:*

/usr/lib/libreoffice/program/oosplash—writer

Neste caso ele me informou quem é este processo. Nos diretórios dos processos, temos dois arquivos interessantes e curiosos: *cmdline* e *status.*

Você pode obter mais informações sobre o diretório /proc consultando sua página de manual:

man proc

Para listar todos os processos de uma maneira mais simples de visualizar, use:

ps aux | more

Explicando o comando:

a	Todos os processos
u	de todos os usuários
x	processos sem controle pelo terminal

Neste caso todo os processos serão mostrados. É muito melhor, usar o comando **ps** do que acessar o diretório **/proc**.

Vamos analisar agora as colunas de saída do comando **ps aux | more :**

user	Usuários responsável pelo processo
PID	Número que identifica o processo, não se repete
%CPU	Quanto de CPU ele usa
%MEM	Quando de MEM ele usa
VSZ	Tamanho virtual do processo
RSS	Indica a quantidade de memória usada (em KB)
TTY	Terminal que gerou o processo
?	Sem terminal
STAT	Estado do processo, sendo representado por uma letra
R	Executável
D	Em espera no disco
S	Suspenso
T	Interrompido
Z	Zumbi
COMMAND	Nome do processo, ou seja, o comando em si.

Com a opção **f** em **ps,** é possível visualizar em um formato de árvore os processos, identificando-se os processos que possuem filhos:

ps faux | more

Você também pode usar o comando:

pstree

Para visualizar os processos vendo uma tabela que se atualiza de tempos em tempos, recomendo o comando **top**:

top

Este comando gera uma tabela dinâmica com informações sobre os processos, sendo possível matar um processo dentro dele mesmo.

Para matar um processo *(normalmente quando ele está travando o sistema) usamos o comando **kill**, você só pode matar processos os quais você seja o dono (O root, pode matar todos os processos).

Usamos a sintaxe **kill** + o número do PID:

kill -9 4247

A opção **-9** é para forçar a morte do processo. Este número é conhecido como “sinal”. Se o sinal for omitido é usado o **15** como padrão. Este sinal joga um sinal de término. Mas se o processo estiver travado, ou executando alguma coisa, ele não vai terminal e vai ignorar o seu pedido. Já o **-9** força o processo a parar de qualquer jeito!

Na prática, usamos apenas o **-9** e o **-15** para matar um processo, mas você pode ver uma lista completo com o comando:

kill -l

Capítulo 12 – Gerenciamento de processos

Um processo pode passar por vários estados. Quando um processo é criado isso não significa que ele será imediatamente executado. Alguns processos podem ser temporariamente parados para que o processador possa executar um processo com maior prioridade.

O linux trabalha com quatro tipos de estados:

Executável (running): O processo pode ser executado imediatamente;
Dormente (waiting): O processo precisa aguardar alguma coisa para ser executado. Só depois dessa “coisa” acontecer é que ele passa para o estado executável.
Zumbi (zombie): O processo é considerado “morto”, mas, por alguma razão ainda existe;
Parado (stopped): O processo está “Congelado”, ou seja, não pode ser executado.

Classificação de processos

foreground (Primeiro plano): São inicializados no terminal de comandos. Podem interagir com os usuários e exibem sua execução na tela. A desvantagem deste tipo de processo é que ele prende o prompt e isso impede a execução de novos processos neste terminal.

Background (Segundo plano): São inicializados no terminal de comandos, NÃO podem interagir com os usuários e NÃO exibem sua execução na tela. A vantagem desse tipo de processo é que ele NÂO prende o prompt e isso NÂO impede a execução de um novo processo nesse terminal.

Interativo: São inicializados a partir de um terminal do usuário e são controlados por ele, usamos os comandos **ctrl+z, ctrl+x, fg, bg** e **jobs** para trabalhar com esse tipo de processo.

Lote (batch): São processos controlados pelos comandos at, batch e cron.

Daemons

Daemons é um programa em background (segundo plano) esperando que um outro processo solicite seu serviço. Os processos do Daemon fornecem frequentemente serviços de rede, tais como o e-mail, web, etc. Eles não precisam de nenhuma entrada e normalmente não produzem nenhuma saída. Muitos daemons do Linux têm os nomes que terminam em um "d", como o **httpd, ftpd** etc. O oposto de um daemon é um processo que funciona em primeiro plano.

Daemons mais populares:
atd – Roda trabalhos agendados pelo comando **at.**
httpd – Servidor web **apache.**
postifix – Agente de transporte de correios substituto do sendmail.
smbd – O daemon **SAMBA** (ou smb).

Jobs

O comando **jobs** lista os processos em execução pelo shell que estão em **background**.

Jobs -l
[2] 1245 Running asmixer &
[3]+ 1333 Running openoffice &

O parâmetro **-l** faz com que o comando jobs liste também o **PID** de cada processo. Em sua saída o comando **jobs** retorna, além do número do serviço, o estado do processo, a linha digitada no comando e, opcionalmente, o PID.

fg

Permite fazer um programa rodando em segundo plano ou parado, rodar em primeiro plano. Você deve usar o comando **jobs** para pegar o número do processo roando em segundo plano ou interrompido, este número será passado ao comando **fg** para ativá-lo em primeiro plano.

Exemplo:

jobs -l
[2] – 1245 Running openoffice &
fg 2
openoffice

bg

Permite fazer um programa rodando em primeiro plano ou parado, rodar em segundo plano. Para fazer um programa em primeiro plano rodar em segundo, devemos primeiro interromper a execução do comando com **ctrl+z**; será mostrado o número da tarefa interrompida. Use este número com o comando **bg** para iniciar a execução dele em segundo plano.

Exemplo:

fg 2
openoffice

Interrompendo: CTRL+Z
[2]+ Stopped openoffice)

bg 2

Capítulo 13 – Introdução a Redes

Neste capítulo:

- ✔ Introdução a teórica de redes TCP / IP.
- ✔ Conceitos básicos de configurações de redes em UNIX.

Os protocolos TCP/IP

Antigamente os protocolos TCP / IP eram usados por militares para a troca de informações, hoje em dia, este é um padrão mundial para a comunicação de redes. O procolo TCP (Transmission Control Protocol), é orientado a conexões, transporta informações por meio de *handshaking*, ou por meio de retransmissão caso algum erro aconteça. Esse protocolo garante o envio das mensagens. Podemos citar alguns serviços de rede que utilizam o procotolo TCP: SMTP, FTP e Telnet.

Já o procolo IP (Internet Protocol) descrito pela RFC 791, é responsável por estabelecer o esquema de endereçamento e pela definição de datagramas.

Entendendo o IP

O endereçamento IP, como deve ser chamado, é composto por 4 octetos e uma máscara, que determina quantos endereços são destinados a hoste e quantos endereços são destinados a rede. No mundo GNU/linux não diferente dos outros, para termos acesso a internet ou a comunicação em rede também precisamos ter nosso número IP. O numero IP está presente em todas as máquinas, mesmo nas que não tem conexão com a internet. Isso é possível pois em todo GNU/Linux há uma interface lógica, chamada *loopback (lo)* cujo enedereço IP será **127.0.0.1** e que sempre deve estar devidamente configurada.

Algumas aplicações de sua máquina utilizam o IP 127.0.0.1 para comunicação com outras aplicações internas. Além disso, você pode desenvolver suas páginas e sistemas Web e testá-las

localmente. Ou mesmo testar a implantação de um servidor de DNS ou um proxy, antes de colocá-lo em produção.

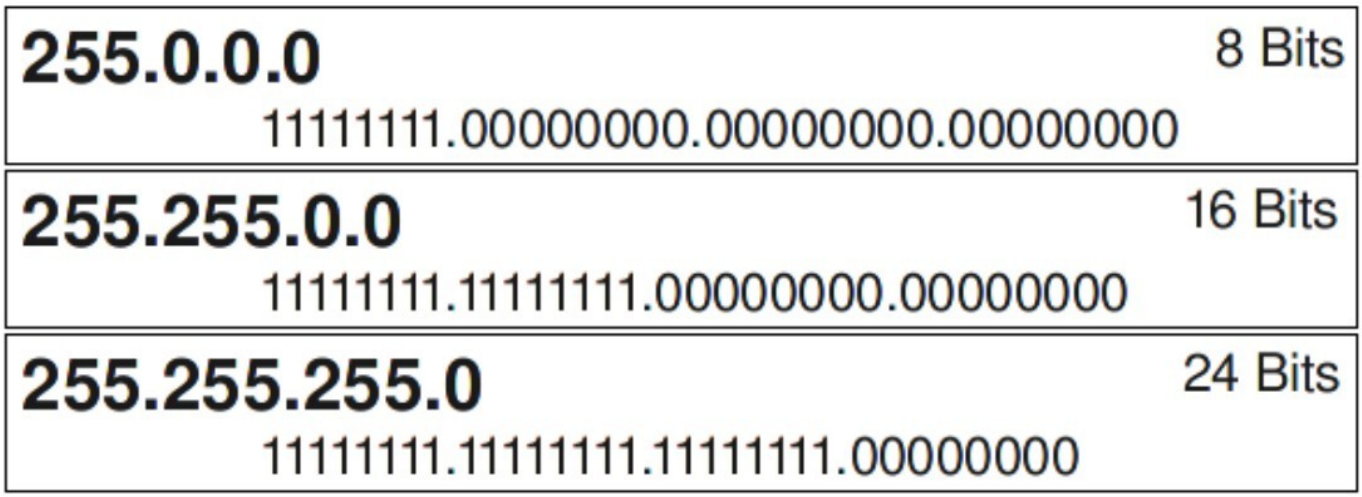
A Internet é totalmente endereçada por números IP's e não depende de um servidor DNS para funcionar. O serviço DNS apenas facilita o nosso acesso a internet, permitindo que a navegação seja feita através de nomes e não de números.

Para configurarmos um número IP em nosso computador, precisamos também, configurar uma "Máscara" para esse número IP. A máscara de rede, também conhecida como **netmask** é um número constituído por 32 bits, que é utilizado para separar redes, determinando quem é "host", quem é rede e quem é "broadcast".

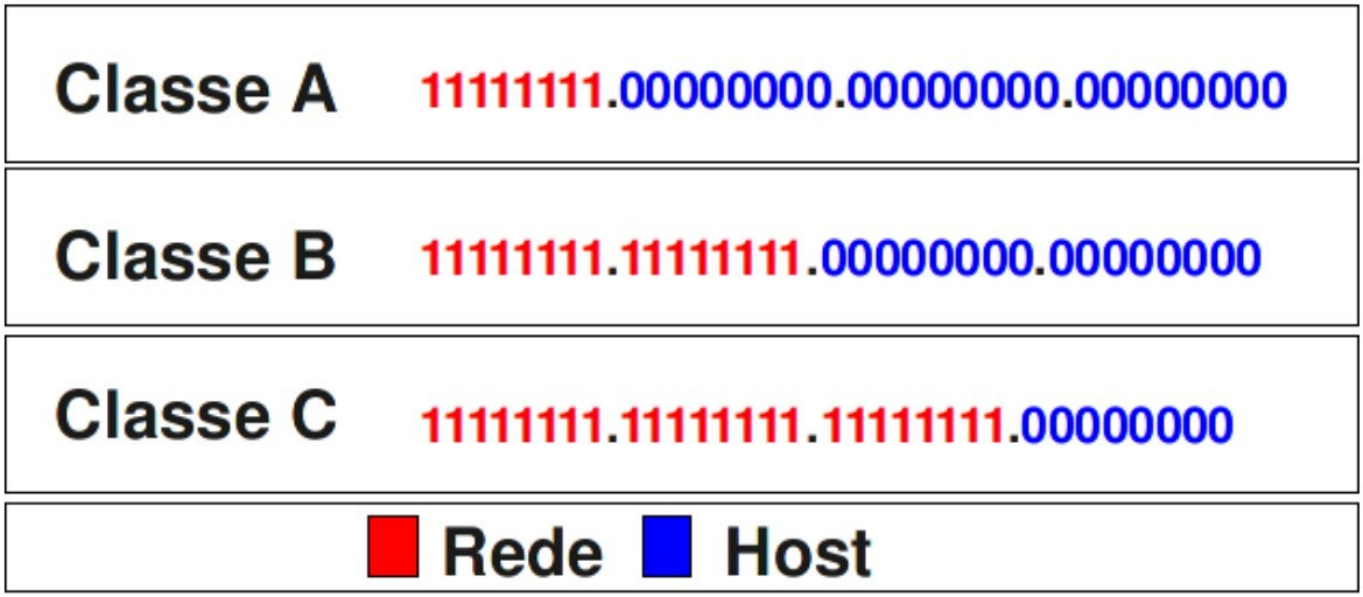
Host – Um endereço disponibilizado para computadores poderem acessar a rede;

Network – Normalmente é o primeiro endereço da rede;

Broadcast – Normalmente é o último endereço da rede, utilizado para que uma máquina possa falar com todas as outras.



Além da máscara de rede é necessário saber que existem três classes padrões de redes.



Entendendo o Gateway

O principal papel do “Gateway” é levar os pacotes “TCP/IP” para outras redes que os hosts que os originaram, não conhecem. É dessa forma que os pacotes saem de uma rede privada para uma rede “WAN”. Para que os pacotes possam transitar pela internet ou mesmo só por uma rede fechada é necessário um “Gateway”. Mesmo em uma rede local, o “gateway” é a própria máquina, pois todos os “hosts” estão normalmente com a mesma configuração de IP, ou seja, mesma máscara, mesma classe de IP, etc.

O servidor DNS

O DNS é um serviço que converte endereços IP em nomes, deste modo, ao invés de você digitar em seu navegador 200.1.100.09 – Você digita: www.tux4.com.br . Como já mencionado o DNS não faz parte da configuração essencial da rede, já que a internet funciona através dos números IP.

Se você testar o endereço: www.tux4.com.br e ele não funcionar, tente acessá-lo com: 200.1.100.09 se conseguir o problema está no seu servidor DNS.

* O IP (200.1.100.09) mencionado neste capítulo é fictício e não pertence ao domínio tux4.com.br

Configurando a Rede

A configuração da rede é baseada em três etapas:

- Configuração do número IP e máscara
- Configuração do Gateway
- Configuração do DNS Server

Em um sistema GNU/Linux, para configurar a nossa interface de rede utilizamos o comando **ifconfig.**

ifconfig

Executando este comando é possível descobrir todas as interfaces presentes no sistema. Por padrão em quase todas as distribuições de Linux a interface padrão é identificada como **eth0**. Para verificar de modo mais apurado, inclusive pesquisando interfaces inativas, usamos o parâmetro **-a** como no exemplo:

ifconfig -a

A sintaxe do comando **ifconfig** para atribuir um endereço de IP:

ifconfig <interface> <ip>

Exemplo:

ifconfig eth0 192.168.1.10

Com este comando estamos atribuindo o endereço (192.168.1.0) para a interface eth0. O comando **ifconfig** calcula automaticamente a máscara, mas se você precisar configurar uma mascara diferenciada, você deve usar o parâmetro netmask:

ifconfig eth0 192.168.1.10 netmask 255.255.254.0

Para ativar ou desabilitar uma placa de rede podemos usar a sintaxe:

ifconfig eth0 up
ifconfig eth0 down

Configurando o gateway

Para que nossos pacotes saibam para onde ir eles precisam conhecer o IP do Gateway da rede. O papel do gateway é simples, ele funciona como uma saída para os pacotes de uma rede, para outras redes.
Para configurar o gateway da nossa rede utilizamos o comando **route** com os seguintes parâmetros:

route add default gw IP

Com este comando é possível configurar a rota padrão de saída da nossa rede. Para listar todas as rotas traçadas, podemos utilizar o comando abaixo:

route -n

Com ele podemos descobrir se as rotas necessárias para que nossa rede funcione estão corretas. Se desejaremos remover a rota padrão, devemos usar o comando:

route del default

Este comando irá remover a rota padrão de saída da rede.

Configuração do DNS Server

Para configurar os servidores DNS para a máquina local, precisamos editar o arquivo de configuração de DNS, chamado **resolv.conf** localizado em /etc.

nano /etc/resolv.conf

Dentro deste arquivo podemos configurar os nossos servidores de DNS, para isto, coloque as seguintes linhas:

nameserver 201.6.0.100

Com esta sintaxe configurarmos um servidor DNS, neste caso é o DNS do Serviço de Banda Larga - Virtua.

Configuração estática da rede

Toda a configuração da rede pode ser atribuída pela linha de comando, como observamos até agora. Porém, para que estas configurações estejam ativas após você reiniciar o computador é necessário editá-las em arquivo.

Configure o arquivo:
/etc/network/interfaces

Exemplo de configuração:

auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 192.168.200.x
netmask 255.255.255.0
broadcast 192.168.200.255
network 192.168.200.0
gateway 192.168.200.254

Podemos também configurar alguns atalhos para endereços de rede. Estes atalhos ficam localizados dentro do arquivo **/etc/hosts**

A sintaxe dele é:

IP apelido apelido

Exemplo:

192.168.200.234 professor ramos

Isso facilita nosso trabalho. Após apelidar os hosts, você não mais precisará decorar endereços IPS.

Comando hostname

O comando hostname altera dinamicamente o nome da máquina e deve ser utilizado da seguinte maneira:

hostname Novo_Nome

Para alterar de forma estática, ou seja, para que o nome se mantenha após a reinicialização do computador, altere o arquivo **/etc/hostname**

Publique seu e-book na Tux4you

Além da rede social (www.tux4.com.br/networking) temos o acervo digital gratuito, que é o local onde disponibilizamos mídias para download de forma gratuita (www.tux4.com.br/acervodigital). Você pode colaborar enviando artigos e e-books, para isto, encaminhe sua documentação em formato aberto (.odt) para o e-mail : contato@tux4.com.br

7 Cursos de Linux para você!

Tornamos todo o nosso trabalho sustentável com a venda de nossos cursos que podem ser encontrados no endereço: (www.cursos.tux4.com.br).

Estamos com uma promoção incrível e digo sem medo de errar que inédita, que são 7 cursos de Linux pelo preço de R$ 156,00.

Saiba mais no link:

http://tux4.com.br/networking/7-cursos-pelo-preco-de-1/