

Guia

Administração

Certificação

LPIC-1 101

LPIC 1 - 101

CompTIA Linux+

Novell Linux Administrator

Administração Linux e Certificação LPIC-1 - 101

*Rede social livre:
www.lcenter.com.br*

@Juliano Ramos



Com sua licença quero me apresentar e compartilhar um pouco de minha vida e intimidade. Meu nome é Juliano Ramos e sou Engenheiro Linux certificado LPI, trabalho com software livre à mais de 11 anos. Na prefeitura de São Paulo acompanhei as primeiras unidades de Telecentro - Centro de Inclusão Digital em comunidades carentes e coordenei estes telecentros por quase cinco anos até receber um grande convite do Presidente César Rossi da empresa IBRATI - Instituto Brasileiro de Tecnologia da informação em São Paulo para coordenar o programa social de sua empresa, certifiquei mais de sete mil pessoas em cursos diversos como Hardware, programação, Linux e Windows. Desenvolvi diversas apostilas, How-To e e-books para a comunidade do software livre. Hoje, sou responsável pela empresa Magazine 360, presidente da ONG ISBIT - Instituto Social Brasileiro de Inclusão Tecnológica, voluntário na ONG Elo Solidário e na ONG Tucça, Palestrante e consultor Linux, Blogueiro ativo de cinco blogs de Linux e comunidades do facebook, pai e um esposo apaixonado e isto tudo é muito pouco!

Mas minha história nem sempre foi assim e é sobre isto que quero compartilhar. Passei doze anos da minha vida nas garras da adicção (dependência química e alcoolismo), estive abaixo do fundo do poço, na lama mesmo, sem saúde, paz e quase sem família. Perdi o amor próprio e como sempre digo:

Vivia para usar e usava para viver.

A doença que obtive é psicológica, fisiológica e espiritual. Mas felizmente, depois de sofrer muito com depressão, síndrome do pânico e os efeitos das drogas em si, tive um grande despertar espiritual.

Deus entrou em minha vida com uma força tão grande que me fez pular literalmente para a vida e em poucos dias já corria anunciando a gloriosa dádiva que é "Viver em sobriedade e serenidade". Neste inferno que estive, contei com grandes amigos para me ajudar à sair dele, mas em especial quero agradecer à minha mãe que calejou os joelhos em oração, minha filha que me deu esperança e minha esposa que é a mulher mais guerreira que conheço e por quem amarei nesta e em todas as vidas! Também vai um abraço e um beijo de irmão para meu amigo Clayton Nogueira, que nunca desistiu de mim; abrigando-me em sua casa e dando-me total apoio na época difícil da abstinência. - OBRIGADO DEUS, Just For Today.

Solicite uma palestra minha sobre drogas e alcoolismo: Juliano@lcenter.com.br

Licenciamento.....	7
A história do GNU/Linux.....	8
Licença Pública Geral GNU.....	12
Distribuição de Gnu/linux.....	19
Instalando uma distro Linux - Trisquel.....	25
LPIC-1 Exame 101 + CompTIA Linux+.....	39
Arquitetura de Hardware	41
Interrupções.....	43
Dispositivos PCI.....	45
Configurando modems e placas de som.....	48
Configurando dispositivos não-IDE.....	50
Configurando dispositivos de Rede.....	52
Configurando IP estático.....	54
Configurando IP dinâmico PPPoX.....	55
Configurando dispositivos USB.....	57
Processos de Boot.....	59
Grub.....	62
Alterando runlevels, shutdown e reboot.....	67
Instalação e gerenciamento de pacotes.....	69
Compilar e instalar programas.....	73
Gerenciamento de pacotes DEB.....	76
Gerenciamento de pacotes RPM.....	81
Gerenciamento de pacotes YUM.....	84
Comandos GNU e Unix.....	85
Utilizando filtros de texto.....	95
Diretórios e arquivos.....	103
Compressores.....	108
Clonar dispositivos.....	109
Executando e eliminando processos.....	110
Editor de texto VI.....	116
Dispositivos, sistemas de arquivos e o FHS.....	121
Administrando cotas de discos.....	138
Permissões de arquivos no Linux.....	144
Links.....	151
Localizando arquivos no linux.....	152

“O Linux é capaz de baratear e democratizar a informática. O custo final de um computador sem o ©Microsoft Windows e o pacote Microsoft Office chega a ser quarenta e cinco por cento mais barato. Com isto podemos levar computadores a classes mais desfavorecidas sem apelar para a pirataria de software”

Licenciamento

Qualquer parte deste material pode ser copiado/reproduzido em qualquer meio gráfico, mecânico ou eletrônico, desde que a nota de rodapé abaixo seja adicionada:

Versão original: Janeiro 2014 - Administração Linux e Certificação LPIC 1- 101 - Juliano Ramos de Oliveira contato@magazine360.com.br / www.magazine360.com.br / (11) 2051-4516

Todos os cuidados foram tomados para garantir a integridade das informações contidas neste documento, sendo que o autor não assume responsabilidades sobre qualquer erro, omissão ou danos resultantes da utilização das informações contidas neste material, incluindo os códigos e comandos de exemplo, direta ou indiretamente.

Quaisquer produtos referenciado neste documento podem ser marcas registradas de seus respectivos proprietários. O autor não detém direitos sobre nenhuma destas marcas.

Seja livre para piratear este livro.

Origem do Gnu/Linux

"Caminhamos pela fé e não pela visão"

Jesus Cristo

A história do GNU/Linux

O nome Linux, surgiu da mistura de Linus + Unix. Linus é o nome do criador do Linux, Linus Torvalds e Unix, é o nome de um sistema operacional de grande porte, no qual contaremos sua história agora, para que você entenda melhor a do Linux: A origem do Unix tem ligação com o sistema operacional Multics, projetado na década de 1960. Esse projeto era realizado pelo Massachusetts Institute of Technology (conhecido como MIT), pela General Electric (GE) e pelos laboratórios Bell (Bell Labs) e American Telephone Telegraph (AT&T). A intenção era de que o Multics tivesse características de tempo compartilhado (vários usuários compartilhando os recursos de um único computador), sendo assim, o sistema mais arrojado da época. Em 1969, já existia uma versão do Multics rodando num computador GE645. Ken Thompson, era um pesquisador do Multics e trabalhava na Bell Labs.

No entanto, a empresa se retirou do projeto tempos depois, mas ele continuou seus estudos no sistema; desde então, sua ideia não era continuar no Multics original e sim criar algo menor, mas que conservasse as ideias básicas do sistema. A partir daí, começou a saga do sistema Unix.

Brian Kernighan, também pesquisador da Bell Labs, foi quem deu esse nome. Em 1973, outro pesquisador da Bell Labs, Dennis Ritchie, rescreveu todo o sistema Unix numa linguagem de alto nível, chamada C, desenvolvida por ele mesmo. Por causa disso, o sistema passou a ter grande aceitação por usuários externos à Bell Labs. Entre 1977 e 1981, a AT&T, alterou o Unix, fazendo algumas mudanças particulares e lançou o System III. Em 1983, após mais uma série de modificações, foi lançado o conhecido Unix System IV, que passou a ser vendido. Até hoje esse sistema é usado no mercado, tornando-se o padrão internacional do Unix. Esse sistema é comercializado por empresas como IBM, HP, Sun, etc. O Unix, é um sistema operacional muito caro e é usado em computadores poderosos por diversas multinacionais.

Mas, qual a relação entre o Unix e o Linux, ou melhor, entre o Unix e Linus Torvalds? Para responder essa pergunta, é necessário falar de outro sistema operacional, o Minix.

O Minix é uma versão do Unix, porém, gratuita e com o código fonte disponível. Isso significa que qualquer programador experiente pode fazer alterações nele. Ele foi criado originalmente para uso educacional, para quem quisesse estudar o Unix em casa. No entanto, vale citar que ele foi escrito do "zero" e apesar de ser uma versão do Unix, não contém nenhum código da AT&T e por isso pode ser distribuído gratuitamente.

A partir daí, "entra em cena", Linus Torvalds. Ele era um estudante de Ciências da Computação da Universidade de Helsinki, na Finlândia. E em 1991, por hobby, Linus decide desenvolver um sistema mais poderoso que o Minix. No mesmo ano, ele disponibilizou a versão do kernel (núcleo dos sistemas operacionais) 0.02 e continuou trabalhando até que em 1994, quando disponibilizou a versão 1.0. Até o momento em que este artigo estava sendo escrito, a versão atual do kernel do linux é o 2.6.26. O Linux é um sistema operacional livre, e é uma re-implementação das especificações POSIX (padronização da IEEE, Instituto de Engenharia Elétrica e Eletrônica) para sistemas com extensões System V e BSD. Isso significa que o Linux, é bem parecido com Unix, mas não vem do mesmo lugar e foi escrito de outra forma.

Linus Benedict Torvalds



Linus Benedict Torvalds, é o criador do kernel do sistema operacional GNU/Linux muitas vezes chamado apenas de "linux". Linus nasceu em Helsinque capital da República da Finlândia no dia 28 de Dezembro de 1969; e no fim dos anos 80 já programava em **linguagem C** com maestria. Sua relação com o Linux começou quando Linus desejou rodar em seu computador pessoal o sistema operacional Minix; descontente com os recursos deste sistema, começou a desenvolver o seu próprio Kernel. No ano de 1991 Torvalds lançou o seu projeto em uma famosa mensagem para a Usenet, desde os primeiros dias ele recebeu ajuda de hackers do Minix, e hoje recebe contribuições de milhares de programadores no mundo todo.

A mensagem de Torvalds na Usenet:

Tradução para o português:

Você sente falta dos dias do Minix/1.1 quando homens eram homens e escreviam seus próprios drivers? Você está sem nenhum projeto legal e está ansioso para mexer num sistema operacional que você possa modificar para atender às suas necessidades? Você está achando chato quando tudo funciona no minix? Não ficar mais a noite inteira tentando arrumar um programa legal? Então esta mensagem pode ser para você.

Como eu disse há um mês (?) atrás, eu estou trabalhando numa versão grátis dum similar para o Minix, para computadores AT-386. Ela finalmente atingiu o estágio onde já é usável (apesar de talvez não ser, dependendo do que você quer), e eu estou a fim de colocar (online) o código fonte para uma distribuição melhor. É apenas a versão 0.02 (com mais um patch) mas eu já rodei bash/gcc/gnu-make/gnu-sed/compress dentro dela. Códigos fontes para este hobby meu podem ser encontradas em nic.funet.fi(128.214.6.100) no diretório /pub/OS/Linux. O diretório também contem alguns arquivos README e um conjunto de arquivos para permitir trabalho no Linux (bash, update e GCC, o que mais você queria? :-).

O código-fonte do kernel está disponível por inteiro, porque nenhum do código do Minix foi usado. Os códigos-fontes das bibliotecas são apenas parcialmente abertos, portanto não podem ser distribuídos. O sistema pode compilar "como está" e é provado que funciona. (hehehe) Código-fonte dos programas (bash e gcc) podem ser encontrados no mesmo FTP em /pub/gnu.PERIGO! AVISO! NOTA! Este código fonte ainda precisa do Minix/386 para compilar (e o gcc-1.4.0, ou o 1.3.7, não testei) e você precisa do Minix para configurá-lo, então ele ainda não é um sistema por si só para vocês que não tem o Minix. Eu já estou trabalhando nisto.

Você também precisa ter um jeito hacker (?) para configurá-lo, então para aqueles torcendo por uma alternativa ao Minix/386, me esqueçam. Ele é atualmente para hackers com interesse no 386 e no Minix. O sistema precisa de um monitor EGA/VGA e um disco rígido compatível (IDE serve). Se você ainda está interessado, pegue no FTP o readme/relnotes e/ou me mande um e-mail para saber mais. Eu posso (bem, quase) ouvir vocês perguntando para si mesmos: porquê? O Hurd vai sair em um ano (ou dois, ou em um mês, quem sabe), e eu já tenho o Minix. Este é um programa feito por e para hackers. Eu gostei de fazer ele, e alguém pode começar a olhá-lo e até mesmo modificá-lo às suas necessidades. Ele ainda é pequeno para entender, usar e modificar, e eu estou otimista em relação a algum comentário que vocês tenham a fazer. Eu também estou interessado em alguém que tenha escrito alguns dos utilitários/bibliotecas para o Minix. Se o seu trabalho pode ser distribuído publicamente (registrado ou mesmo domínio público), eu gostaria de ouvir comentários de vocês, e para que eu possa adicioná-los ao sistema. Eu estou usando o Earl Chews estudio agora mesmo (obrigado, Earl, por um sistema que funciona), e trabalhos similares seriam bem-vindos. Seus (C)'s obviamente serão mantidos. Me deixe uma mensagem se você quer deixar que a gente use seu código.

GNU, GPL e Software Livre



O projeto GNU que significa (GNU's Not Unix) "Gnu não é UNIX", foi iniciado em 1984 por Richard Stallman, com o intuito de que seus utilizadores possam gozar de mais liberdade, e não serem reféns de empresas que os vendem. Da mesma forma, os programadores não devem ser limitados, mas sim ter livre acesso a reutilizar o código fonte. O objetivo principal de Richard Stallman era a construção de um sistema operacional chamado GNU completo e livre, similar ao UNIX. No início da década de 90, o GNU estava quase pronto. Já possuía um conjunto de

*bibliotecas denominado de GNU lib C, além de outras ferramentas como os comandos: `grep`, `cut`, `ls`, `mkdir`, `rmdir`, `ln` e algumas ferramentas de compilação como: `make`, `gcc`, `autoconf`. Mas para que o sistema operacional estivesse completo era necessário um kernel e este estava sendo criado com o nome de **HURD**, infelizmente a um processo muito lento. Neste tempo Linus Torvalds lançou o seu kernel e muitas pessoas começaram a juntar este kernel com os aplicativos GNU, formalizando uma parceria perfeita. Daí a designação mais correta dos sistemas denominados "Linux" é "GNU/Linux".*

Licença Pública Geral do GNU (GPL) [General Public License]

This is an unofficial translation of the GNU General Public License into Portuguese. It was not published by the Free Software Foundation, and does not legally state the distribution terms for software that uses the GNU GPL--only the original English text of the GNU GPL does that. However, we hope that this translation will help Portuguese speakers understand the GNU GPL better.

Esta é uma tradução não-oficial da GNU General Public License para o Português. Ela não é publicada pela Free Software Foundation e não traz os termos de distribuição legal do software que usa a GNU GPL -- estes termos estão contidos apenas no texto da GNU GPL original em inglês. No entanto, esperamos que esta tradução ajudará no melhor entendimento da GNU GPL em Português.

Versão 2, Junho de 1991 Direitos Autorais Reservados © 1989, 1991 Free Software Foundation, Inc. 59 Temple Place, Suite [conjunto] 330, Boston, MA [Massachusetts] 02111-1307 USA [Estados Unidos da América] É permitido a qualquer pessoa copiar e distribuir cópias sem alterações deste documento de licença, sendo vedada, entretanto, qualquer modificação.

Introdução

As licenças da maioria dos softwares são elaboradas para suprimir sua liberdade de compartilhá-los e modificá-los. A Licença Pública Geral do GNU, ao contrário, visa garantir sua liberdade de compartilhar e modificar softwares livres para assegurar que o software seja livre para todos os seus usuários. Esta Licença Pública Geral é aplicável à maioria dos softwares da [Free Software Foundation](#) [Fundação do Software Livre] e a qualquer outro programa cujos autores se comprometerem a usá-la. (Em vez dela, alguns outros softwares da Free Software Foundation são cobertos pela Licença Pública Geral de Biblioteca do GNU). Você também poderá aplicá-la aos seus programas. Quando falamos de software livre, estamos nos referindo à liberdade, não ao preço. Nossas Licenças Públicas Gerais visam garantir que você tenha a liberdade de distribuir cópias de software livre (e cobrar por isso se desejar), que receba código-fonte ou possa obtê-lo se desejar, que possa modificá-lo ou usar partes dele em novos programas livres; finalmente, que você tenha ciência de que pode fazer tudo isso. Para proteger seus direitos, necessitamos fazer restrições que proíbem que alguém negue esses direitos a você ou que solicite que você renuncie a eles. Essas restrições se traduzem em determinadas responsabilidades que você deverá assumir, se for distribuir cópias do software ou modificá-lo. Por exemplo, se você distribuir cópias de algum desses programas, tanto gratuitamente como mediante uma taxa, você terá de conceder aos receptores todos os direitos que você possui. Você terá de garantir que, também eles, recebam ou possam obter o código-fonte. E você terá a obrigação de exibir a eles esses termos, para que eles conheçam seus direitos. Protegemos seus direitos através de dois passos: (1) estabelecendo direitos autorais sobre o software e (2) concedendo a você esta licença, que dá permissão legal para copiar, distribuir e/ou modificar o software. Além disso, para a proteção de cada autor e a nossa, queremos ter certeza de que todos entendam que não há nenhuma garantia para este software livre. Se o software for modificado por alguém e passado adiante, queremos que seus receptores saibam que o que receberam não é o original, de forma que quaisquer

problemas introduzidos por terceiros não afetem as reputações dos autores originais. Finalmente, qualquer programa livre é constantemente ameaçado por patentes de software. Queremos evitar o risco de que redistribuidores de um programa livre obtenham individualmente licenças sob uma patente, tornando o programa, com efeito, proprietário. Para impedir isso, deixamos claro que qualquer patente deve ser licenciada para o uso livre por parte de qualquer pessoa ou, então, simplesmente não deve ser licenciada. Os exatos termos e condições para cópia, distribuição e modificação seguem abaixo.

TERMOS E CONDIÇÕES PARA CÓPIA, DISTRIBUIÇÃO E MODIFICAÇÃO

0. Esta Licença se aplica a qualquer programa ou outra obra que contenha um aviso inserido pelo respectivo titular dos direitos autorais <http://linuxead.com/blog/ais>, informando que a referida obra pode ser distribuída em conformidade com os termos desta Licença Pública Geral. O termo "Programa", utilizado abaixo, refere-se a qualquer programa ou obra, e o termo "obras baseadas no Programa" significa tanto o Programa, como qualquer obra derivada nos termos da legislação de direitos autorais: isto é, uma obra contendo o Programa ou uma parte dele, tanto de forma idêntica como com modificações, e/ou traduzida para outra linguagem. (Doravante, o termo "modificação" inclui também, sem reservas, a tradução). Cada licenciado, doravante, será denominado "você".

Outras atividades que não a cópia, distribuição e modificação, não são cobertas por esta Licença; elas estão fora de seu escopo. O ato de executar o Programa não tem restrições e o resultado gerado a partir do Programa encontra-se coberto somente se seu conteúdo constituir uma obra baseada no Programa (independente de ter sido produzida pela execução do Programa). Na verdade, isto dependerá daquilo que o Programa faz.

1. Você poderá fazer cópias idênticas do código-fonte do Programa ao recebê-lo e distribuí-las, em qualquer mídia ou meio, desde que publique, de forma ostensiva e adequada, em cada cópia, um aviso de direitos autorais (ou copyright) apropriado e uma notificação sobre a exoneração de garantia; mantenha intactas as informações, avisos ou notificações referentes a esta Licença e à ausência de qualquer garantia; e forneça a quaisquer outros receptores do Programa uma cópia desta Licença junto com o Programa.

Você poderá cobrar um valor pelo ato físico de transferir uma cópia, e você pode oferecer, se quiser, a proteção de uma garantia em troca de um valor.

2. Você poderá modificar sua cópia ou cópias do Programa ou qualquer parte dele, formando, dessa forma, uma obra baseada no Programa, bem como copiar e distribuir essas modificações ou obra, de acordo com os termos da Cláusula 1 acima, desde que você também atenda a todas as seguintes condições:

a. Você deve fazer com que os arquivos modificados contenham avisos, em destaque, informando que você modificou os arquivos, bem como a data de qualquer modificação.

- b. *Você deve fazer com que qualquer obra que você distribuir ou publicar, que no todo ou em parte contenha o Programa ou seja dele derivada, ou derivada de qualquer parte dele, seja licenciada como um todo sem qualquer custo para todos terceiros nos termos desta licença.*
- c. *Se o programa modificado normalmente lê comandos interativamente quando executado, você deverá fazer com que ele, ao começar a ser executado para esse uso interativo em sua forma mais simples, imprima ou exiba um aviso incluindo o aviso de direitos autorais (ou copyright) apropriado, além de uma notificação de que não há garantia (ou, então, informando que você oferece garantia) e informando que os usuários poderão redistribuir o programa de acordo com essas condições, esclarecendo ao usuário como visualizar uma cópia desta Licença. (Exceção: se o Programa em si for interativo mas não imprimir normalmente avisos como esses, não é obrigatório que a sua obra baseada no Programa imprima um aviso). Essas exigências se aplicam à obra modificada como um todo. Se partes identificáveis dessa obra não forem derivadas do Programa e puderem ser consideradas razoavelmente como obras independentes e separadas por si próprias, nesse caso, esta Licença e seus termos não se aplicarão a essas partes quando você distribuí-las como obras separadas. Todavia, quando você distribuí-las como parte de um todo que constitui uma obra baseada no Programa, a distribuição deste todo terá de ser realizada em conformidade com esta Licença, cujas permissões para outros licenciados se estenderão à obra por completo e, conseqüentemente, a toda e qualquer parte, independentemente de quem a escreveu. Portanto, esta cláusula não tem a intenção de afirmar direitos ou contestar os seus direitos sobre uma obra escrita inteiramente por você; a intenção é, antes, de exercer o direito de controlar a distribuição de obras derivadas ou obras coletivas baseadas no Programa.*
- Além do mais, a simples agregação de outra obra que não seja baseada no Programa a ele (ou a uma obra baseada no Programa) em um volume de mídia ou meio de armazenamento ou distribuição, não inclui esta outra obra no âmbito desta Licença.*
3. *Você poderá copiar e distribuir o Programa (ou uma obra baseada nele, de acordo com a Cláusula 2) em código-objeto ou formato executável de acordo com os termos das Cláusulas 1 e 2 acima, desde que você também tome uma das providências seguintes:*
- a. *Incluir o código-fonte correspondente completo, passível de leitura pela máquina, o qual terá de ser distribuído de acordo com as Cláusulas 1 e 2 acima, em um meio ou mídia habitualmente usado para intercâmbio de software; ou,*
- b. *Incluir uma oferta por escrito, válida por pelo menos três anos, para fornecer a qualquer terceiro, por um custo que não seja superior ao seu custo de fisicamente realizar a distribuição da fonte, uma cópia completa passível de leitura pela máquina, do código-fonte correspondente, a ser distribuído de acordo com as Cláusulas 1 e 2*

acima, em um meio ou mídia habitualmente usado para intercâmbio de software; ou,

- c. Incluir as informações recebidas por você, quanto à oferta para distribuir o código-fonte correspondente. (Esta alternativa é permitida somente para distribuição não-comercial e apenas se você tiver recebido o programa em código-objeto ou formato executável com essa oferta, de acordo com a letra b, acima).*

O código-fonte de uma obra significa o formato preferencial da obra para que sejam feitas modificações na mesma. Para uma obra executável, o código-fonte completo significa o código-fonte inteiro de todos os módulos que ela contiver, mais quaisquer arquivos de definição de interface associados, além dos scripts usados para controlar a compilação e instalação do executável. Entretanto, como uma exceção especial, o código-fonte distribuído não precisa incluir nada que não seja normalmente distribuído (tanto no formato fonte como no binário) com os componentes principais (compilador, kernel e assim por diante) do sistema operacional no qual o executável é executado, a menos que este componente em si acompanhe o executável.

Se a distribuição do executável ou código-objeto for feita mediante a permissão de acesso para copiar, a partir de um local designado, então, a permissão de acesso equivalente para copiar o código-fonte a partir do mesmo local será considerada como distribuição do código-fonte, mesmo que os terceiros não sejam levados a copiar a fonte junto com o código-objeto.

- 4. Você não poderá copiar, modificar, sublicenciar ou distribuir o Programa, exceto conforme expressamente estabelecido nesta Licença. Qualquer tentativa de, de outro modo, copiar, modificar, sublicenciar ou distribuir o Programa será inválida, e automaticamente rescindir seus direitos sob esta Licença. Entretanto, terceiros que tiverem recebido cópias ou direitos de você de acordo esta Licença não terão suas licenças rescindidas, enquanto estes terceiros mantiverem o seu pleno cumprimento.*
- 5. Você não é obrigado a aceitar esta Licença, uma vez que você não a assinou. Porém, nada mais concede a você permissão para modificar ou distribuir o Programa ou respectivas obras derivativas. Tais atos são proibidos por lei se você não aceitar esta Licença. Consequentemente, ao modificar ou distribuir o Programa (ou qualquer obra baseada no Programa), você estará manifestando sua aceitação desta Licença para fazê-lo, bem como de todos os seus termos e condições para copiar, distribuir ou modificar o Programa ou obras nele baseadas.*
- 6. Cada vez que você redistribuir o Programa (ou obra baseada no Programa), o receptor receberá, automaticamente, uma licença do licenciante original, para copiar, distribuir ou modificar o Programa, sujeito a estes termos e condições. Você não poderá impor quaisquer restrições adicionais ao exercício, pelos receptores, dos direitos concedidos por este instrumento. Você não tem responsabilidade de promover o cumprimento por parte de terceiros desta licença.*
- 7. Se, como resultado de uma sentença judicial ou alegação de violação de patente, ou por qualquer outro motivo (não restrito às questões de*

patentes), forem impostas a você condições (tanto através de mandado judicial, contrato ou qualquer outra forma) que contradigam as condições desta Licença, você não estará desobrigado quanto às condições desta Licença. Se você não puder atuar como distribuidor de modo a satisfazer simultaneamente suas obrigações sob esta licença e quaisquer outras obrigações pertinentes, então, como consequência, você não poderá distribuir o Programa de nenhuma forma. Por exemplo, se uma licença sob uma paten<http://linuxead.com/blog/te> não permite a redistribuição por parte de todos aqueles que tiverem recebido cópias, direta ou indiretamente de você, sem o pagamento de royalties, então, a única forma de cumprir tanto com esta exigência quanto com esta licença será deixar de distribuir, por completo, o Programa. Se qualquer parte desta Cláusula for considerada inválida ou não executável, sob qualquer circunstância específica, o restante da cláusula deverá continuar a ser aplicado e a cláusula, como um todo, deverá ser aplicada em outras circunstâncias.

Esta cláusula não tem a finalidade de induzir você a infringir quaisquer patentes ou direitos de propriedade, nem de contestar a validade de quaisquer reivindicações deste tipo; a única finalidade desta cláusula é proteger a integridade do sistema de distribuição do software livre, o qual é implementado mediante práticas de licenças públicas.

8. Muitas pessoas têm feito generosas contribuições à ampla gama de software distribuído através desse sistema, confiando na aplicação consistente deste sistema; cabe ao autor/doador decidir se deseja distribuir software através de qualquer outro sistema e um licenciado não pode impor esta escolha.

Esta cláusula visa deixar absolutamente claro o que se acredita ser uma consequência do restante desta Licença.

9. Se a distribuição e/ou uso do Programa for restrito em determinados países, tanto por patentes ou por interfaces protegidas por direito autoral, o titular original dos direitos autorais que colocar o Programa sob esta Licença poderá acrescentar uma limitação geográfica de distribuição explícita excluindo esses países, de modo que a distribuição seja permitida somente nos países ou entre os países que não foram excluídos dessa forma. Nesse caso, esta Licença passa a incorporar a limitação como se esta tivesse sido escrita no corpo desta Licença.

10. A Free Software Foundation poderá de tempos em tempos publicar novas versões e/ou versões revisadas da Licença Pública Geral. Essas novas versões serão semelhantes em espírito à presente versão, mas podem diferenciar-se, porém, em detalhe, para tratar de novos problemas ou preocupações.

Cada versão recebe um número de versão distinto. Se o Programa especificar um número de versão desta Licença que se aplique a ela e a "qualquer versão posterior", você terá a opção de seguir os termos e condições tanto daquela versão como de qualquer versão posterior publicada pela Free Software Foundation. Se o Programa não especificar um número de versão desta Licença, você poderá escolher qualquer versão já publicada pela Free Software Foundation.

11. Se você desejar incorporar partes do Programa em outros programas livres cujas condições de distribuição sejam diferentes, escreva ao autor solicitando a respectiva permissão. Para software cujos direitos autorais sejam da Free Software Foundation, escreva para ela; algumas vezes, abrimos exceções para isso. Nossa decisão será guiada pelos dois objetivos de preservar a condição livre de todos os derivados de nosso software livre e de promover o compartilhamento e reutilização de software, de modo geral.

EXCLUSÃO DE GARANTIA

11. COMO O PROGRAMA É LICENCIADO SEM CUSTO, NÃO HÁ NENHUMA GARANTIA PARA O PROGRAMA, NO LIMITE PERMITIDO PELA LEI APLICÁVEL. EXCETO QUANDO DE OUTRA FORMA ESTABELECIDO POR ESCRITO, OS TITULARES DOS DIREITOS AUTORAIS E/OU OUTRAS PARTES, FORNECEM O PROGRAMA "NO ESTADO EM QUE SE ENCONTRA", SEM NENHUMA GARANTIA DE QUALQUER TIPO, TANTO EXPRESSA COMO IMPLÍCITA, INCLUINDO, DENTRE OUTRAS, AS GARANTIAS IMPLÍCITAS DE COMERCIALIZAÇÃO E ADEQUAÇÃO A UMA FINALIDADE ESPECÍFICA. O RISCO INTEGRAL QUANTO À QUALIDADE E DESEMPENHO DO PROGRAMA É ASSUMIDO POR VOCÊ. CASO O PROGRAMA CONTENHA DEFEITOS, VOCÊ ARCARÁ COM OS CUSTOS DE TODOS OS SERVIÇOS, REPAROS OU CORREÇÕES NECESSÁRIAS.

12. EM NENHUMA CIRCUNSTÂNCIA, A MENOS QUE EXIGIDO PELA LEI APLICÁVEL OU ACORDADO POR ESCRITO, QUALQUER TITULAR DE DIREITOS AUTORAIS OU QUALQUER OUTRA PARTE QUE POSSA MODIFICAR E/OU REDISTRIBUIR O PROGRAMA, CONFORME PERMITIDO ACIMA, SERÁ RESPONSÁVEL PARA COM VOCÊ POR DANOS, INCLUINDO ENTRE OUTROS, QUAISQUER DANOS GERAIS, ESPECIAIS, FORTUITOS OU EMERGENTES, ADVINDOS DO USO OU IMPOSSIBILIDADE DE USO DO PROGRAMA (INCLUINDO, ENTRE OUTROS, PERDAS DE DADOS OU DADOS SENDO GERADOS DE FORMA IMPRECISA, PERDAS SOFRIDAS POR VOCÊ OU TERCEIROS OU A IMPOSSIBILIDADE DO PROGRAMA DE OPERAR COM QUAISQUER OUTROS PROGRAMAS), MESMO QUE ESSE TITULAR, OU OUTRA PARTE, TENHA SIDO ALERTADA SOBRE A POSSIBILIDADE DE OCORRÊNCIA DESSES DANOS.

FINAL DOS TERMOS E CONDIÇÕES

Como Aplicar Estes Termos para Seus Novos Programas

Se você desenvolver um programa novo e quiser que ele seja da maior utilidade possível para o público, o melhor caminho para obter isto é fazer dele um software livre, o qual qualquer pessoa pode redistribuir e modificar sob os presentes termos.

Para fazer isto, anexe as notificações seguintes ao programa. É mais seguro anexá-las ao começo de cada arquivo-fonte, de modo a transmitir do modo mais eficiente a exclusão de garantia; e cada arquivo deve ter ao menos a linha de "direitos autorais reservados" e uma indicação de onde a notificação completa se encontra.

<uma linha para informar o nome do programa e uma breve idéia do que ele faz.>

Direitos Autorais Reservados (c) <ano> <nome do autor>

Este programa é software livre; você pode redistribuí-lo e/ou modificá-lo sob os termos da Licença Pública Geral GNU conforme publicada pela Free Software Foundation; tanto a versão 2 da Licença, como (a seu critério) qualquer versão posterior.

Este programa é distribuído na expectativa de que seja útil, porém, SEM NENHUMA GARANTIA; nem mesmo a garantia implícita de COMERCIALIZABILIDADE OU ADEQUAÇÃO A UMA FINALIDADE ESPECÍFICA. Consulte a Licença Pública Geral do GNU para mais detalhes.

Você deve ter recebido uma cópia da Licença Pública Geral do GNU junto com este programa; se não, escreva para a Free Software Foundation, Inc., no endereço 59 Temple Street, Suite 330, Boston, MA 02111-1307 USA.

Inclua também informações sobre como contatar você por correio eletrônico e por meio postal.

Se o programa for interativo, faça com que produza uma pequena notificação como esta, quando for iniciado em um modo interativo:

Versão 69 do Gnomovision, Direitos Autorais Reservados (c) ano nome do autor. O Gnomovision NÃO POSSUI QUALQUER TIPO DE GARANTIA; para detalhes, digite 'show w'. Este é um software livre e você é bem-vindo para redistribuí-lo sob certas condições; digite 'show c' para detalhes. Os comandos hipotéticos `show w' e `show c' devem mostrar as partes apropriadas da Licença Pública Geral. Naturalmente, os comandos que você utilizar poderão ter outras denominações que não `show w' e `show c'; eles poderão até ser cliques do mouse ou itens de um menu - o que for adequado ao seu programa.

Você também pode solicitar a seu empregador (se você for um programador) ou sua instituição acadêmica, se for o caso, para assinar uma "renúncia de direitos autorais" sobre o programa, se necessário. Segue um exemplo; altere os nomes:

A Yoyodyne Ltda., neste ato, renuncia a todos eventuais direitos autorais sobre o programa `Gnomovision' (que realiza passagens em compiladores), escrito por James Hacker.

<Assinatura de Ty Coon>

1º de abril de 1989, Ty Coon, Presidente

Esta Licença Pública Geral não permite a incorporação do seu programa a programas proprietários. Se seu programa é uma biblioteca de sub-rotinas, você poderá considerar ser mais útil permitir a ligação de aplicações proprietárias à sua biblioteca. Se isso é o que você deseja fazer, utilize a Licença Pública Geral de Biblioteca do GNU, ao invés desta Licença.



Richard Matthew Stallman nasceu em Nova Iorque no dia 16 de março de 1953; frequentemente abreviado para “RMS” ele é considerado um grande hacker, fundador do movimento *free software* e do projeto *GNU*. Desde a metade dos anos 90, Stallman tem dedicado a maior parte do seu tempo ao ativismo político, defendendo o software livre. Ele também é o criador da fundação do Software livre (FSF) que prateje de forma legal os softwares registrados sobre a lei GPL.

Distribuição de GNU/Linux

Uma distribuição ou distro Linux é um sistema operacional Unix-Like que inclui o Kernel Linux e outros softwares de aplicação, formando um conjunto. Como grande parte destes softwares são aplicações que em primórdio foram criadas para o sistema operacional GNU é conveniente que recebam o nome de distribuição GNU/Linux.

Algumas distros são mantidas por organizações comerciais, como a Red Hat, Ubuntu, Suse e Mandriva, outras são mantidas pela comunidade como o Debian e o Gentoo. São mais de trezentas distribuições de Gnu/Linux embora cerca de vinte sejam conhecidas.

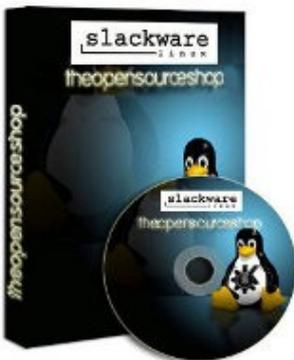
Vejamos um esquema de distribuição Gnu/Linux:



As distribuições mais conhecidas

As primeiras distribuições de GNU/Linux começam a existir no ano de 1992. Entre elas está a SSL (Softlanding Linux System) criada por Peter MacDonald que já vinha com o X Window System e suporte a TCP/IP e o MCC Interim Linux que é considerada a primeira distribuição oficial de Gnu/Linux.

Slackware Linux



Desenvolvida em meados de 1993 por Patrick Volkerding e mantida pelo próprio. Desde o início de sua criação Volkerding mantinha todo o projeto sozinho, contudo ao longo dos anos, acabou aceitando ajuda de alguns colaboradores a fim de ajudar no desenvolvimento da distribuição. É a distribuição mais velha em atividade. Ela é baseada na SLS Linux.

Distribuições mais conhecidas baseadas no Slackware:

- GoblinX
- Slax
- VectorLinux
- Zenwalk
- Kate OS
- Suse Linux

Debian Linux



O projeto Debian foi fundado por Ian Murdock em 16 de agosto de 1993. Ian pretendia que o Debian fosse uma distribuição criada abertamente. Tendo o mesmo princípio do sistema GNU o projeto de Ian foi apoiado pela comunidade FSF durante um ano (Novembro de 1994 a novembro de 1995). O Debian tem hoje um grupo de hackers do software livre desenvolvendo o sistema e uma comunidade sólida de usuários. Qualquer pessoa pode contribuir para o projeto Debian.

Distribuições mais conhecidas baseadas no Debian:

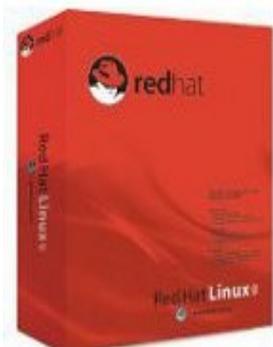
- Knoppix
 - Início do conceito Live-CD
- Kurumin
- Ubuntu
- Kubuntu
- Linux Mint

Suse Linux



Fundada na Alemanha em setembro de 1992 a companhia S.U.S.E era uma consultoria UNIX, a qual entre outras atividades lançava pacotes de softwares para o SLS e o Slackware Linux. SUSE é um acrônimo para a expressão Alemã: “Software Und System-Entwicklung” (Desenvolvimento de Software e de Sistemas). No começo o SUSE linux era uma tradução do Slackware Linux para o idioma Alemão. A sua primeira versão oficial foi lançada em 1994 com o nome de S.U.S.E Linux 1.0. Hoje o SUSE pertence a empresa Novell é um produto pago e para fins corporativos. Porém os desenvolvedores do SUSE apoiam uma iniciativa chamada “OpenSuse” que é um sistema aberto para usuários domésticos e desenvolvido por uma comunidade com apoio da Novell.

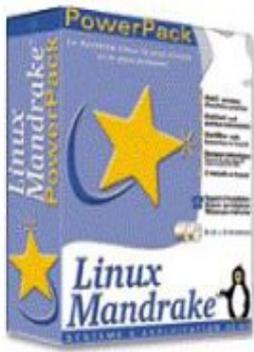
Red Hat Linux



Em 1993, Bob Young incorporava a ACC Corporation, um conjunto de empresas que vendiam softwares e acessórios Linux e Unix. Em 1994 Marc Ewing criou a sua própria distribuição de Linux, que chamou de Red Hat. Em 1994 a empresa de Bob Young compra o trabalho de Ewing e junta-se com ele para formar a empresa Red Hat Software tendo seu primeiro lançamento público em 1999.

O Red Hat é uma distribuição para fins corporativos e a sua empresa de mesmo nome recebeu neste ano de 2012 uma receita de US\$ 1 Bilhão. Semelhante a Novell a Red Hat apoia uma distribuição aberta para usuários domésticos chamada de Fedora.

Mandrake Linux



A empresa Francesa MandrakeSoft lança em 1998 uma distribuição baseada em Red Hat chamada de Mandrake Linux. O Foco do Mandrake era ser de fácil configuração para o usuário doméstico.

Em 2004 a empresa é ameaçada de processo pela Hearst Corporation por causa do nome "Mandrake" que era uma marca registrada. Neste ano a empresa compra a empresa Brasileira Conectiva que desenvolvia uma distribuição de mesmo nome e passa a se chamar Mandriva.

Atualmente a Mandriva esta passando por crises financeiras e seu futuro é incerto. Um fork do sistema já está sendo desenvolvido de nome: Mageia Linux.

Gentoo Linux



No final de 1999 Daniel Robbins lançou uma distribuição de nome Enoch Linux seu objetivo era ser a distribuição mais rápida de todas e para isto todos os pacotes eram compilados um a um, sintonizado com o Hardware de seu equipamento. Por ser de grande desempenho o nome foi trocado para Gentoo (Que é uma espécie de pinguim capaz de nadar mais rápido). Esta distribuição é considerada a mais rápida do planeta.

A primeira versão com o nome oficial foi lançada em 2002 (Gentoo Linux 1.0).

Ubuntu Linux

Ubuntu é um sistema operacional desenvolvido pela [comunidade](#), e é perfeito para laptops, desktops e servidores. Seja para uso em casa, escola ou no trabalho, o Ubuntu contém todas as ferramentas que você necessita, desde processador de texto e leitor de emails a servidores web e ferramentas de programação.

O Ubuntu é e sempre será **gratuito**. Você não paga por nenhum encargo de licença. Você pode baixar, usar e compartilhar com seus amigos e familiares, na escola ou no trabalho, sem pagar nada por isto.

Nós **lançamos uma nova versão para desktops e servidores a cada seis meses**. O que significa que você sempre terá as últimas versões dos maiores e melhores aplicativos de código aberto que o mundo tem a oferecer.

O Ubuntu é desenvolvido visando segurança. Você tem **atualizações de segurança gratuitas por pelo menos 18 meses** para desktops e servidores. Com a versão de Longo Tempo de Suporte (LTS) você tem três anos de suporte para desktops, e cinco anos para servidores. Não é cobrado nenhum valor pela versão LTS, bem como qualquer outra, nós disponibilizamos livremente o melhor que podemos oferecer para todos sob os mesmos termos. Atualizações para novas versões do Ubuntu são e sempre serão gratuitas.

Tudo o que você precisa em apenas um CD, que lhe proporciona um ambiente completo e funcional. Programas adicionais são disponibilizados através da Internet.

O instalador gráfico lhe permite **ter um sistema funcional de forma rápida e fácil**. Uma instalação padrão deve levar menos de 30 minutos.

Uma vez instalado, seu sistema está imediatamente **pronto para o uso**. Na versão desktop você tem um conjunto completo de aplicativos para produtividade, internet, imagens, jogos, entre outras ferramentas. Na versão servidor você tem tudo o que precisa para ter seu servidor funcional sem coisas desnecessárias.

O que a palavra ubuntu Significa?

Ubuntu é uma antiga palavra africana que significa algo como "Humanidade para os outros" ou ainda "Sou o que sou pelo que nós somos". A distribuição Ubuntu traz o espírito desta palavra para o mundo do software livre.

"Uma pessoa com Ubuntu está aberta e disponível aos outros, assegurada pelos outros, não sente intimidada que os outros sejam capazes e bons, para ele ou ela ter própria auto-confiança que vem do conhecimento que ele ou ela tem o seu próprio lugar no grande todo."

-- Arcebispo Desmond Tutu em Nenhum Futuro Sem Perdão (No Future Without Forgiveness).

Instalando uma Distribuição de GNU/Linux

*"A prisão não são as grades, e a liberdade não é a rua;
existem homens presos na rua e livres na prisão.
É uma questão de consciência."*

Mahatma Gandhi

Trisquel GNU/Linux - Por quê ele?

Entre tantas distribuições disponíveis vamos abordar neste livro o Trisquel Linux. Alguns conhecidos até argumentaram comigo que era interessante abordar o Ubuntu Linux que é a distribuição mais comentada do momento. Mas optei pelo Trisquel por alguns motivos a serem considerados:



Pensando sempre na usabilidade do usuário doméstico diversas distribuições como o Ubuntu que é da empresa Canonical carregam código proprietário em suas configurações. Estes códigos estão no Kernel e em programas aplicativos. As empresas que fabricam Hardware como a Nvidia fabricam pacotes de “drivers” e estes são incorporados nas diversas distribuições de GNU/Linux como o Ubuntu. O que não torna o sistema 100% Software Livre. Também é adicionados plugins e codecs como o suporte ao flash (Proprietário da Adobe) e ao java (Proprietário da Oracle).

Para muitos isto ajuda no dia a dia e torna o Linux mais democrático. No entanto, na minha visão, estamos viciando as pessoas a usarem softwares proprietários que estão cada vez mais presentes no corpo das distribuições “Ditas para usuários iniciantes” o que vai contra a filosofia de criação do sistema GNU e do próprio Kernel Linux.

A distribuição de GNU/Linux Trisquel é baseada no Ubuntu Linux porém com um grande diferencial, todos os softwares são livres.

O Kernel foi compilado excluindo todos os módulos (drivers) proprietários (Linux-libre) e apesar da liberdade do usuário em poder instalar programas proprietários por padrão somente softwares livres são compilados junto com a distribuição.

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.

Originalmente foi baseada no Debian, mas em 2008 começou a utilizar pacotes do repositório do Ubuntu, porém com todo o software proprietário eliminado.

Em dezembro de 2008, Trisquel Gnu/Linux foi incluído pela Free Software Foundation em uma lista que mostra as distribuições 100% Livres.

Você pode conhecer mais sobre distribuições 100% livres acessando:

<http://www.gnu.org/distros/free-distros.html>

Como é feito o Trisquel?

Como já mencionei o Trisquel é baseado no Ubuntu, então o primeiro processo de criação da distribuição é fazer um espelho local do Ubuntu e limpá-lo. O espelho é atualizado todos os dias usando scripts que possuem uma grande lista negra de software não-livre os quais são excluídos.

Após esta limpeza o espelho é liberado para uso. Alguns pacotes são modificados pela equipe Trisquel você pode saber mais sobre isto seguindo este link:

http://trisquel.info/wiki/package-helpers&usg=ALkJrhbinttQR9CtLOvS_yzCpFCItCA9w

Um Kernel totalmente Livre

Para fornecer um kernel totalmente livre o Trisquel usa o Kernel do projeto “Linux-Libre” que é apoiado pela FSF e por Richard Stallman.

O Kernel desenvolvido e distribuído por Linus Torvalds contém software não-livre, ou seja, softwares que não respeitam as liberdades essenciais do software aberto e induz você a instalar softwares não-livres adicionais. GNU Linux-libre é um projeto para manter e publicar distribuições 100% livre de linux.



Freetz, irmão mais velho do mascote do Kernel-libre.

Para mais informações acesse: <http://www.fsfla.org/svnwiki/selibre/linux-libre/>

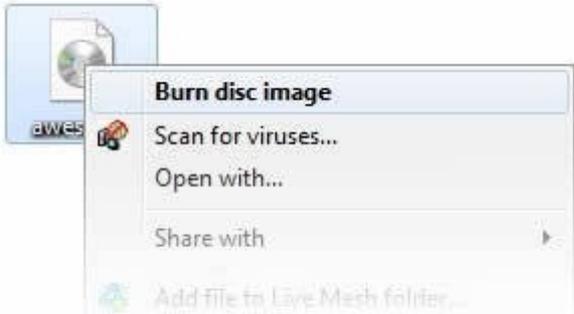
Instalando o Trisquel GNU/Linux

O primeiro passo é fazer o download da imagem do Trisquel GNU/Linux no site: <http://trisquel.info/en/download>

A versão atual até o término desta documentação é a 5.5 STS Brigantia. Escolha a versão do seu equipamento (32 bits ou 64 bits) e faça o download.

Gravar corretamente a imagem de CD

O Trisquel GNU/Linux é distribuído em imagem (.ISO) se você estiver no sistema Microsoft Windows 7, clique com o botão direito e escolha a opção “Gravar imagem de disco”, exemplo:

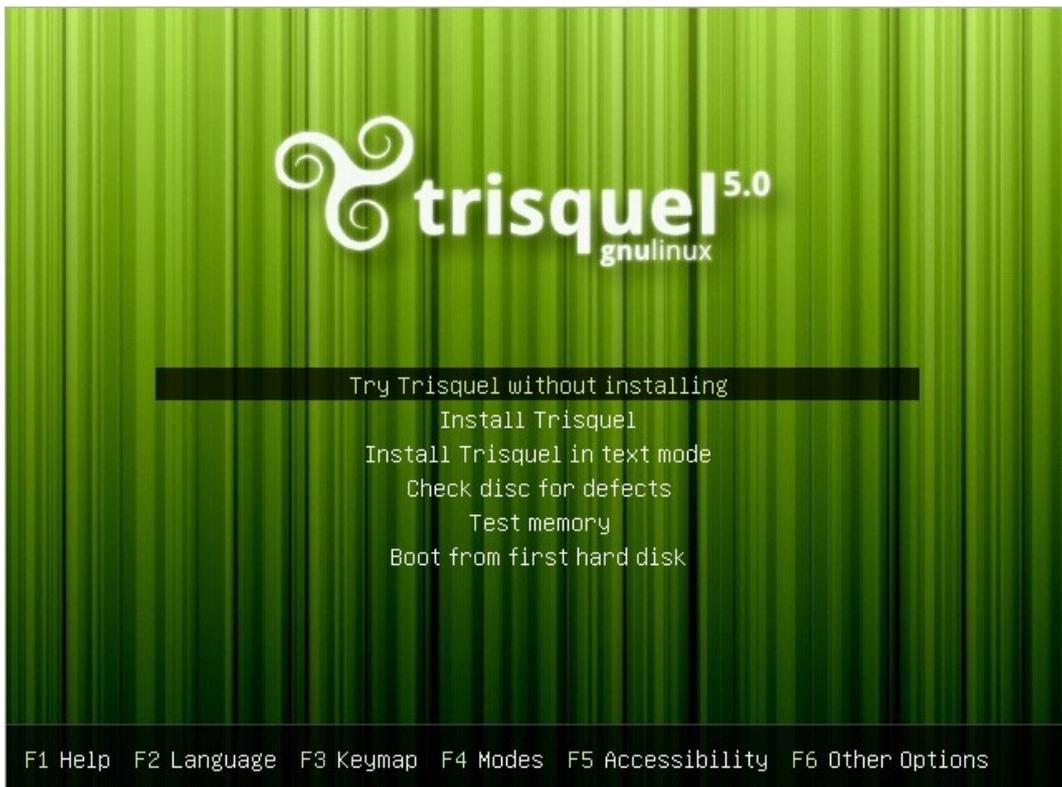


Coloque o CD ou DVD virgem e clique em queimar a imagem, exemplo:

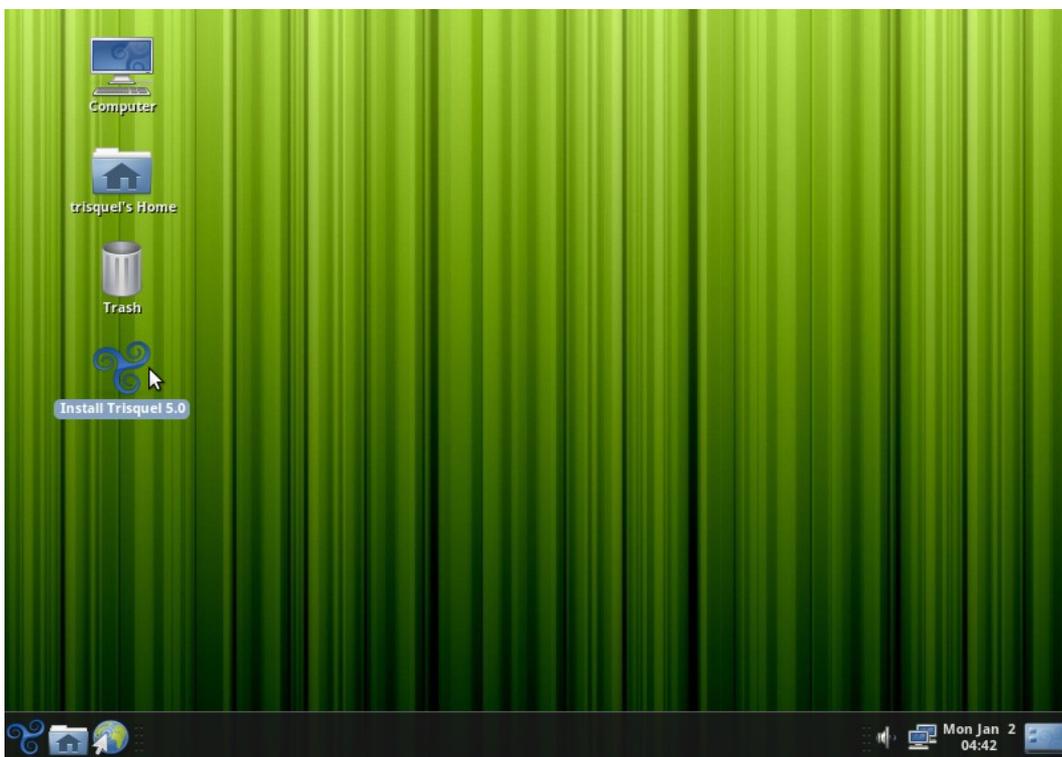


Agora que o CD/DVD foi gravado configure sua BIOS para dar início pelo CD/DVD e deixe o disco no drive. Na primeira tela você tem a opção de “Testar o Trisquel sem instalar” - “Try Trisquel without installing”.

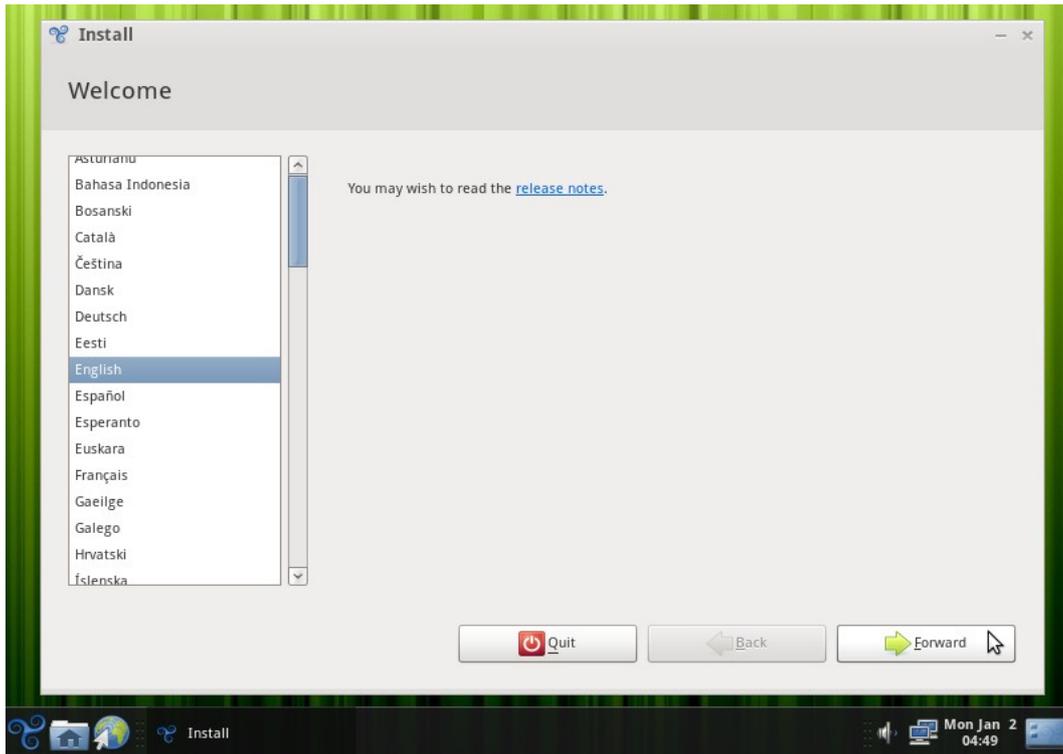
Obs: Este processo de instalação é baseado na versão 5.0 que não muda em relação a versão 5.5.



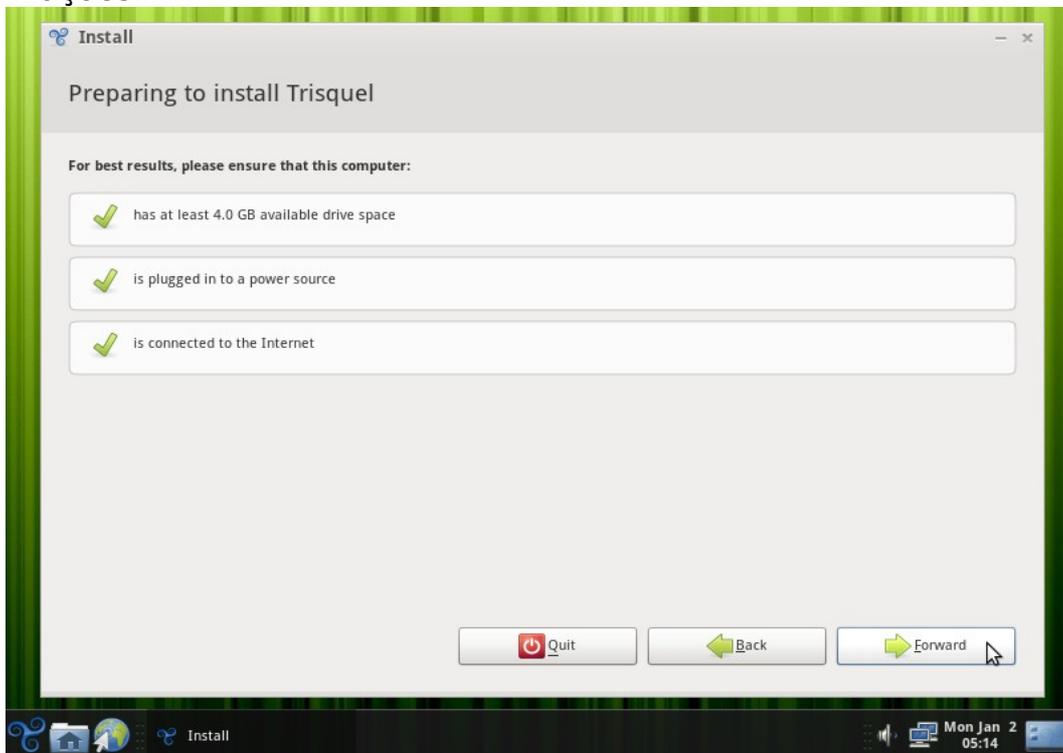
Após iniciar o sistema é só clicar no ícone “Install Trisquel” na área de trabalho:



Agora o processo é muito intuitivo. Escolha o seu idioma (português - Brasil) caso esteja com a internet ativa durante a instalação os pacotes de tradução serão baixados e instalados. Caso sua conexão ainda não esteja configurada, os pacotes serão baixados e instalados posteriormente.

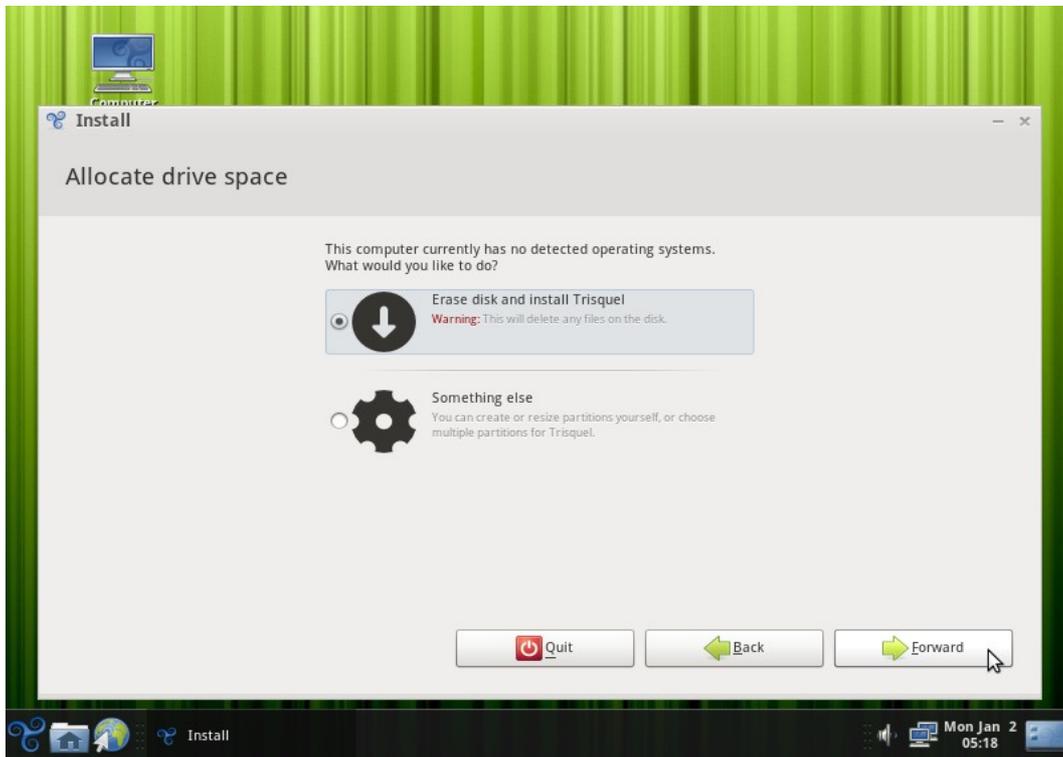


Nesta etapa o sistema verifica se você está conectado a internet e no caso de notebook se está instalado em um fonte de energia externa. Caso não esteja conectado não se preocupe, após instalar todo o sistema é fácil proceder com as atualizações.

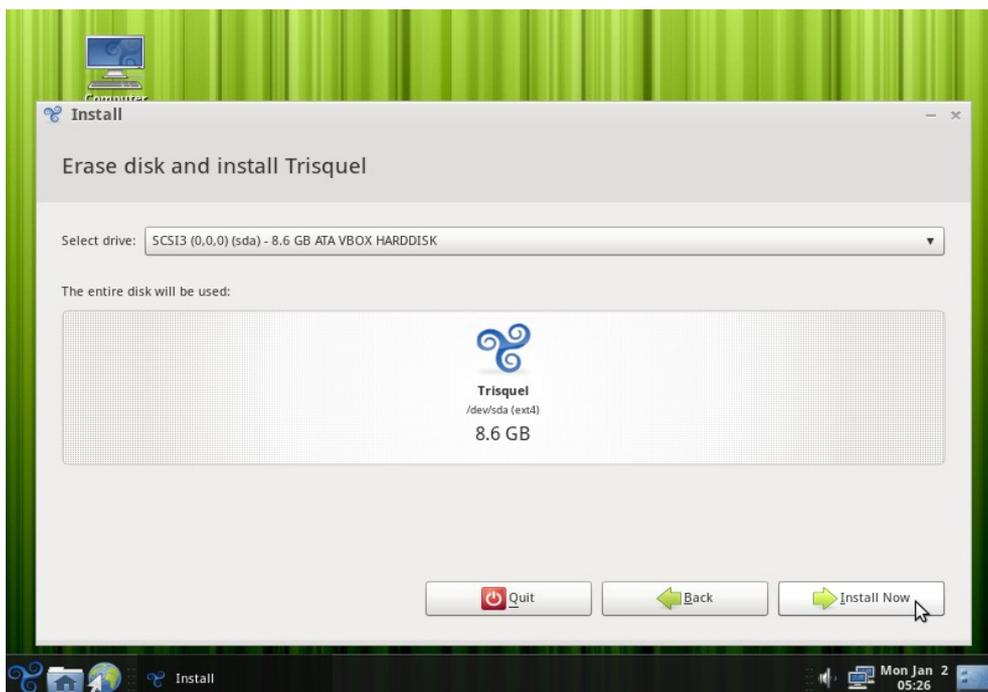


Agora é a etapa que mais exige atenção “Particionamento de disco”. Eu sugiro que você instale usando o disco todo (**Isto significa apagar todos os seus dados e instalar o Trisquel**) com o tempo e experiências obtidas você saberá realizar particionamentos personalizados.

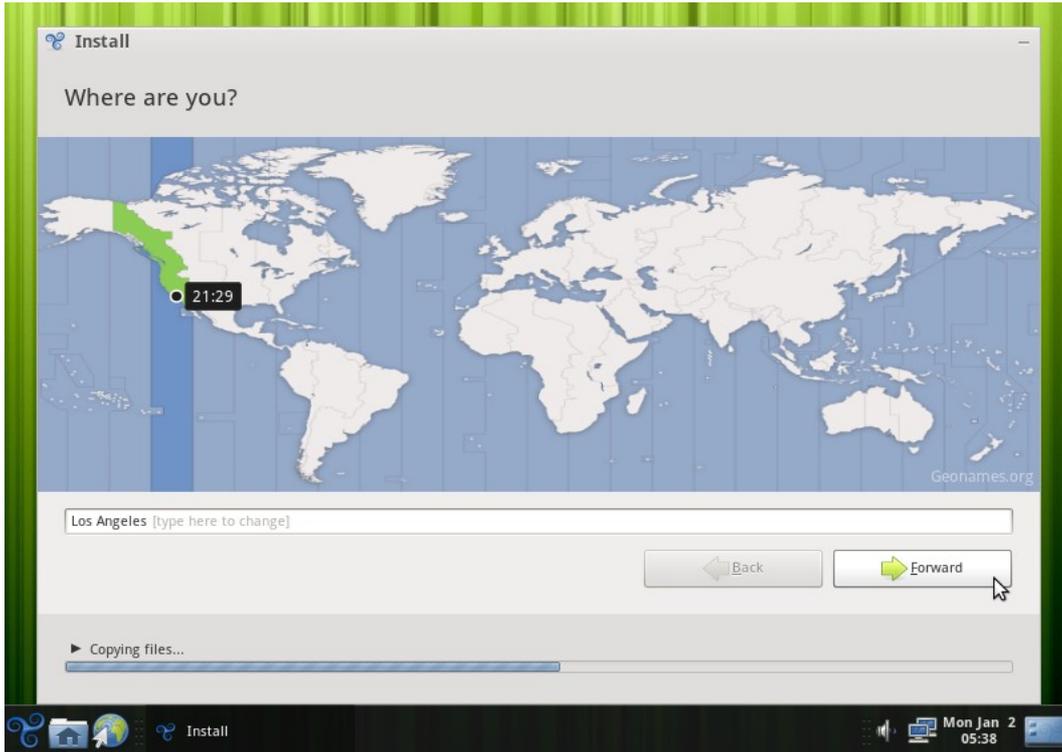
Caso não possa apagar seus dados e até mesmo outros sistemas operacionais que estejam instalados, recomendo que leia nosso capítulo sobre “Máquinas virtuais”.



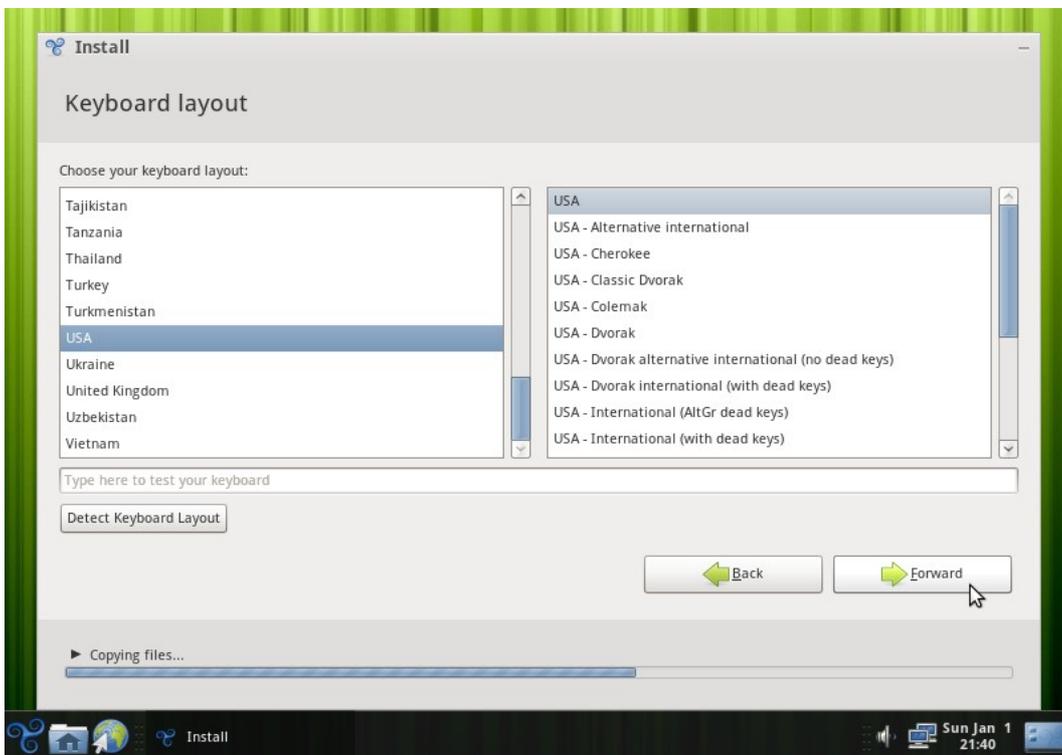
Agora basta avançar e clicar em “Install Now”:



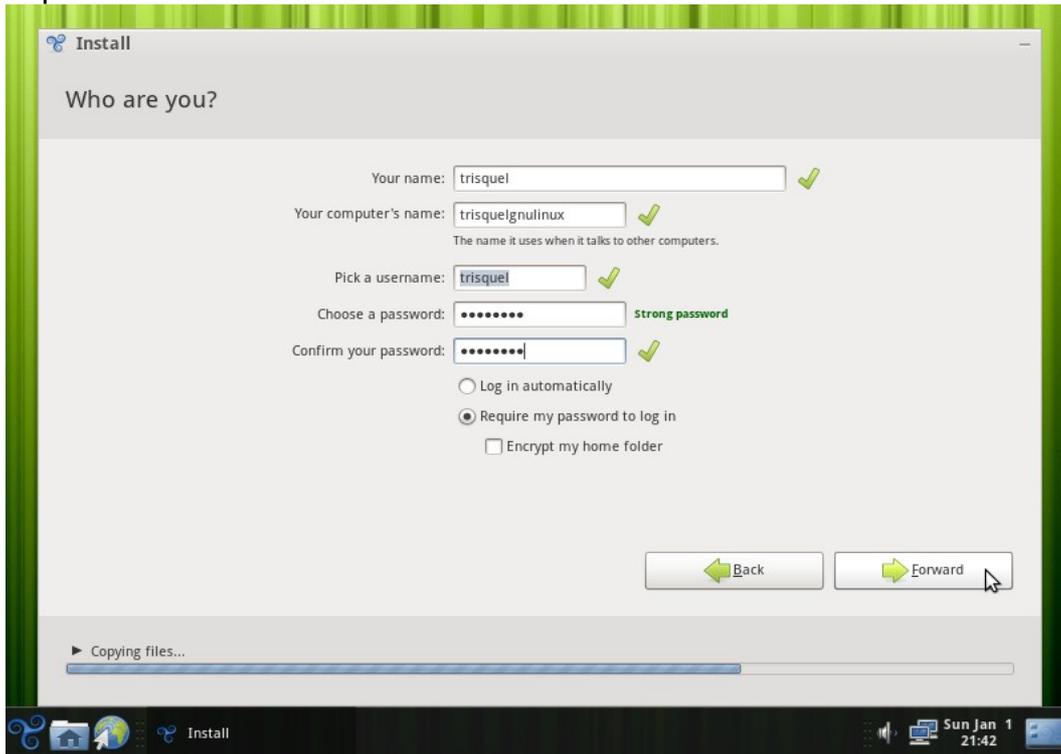
Selecione sua localidade:



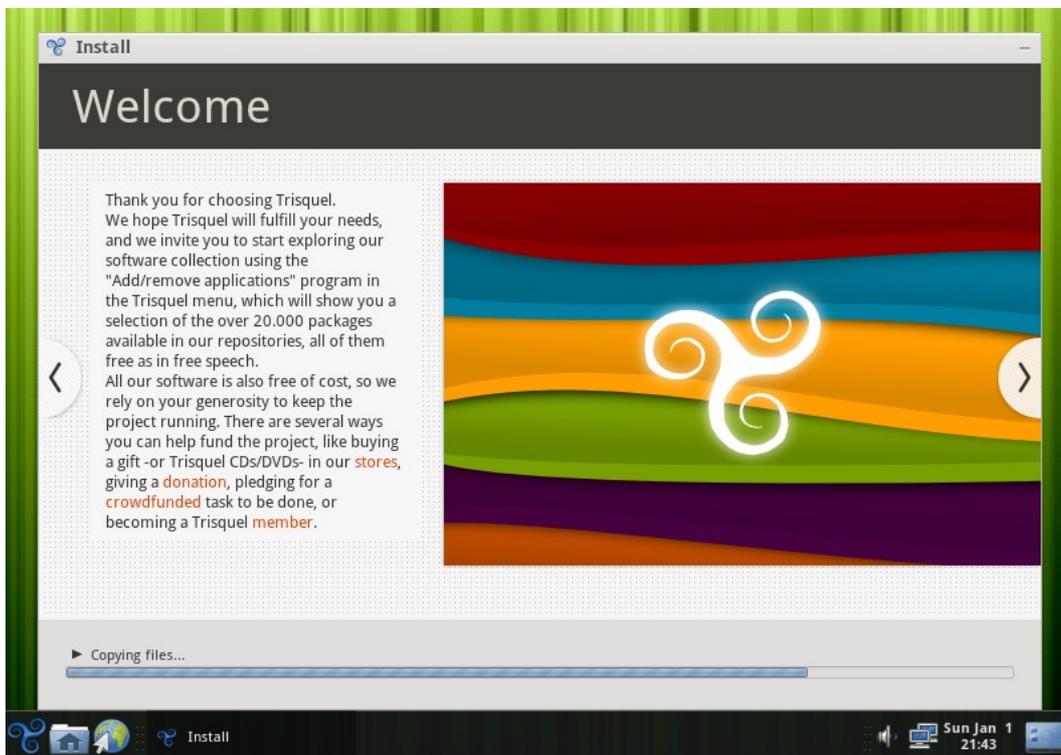
Selecione o layout do teclado (Use a área de teste para checar a configuração).



Nesta etapa você deve definir os dados do seu usuário.



Basta aguardar a cópia dos arquivos:

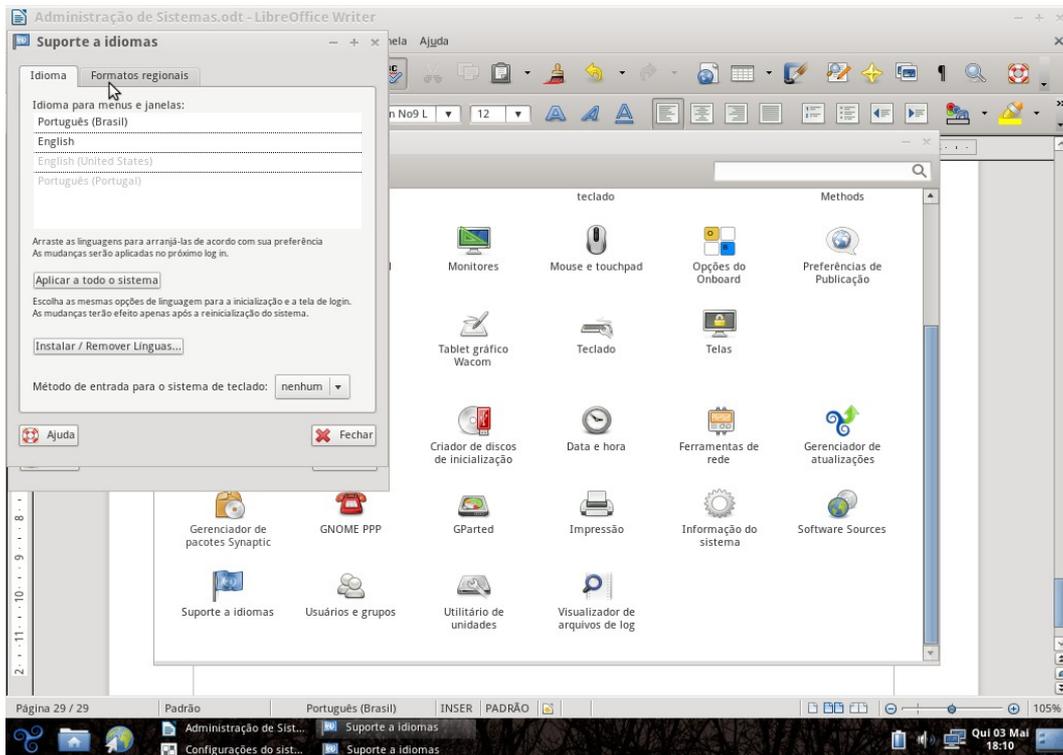


Faça a reinicialização do sistema. Retire o disco do Drive e inicie pelo seu disco rígido.

Após a instalação

O Trisquel já oferece os recursos básicos para um sistema de uso doméstico. Vamos conhecer alguns pontos básicos do sistema.

Eu recomendo que você ative sua conexão com a internet e verifique se os pacotes de idioma estão corretamente instalados. Para isto clique no menu de início - Configuração do Sistema - Suporte a idiomas. Caso não esteja correto ele vai pedir para você instalar os pacotes, após a instalação conseguirá abrir a aplicação:



Com o suporte de idiomas correto, vamos conhecer alguns aplicativos nativos desta distribuição.

Navegador Abrowser

Clicando no menu início - Internet - Navegador WEB Abrowser

O navegador Abrowser é um fork do Firefox. Isto ocorre porquê apesar de o Mozilla Firefox ser um software livre seu nome e seu logotipo são marcas registradas. Como o Trisquel tem como base a liberdade dos softwares o sistema recebeu um novo nome livre.

Add-ons que não sejam livres também não estão disponíveis por padrão no Abrowser.

O Abrowser usa o Gnash (Player de flash livre criado para GNU) para executar os sites e vídeos em Flash Player da WEB. Não tive problema algum em assistir a vídeos no youtube com este plugin. Ao abrir o vídeo abaixo da área de visualização você terá um botão "Click to play" - Ao clicá-lo o vídeo inicia normalmente.

Aconselho você a utilizar os Add-ons que são distribuídos pelos desenvolvedores que colaboram com o Trisquel. Clique em Abrowser - Complementos - Adicionar - Procure pelos Add-ons.

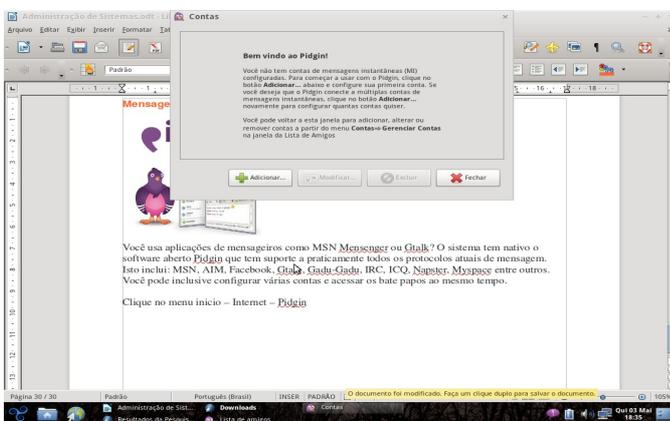
Mensageiro Pidgin



Você usa aplicações de mensageiros como MSN Messenger ou Gtalk? O sistema tem nativo o software aberto Pidgin que tem suporte a praticamente todos os protocolos atuais de mensagem. Isto inclui: MSN, AIM, Facebook, Gtalk, Gadu-Gadu, IRC, ICQ, Napster, Myspace entre outros. Você pode inclusive configurar várias contas e acessar os bate papos ao mesmo tempo.

Clique no menu inicio - Internet - Pidgin

Na primeira tela clique em Adicionar:



Escolha o protocolo da sua conta (MSN, GTALK entre outros). Coloque seu nome de usuário e senha, clique em adicionar. Caso queira adicionar mais contas clique no menu superior - Contas - Gerenciar Contas - Adicionar.

Suíte Escritório Broffice

A suíte escritório Broffice é a versão nacional do LibreOffice. Quando você instala os pacotes de idioma ele é instalado para que você tenha o aplicativo em nosso idioma com correção ortográfica.

LibreOffice ou OpenOffice?

A suíte de escritório mais popular do sistema Gnu/Linux é o OpenOffice que foi desenvolvido pela empresa Sun Microsystems, a mesma do Java. A Sun tinha uma suíte proprietária chamada StarOffice que não conseguia avançar no mercado por causa do Microsoft Office então para não abandonar o projeto eles deram origem a suíte de escritório livre.

Além desta aplicação a Sun apoiava alguns softwares de código aberto como o OpenSolaris. Infelizmente a empresa foi vendida para a Oracle que não é muito amiga do código aberto e que logo após a aquisição informou que iria descontinuar o OpenSolaris. A comunidade de usuários deste sistema se reuniu e criou o OpenIndiana.

Com receio do mesmo acontecer com o OpenOffice a melhor suíte escritório livre do momento, alguns hackers que eram desenvolvedores da Sun, realizaram um fork do software o qual chamaram de Libre-office. No Brasil o OpenOffice era chamado de BrOffice por questões legais já que uma empresa reclamou do nome dizendo ser sua marca registrada. O software continua se chamando Broffice porém agora com o código do LibreOffice e não mais do OpenOffice.

Neste livro não vou abordar o uso dos aplicativos de escritório. Porém recomendo o link com apostilas gratuitas sobre o uso das aplicações:

http://www.broffice.org/?q=apostila_comunidade

Vale saber para quem está migrando do Microsoft Windows:

Tipo de Software	Trisquel Gnu/Linux	Microsoft Windows
Editor de Texto	Broffice Writer	Microsoft Office Word
Planilha Eletrônica	Broffice Calc	Microsoft Office Calc
Apresentador de Slides	Broffice Impress	Microsoft Office Power Point

Multimídia

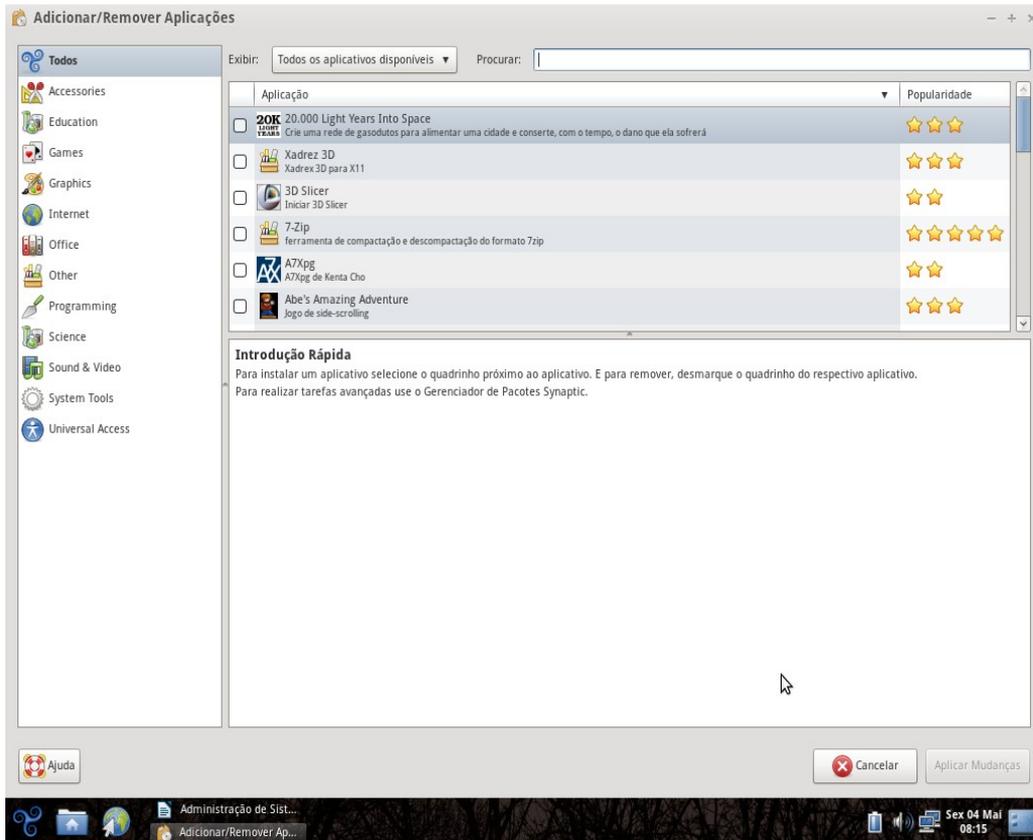
A distribuição Trisquel para o Brasil já possui suporte aos arquivos .mp3 e diversos formatos de vídeo como .rmvb (Real Player), AVI, entre outros. Todos os codecs necessários para estes arquivos funcionarem no Trisquel são de código fonte aberto, apesar de estes arquivos estarem compactados em formatos fechados.

Eu recomendo que você transforme estes arquivos em um formato livre.

Clicando no menu início - Multimídia - Oggconvert você terá acesso à uma aplicação de conversão de arquivos, muito simples e intuitiva de se utilizar.

Instalando um pacote adicional

Em sistemas operacionais Gnu/Linux temos o costume de chamar de pacote de programas os softwares aplicativos. Para instalar novos pacotes de programas clique no menu início tecla de atalho (Alt + f1) - Adicionar/Remover Aplicações:



O aplicativo é muito simples. Basta você digitar o nome do programa na guia "Procurar" clicar sobre ele marcando-o e depois na opção aplicar mudanças. O programa será instalado e um ícone sera adicionado no seu menu de início. Você também pode instalar aplicativos via linha de comando. Porém vamos falar sobre isto no próximo capítulo: Comandos GNU e Unix.

Comparativo de aplicações Gnu/Linux e Microsoft Windows:

Tipo de Programa	Trisquel Gnu/Linux	Microsoft Windows	Observação
Editor HTML	Kompozer	Adobe Dreamweaver	Alguns Webdesigners reclamam por não possuir tantos recursos quanto o software proprietário da Adobe. Eu utilizo este software e na minha opinião ele é mais leve e uma opção viável para quem não quer gastar uma fortuna com software.
Tipo de Programa	Trisquel Gnu/Linux	Microsoft Windows	Observação
Compartilhamento P2P	Qbitorrent	Ares	Esta aplicação é simples e intuitiva. A rede é imensa e quase sempre você encontra o que procura.
Editor de Imagens	GIMP	Adobe Photoshop	Já assisti muita palestra de designer gráfico que prefere o Gimp ao seu concorrente.
Media Player	Smplayer	Windows Media Player	Este player é leve e com suporte a diversos formatos de áudio e vídeo.
Desenvolvimento	Gambas	Visual Basic	Mais rápido que o VB
Softwares CAD	Qcad	AutoCad	É livre e gratuito. Tem a limitação de só suportar o formato 2D.
Gravação de CD/DVD	K3B	Nero	Muito superior ao seu concorrente proprietário.

LPIC-1 exame 101 - CompTIA linux+

*"Sério, eu não tenho por meta destruir a Microsoft.
Este será um efeito colateral completamente involuntário."
Linus Torvalds*

Sobre a LPI

A LPI surgiu em 1999 com o objetivo de criar uma certificação independente da distribuição de Linux, de forma que seus profissionais certificados estão aptos à trabalhar com qualquer versão de Linux. O LPI conta com três níveis de certificação: LPIC-1, LPIC-2 e LPIC-3. Cada nível pretende certificar um profissional apto a desempenhar tarefas que devam ser executadas com um crescente grau de dificuldade e complexidade.

Os tópicos da prova da certificação LPIC-1 são:

1. Arquitetura de sistema (tópico 101) - peso 8
2. Instalação do Linux e gerenciamento de pacotes (tópico 102) - peso 11
3. Comandos GNU e UNIX (tópico 103) - peso 26
4. Dispositivos e sistemas de arquivos (tópico 104) - peso 15

O “peso” representa o grau de importância no exame. A somatória do exame 101 é 60. O Exame 101 que abordamos neste livro é a primeira prova requerida para a certificação LPIC Nível 1. Nesta prova são abordados 4 tópicos como descritos acima.

O Exame 101 tem 90 minutos de duração e aproximadamente 60 questões. Cerca de 75% das questões são de múltipla escolha onde existe apenas uma opção correta.

Outros 10% das questões são de múltipla escolha com mais de uma opção correta.

Este tipo de questão é mais difícil porque apenas uma opção incorreta invalida toda a questão. Após a prova da LPI 101 é necessário realizar a prova 102 para que você obtenha a LPIC nível 1.

Como marcar a prova

Acesse [HTTP://www.lpi.org](http://www.lpi.org) e faça seu cadastro para obter um LPI-ID. Escolha agora um centro Pearson VUE através do endereço: [HTTP://www.pearsonvue.com](http://www.pearsonvue.com) Selecione Tecnologia da Informação Linux Professional Institute e depois Brasil.

Então escolha sua cidade e estado.

Ajuda na WEB

Muita informação sobre a prova está na WEB, recomendo nossa rede social: www.lcenter.com.br (Grupo LPI) e o site: [HTTP://www.tdpl.org/](http://www.tdpl.org/). - Simulado: <http://www.linux-praxis.de/lpim/lpi.html> português: <http://lpibrasil.com.br/lime4linux/index.php/survey/index>

Arquitetura de Hardware - “Peso 8 no exame”

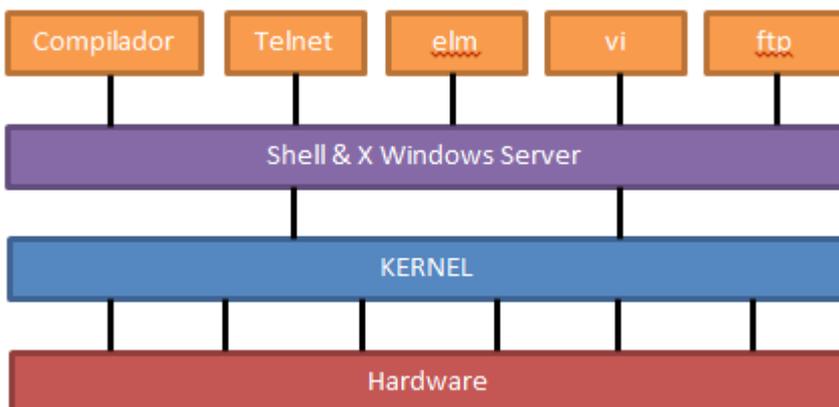
De forma simplista os sistemas operacionais podem ser entendidos como programas especiais que controlam todos os recursos do computador e fornecem toda a base para a execução de outros programas e aplicações. No início quando não existiam os sistemas operacionais, os aplicativos tinham de cuidar de tudo. Era preciso que o programador tivesse conhecimento da plataforma em que seus programas iriam rodar. Para facilitar o desenvolvimento foi criada uma camada de software para lidar com o hardware e fornecer uma interface amigável para os programadores interagirem com a máquina.

Os primeiros sistemas operacionais eram simples e monoprogramados, com um único programa de usuário em execução. A necessidade de mais programas serem executados ao mesmo tempo fez com que eles evoluíssem a multiprogramação. Suportando a execução de múltiplos programas de usuários de forma concorrente.

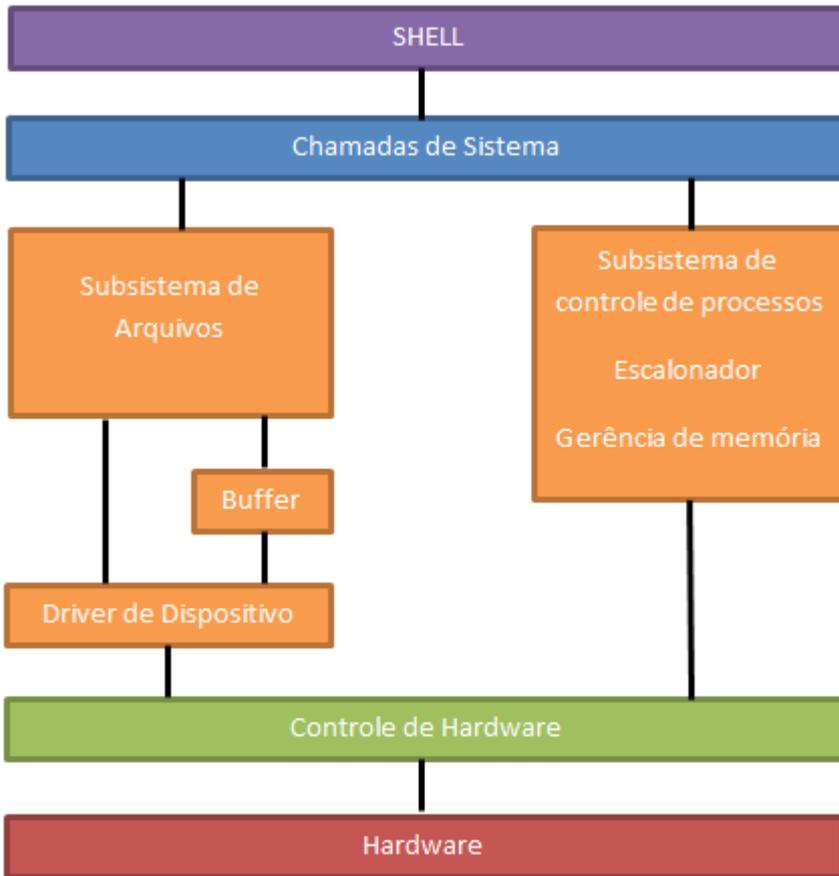
Kernel dos sistemas operacionais

Sem dúvida o Kernel é a parte mais importante de qualquer sistema operacional sendo considerado o núcleo do sistema. Ele é responsável pelo gerenciamento das memórias, processos, sub-sistemas de arquivos, suporte aos dispositivos e periféricos conectados ao computador. Os núcleos dos sistemas operacionais podem ser implementados de duas formas básicas Kernel Monolítico e o microkernel.

Kernel monolítico é estruturado em um único arquivo binário, um único processo que executa inteiramente em modo protegido. Ele possui desempenho superior na passagem de mensagens, mas apresenta inúmeras desvantagens como a dificuldade de alterações no núcleo e o desperdício de recursos, pois os drivers de dispositivos permanecem constantemente em memória, mesmo quando os dispositivos não estão sendo utilizados.



Kernel monolítico



MicroKernel

No microkernel apenas uma pequena parte do núcleo executa em modo protegido para acessar diretamente o hardware, como também é responsável pela comunicação entre processos e gerência de memória.

O restante do sistema roda em modo usuário, uma vez que executa tarefas que não necessitam acessar diretamente o hardware, e seus serviços clássicos são assegurados por processos servidores.

Os recursos do sistema são acessados através de um protocolo cliente/servidor, e para incluir um novo serviço basta acrescentar um novo servidor. O microkernel possui um desempenho inferior ao modelo monolítico. Mas podem-se alterar suas partes sem a necessidade de reiniciar a máquina permitindo a expansão para um sistema distribuído de forma mais fácil.

Interrupções

Para que tudo funcione é necessário que os dispositivos estejam alocando recursos do computador. Estes recursos podem ser portas de entrada e saída, requisições de interrupção (IRQ) e acesso direto a memória (DMA).

“As portas de entrada e saída são endereços de memória reservados no microprocessador para os dispositivos realizarem entrada e saída de informações”.

Os dispositivos podem usar mais de uma porta de entrada/saída ou uma faixa de endereços. Por exemplo, uma placa de som padrão usa as portas 0x220, 0x330, 0x388. Cada dispositivo possui um endereço de porta único, que não pode ser compartilhado entre outros dispositivos.

O acesso direto à memória (DMA) é usado para permitir a transferência de dados entre dispositivos e a memória sem a intervenção do processador. Este acesso é feito através de canais (Channels).

A maioria dos computadores tem dois controladores de DMA. O primeiro controla os canais 0,1,2,3 e o segundo, os canais 4,5,6,7 totalizando 8 canais.

“As requisições de interrupção (IRQ) são chamadas que os dispositivos podem fazer para requerer atenção especial ao processador.”

A maioria dos computadores oferece apenas 16 interrupções de hardware. As IRQs podem ser compartilhadas.

As interrupções podem ser geradas pelos seguintes eventos:

- Geradas pelo programa aplicação que são implementadas com o uso de instruções especiais.
- Por erro, como na divisão por zero, referência à memória fora do espaço permitido
- Por tempo, como no escalonamento
- Falha de hardware
- Por eventos de entrada e saída de dados sinalizando o final de operação ou condição de erro.

BIOS

A BIOS (Basic Input Output System) é um software especial gravado em memória flash situada na placa-mãe do computador. Este software é responsável por realizar todos os testes de hardware e reconhecer os dispositivos ligados à placa mãe.

Nem sempre as placas de vídeo, som, rede SCSI, placas aceleradores, dentre outras, são reconhecidas e configuradas automaticamente pela BIOS. Alguns dispositivos requerem configuração manual.

Portas de entrada/saída (Input/Output)

Os endereços de E/S em uso no sistema podem ser visualizados com o comando:

```
cat /proc/ioports
```

Requisições de interrupção (IRQ)

As interrupções do sistema podem ser visualizadas no kernel com o comando:

```
cat /proc/interrupts
```

Acesso direto a memória (DMA)

Os canais de DMA em uso no sistema podem ser visualizados com o comando:

```
cat /proc/dma
```

Nome do dispositivo	Windows	Porta E/S	DMA	IRQ
ttys0	com1	0x3f8		4
ttys1	com2	0x2f8		3
ttys2	com3	0x3e8		4
ttys3	com4	0x2e8		3
lp0	Lpt1	0x378		7
lp1	Lpt2	0x278		5
/dev/hda1	C:	0x1f0		14
/dev/fd0	A:	0x3f0	2	16

Dispositivos PCI

Durante o boot, o kernel detecta os dispositivos PCI conectados no micro. A lista das placas de expansão poderá ser encontrada no arquivo `/proc/pci`

Nem todas as distribuições possui este arquivo, neste caso podemos visualizar as informações sobre os barramentos PCI do sistema com o comando:

```
lspci
```

Opções comuns para “lspci”:

- v Mostra informações detalhadas sobre os dispositivos
- vv Mostra ainda mais informações sobre os dispositivos
- n Mostra os códigos que são fornecidos pelos fabricantes
- x Mostra os primeiros 64 bytes da configuração PCI em Hexadecimal
- xxx Mostra toda a configuração PCI em Hexadecimal

lsusb

O comando “lsusb” é utilizado para visualizar todos os dispositivos USB conectados à este tipo de barramento.

A opção -v (Mostra detalhes sobre o dispositivo)

```
lsusb -v
```

O kernel 2.6 do Linux possui um robusto sistema de reconhecimento de hardware. Este reconhecimento foi implementado principalmente através do sysfs,udev e do d-bus.

sysfs

O sysfs é um recurso do kernel 2.6 para exportar informações úteis sobre o sistema para os processos (programas) dos usuários através de um sistema de arquivo em memória. Normalmente o sysfs é montado no diretório: `/sys` Os principais diretórios de `/sys` são:

block, bus,class,devices,firmware,fs,kernel,module e power

BLOCK

Este diretório contém subdiretórios para cada sistema de arquivos de blocos (discos rígidos principalmente) existentes no sistema. Dentro dos diretórios dos dispositivos, há diversos arquivos que indicam entre outras coisas, o tamanho do dispositivo, status etc.

BUS

Este diretório contém subdiretórios para cada tipo de barramento suportado pelo Kernel.

E cada subdiretório contém os diretórios devices e drivers.

Devices: Lista de todos os dispositivos encontrados que são do tipo de barramento indicado

Drivers: Contém os drivers de dispositivos daquele tipo de barramento

CLASS

O diretório class contém representações de cada classe de dispositivo que está registrado no kernel. As classes indicam o tipo de dispositivo que elas representam. Por exemplo, classe de impressoras, de discos entre outras.

Cada subdiretório terá um diretório sobre o tipo da classe, exemplo:

```
/sys/class/net/eth0
```

DEVICES

Contém a hierarquia global dos dispositivos encontrados e suportados pelo Kernel.

Sua hierarquia obedece a organização dos dispositivos em qual tipo de conexão elétrica eles estão conectados.

Dispositivos de plataforma (platform) – Periféricos inerentes à plataforma de processamento e barramento do hardware, exemplo: Controladores seriais e paralelo

Dispositivos do sistema (system) – Periféricos que não se encaixam em qualquer outra classificação. Por exemplo: CPUs, APICS, temporizadores.

FIRMWARE

Este diretório contém interfaces para verificar e atribuir objetos específicos de firmware. Firmwares são códigos executados durante o processo de carga da máquina. Como software da BIOS.

MODULE

Este diretório contém subdiretórios com todos os módulos de kernel carregados para a memória. Os módulos são pedaços de programas que podem ou não fazer parte do kernel.

Exemplo:

Se você não utiliza com frequência o drive de disquete, pode optar em apenas carregar o módulo que habilita suporte à disquete quando for fazer uso deste tipo de mídia.

Este tipo de arquitetura “MODULAR” permite que o kernel fique mais leve, ocupando menos memória e processamento.

POWER

Este diretório representa o subsistema de gerenciamento de energia.

UDEV

O gerenciamento dinâmico de dispositivos (udev) é responsável por manter a estrutura do diretório /dev (diretório que contém os links que representam os dispositivos) de forma a permitir que somente os dispositivos conectados sejam listados na estrutura do /dev

Geralmente este gerenciador é executado na forma do programa chamado udevd em modo de servidor (daemon) que fica escutando os eventos do kernel para encontrar quando um dispositivo é removido ou adicionado ao sistema.

Configurando modems e placas de som

Embora não-essenciais, os modems e placas de som são hardwares de uso comum.

Modems

Um modem (palavra derivada de modular e demodular) é aquele conhecido dispositivo que modula um sinal digital para a forma de um sinal analógico capaz de transmitir informações através de linhas telefônicas.

Tipos de modems: Interno, Externo, Softmodem

O modem externo utiliza uma fonte de alimentação externa e é conectado ao micro através de um cabo serial padrão RS-232 em uma porta serial `/dev/ttys0` (COM1) ou `/dev/ttys1` (COM2).

O modem interno é conectado à um barramento ISA ou PCI dependendo do modelo. Ele também utiliza uma porta serial `/dev/ttys2` ou `/dev/ttys3`. Do ponto de vista técnico, eles são iguais.

A outra variação de modem foi desenhada especialmente para funcionar no sistema operacional da Microsoft e foi apelidado de Winmodem, Softmodem ou HSP (Host Signal Processor) este tipo de modem utiliza muita carga da CPU e um software específico e por isto são baratos.

O suporte para Linux destes modems são limitados e depende de um driver específico.

Como acontece com qualquer placa de expansão, em particular aquelas configuradas manualmente, o usuário precisa ter cuidado com conflitos de recursos. Os modems não deverão causar muitas dificuldades, uma vez que são simples portas seriais. Entretanto, você precisará confirmar que os endereços de interrupção e de E/S corretos estão configurados para seu modem.

setserial

Este utilitário foi especificamente desenvolvido para configurar portas seriais no Linux. Não é padrão no Debian (Já que caiu em desuso à utilização de modems). Basicamente o comando funciona:

```
setserial /dev/ttys0 irq 4
```

Definimos neste comando à porta `/dev/ttys0` (COM1) o IRQ 4

Dispositivo de som

Quase todos os PCs, Laptops e desktops de hoje possuem um dispositivo de som. Felizmente, estão disponíveis drivers de som do Linux para a maioria dos chipsets de áudio.

Dependendo do modelo as placas de som podem utilizar barramentos ISA, PCI ou embutidas na própria placa mãe. Os recursos padrões utilizados para estes dispositivos são: IRQ 5, DMA 1, DMA 16 e as portas de entrada e saída: 0x220, 0x330 e 0x388.

O Linux suporta dois padrões de sistema de som: OSS (Open Sound System) e ALSA (Advanced Linux Sound Architecture)

Para utilizar o dispositivo de som, carregue o módulo da placa com o comando:

```
modprobe nome_do_módulo
```

Dica: Caso você não saiba o nome do módulo; poderá utilizar uma distro live-cd e rodar o comando:

```
lsmod
```

Para verificar o nome do módulo.

Depois de instanciado no Kernel o módulo do dispositivo de som, os recursos: /dev/audio, /dev/dsp e /dev/mixer estarão disponíveis.

O próximo passo será instalar um programa para controlar volume, exemplo: alsamixer

```
apt-get install alsamixer
```

Configurar dispositivo Não-IDE

Dispositivos SCSI (scuzzy)

Há dois tipos de dispositivos SCSI:

8 bit: 8 dispositivos incluindo o controlador.

16 bit: 16 dispositivos incluindo o controlador.

Dispositivos SCSI são identificados através de um conjunto de três números chamado **SCSI_ID**:

- 1- O Canal SCSI. Cada adaptador SCSI suporta um canal de dados no qual são anexados os dispositivos SCSI. São numerados a partir de zero.
- 2- O ID do dispositivo. A cada dispositivo são atribuído um número ID único alterável através de jumpers. A gama de IDs vai de 0 a 7 em controladores de 8 bit e de 0 a 15 em controladores de 16 bit. O ID do controlador costuma ser 7.
- 3- O número lógico da unidade (LUN). É usado para determinar diferentes dispositivos dentro de um mesmo alvo SCSI. Pode indicar uma partição em um disco ou um dispositivo de fita específico em um dispositivo multi-fita. Hoje não é muito utilizado pois adaptadores SCSI estão mais baratos e podem comportar mais alvos por barramento.

Todos dispositivos SCSI são listados em `/proc/scsi/scsi`:

```
cat /proc/scsi/scsi
```

O comando `scsi_info` usa as informações deste arquivo para mostrar o SCSI_ID e o modelo do dispositivo solicitado:

```
scsi_info /dev/scd0
```

Por padrão, o dispositivo SCSI de boot é o de ID 0, o que pode ser alterado na BIOS SCSI. Se existirem tanto dispositivos SCSI quanto IDE, a ordem do boot precisa ser especificada na BIOS da máquina.

Configurar Placas de Expansão para PCS

A configuração de placas de expansão diversas engloba os aspectos abordados nos objetivos anteriores, mais um conhecimento mais sólido sobre coldplug, hotplug e inspeção de hardware.

Em linhas gerais, **coldplug** significa a impossibilidade de se conectar dispositivo sem a necessidade de desligar a máquina. Exemplos de dispositivos coldplug são placas PCI, ISA e dispositivos IDE. Na maioria dos computadores, CPU e pentes de memória são coldplug. Porém, alguns servidores de alta performance suportam hotplug para esses componentes.

Hotplug é o sistema que permite conectar novos dispositivos à máquina em funcionamento e usá-los imediatamente, como no caso de dispositivos USB. O sistema Hotplug foi incorporado ao núcleo do modelo de driver do kernel 2.6, assim qualquer barramento ou classe pode disparar eventos hotplug quando um dispositivo é conectado ou desconectado. Assim que um dispositivo é conectado ou desconectado, o hotplug dispara um evento correspondente, geralmente trabalhando junto do sub-sistema udev, que atualiza os os arquivos de dispositivos em /dev.

O hotplug precisa estar liberado no kernel, através da opção CONFIG_HOTPLUG. Dessa forma, haverá o arquivo /proc/sys/kernel/hotplug contendo o caminho para o programa hotplug (normalmente em /sbin/hotplug). A ação tomada pelo hotplug dependerá do nome do agente passado pelo kernel (nomes de agentes podem ser "USB", "pci", "net", etc.). Para cada agente existe um script correspondente em /etc/hotplug/, que se encarrega de configurar corretamente o dispositivo no sistema.

Mesmo alguns dispositivos coldplug são configurados pelo sistema hotplug. Na hora do boot, o script /etc/init.d/hotplug (ou /etc/rc.d/rc.hotplug no slackware) dispara os agentes em /etc/hotplug/ para configurar aqueles dispositivos presentes antes da máquina ser ligada.

Configurando dispositivos de Rede

*"Há uma grande diferença entre saber o caminho
e percorrer o caminho"*

Filme: Matrix

Configurando dispositivos de rede

O primeiro passo para configurarmos um dispositivo de rede é identificar o tipo de barramento que ele utiliza. Se o barramento for ISA pode ser preciso alterar a configuração de jumpers retirando ou colocando pequenos terminais na placa para adequar o endereço de E/S e IRQs. Algumas dispensam o acesso físico à placa, pois são reconhecidas pela BIOS que fornecerá a configuração dos recursos.

Caso sua placa seja PCI ou CNR, o comando `lspci` pode ser útil para identificar o fabricante e o modelo do dispositivo:

```
lspci
```

Modelos que utilizam barramento PCI na maioria das vezes dispensam a configuração manual dos recursos despendidos.

Você pode consultar o endereço e a interrupção que o dispositivo está utilizando:

```
cat /proc/interrupts
```

```
cat /proc/ioports
```

Com o resultado obtido no comando acima você já está apto à configurar seu dispositivo.

O primeiro passo é carregar o módulo do seu dispositivo, exemplo:

```
modprobe 8139too
```

Este módulo dá suporte as placas Realtek deste modelo.

Algumas placas ISA necessitam que o endereço de E/S e IRQ sejam passados como argumentos para o comando `modprobe` alocar os recursos corretamente.

```
modprobe ne io=0x300 irq=10
```

O Linux suporta que os módulos sejam carregados automaticamente durante o boot do sistema. As configurações devem ser feitas no arquivo `/etc/modules.conf` ou `(/etc/modules)` no Debian.

Seguindo o exemplo eu colocaria neste arquivo:

```
Alias eth0 8139too
Alias eth1 ne
Options ne io=0x300 irq=10
```

Configurando IP Estático

Neste tipo de conexão o endereço IP, máscara de rede, Gateway default e os endereços de servidor de nomes (DNS) à serem configurados deverão ser informados pelo provedor.

Em linha gerais, utiliza-se o comando `ifconfig` para configurar o endereço IP e máscara de sub-rede da placa ethernet que está conectada ao modem.

```
ifconfig eth0 200.150.233.45 up netmask 255.255.255.0
```

Já o default gateway deverá ser configurado utilizando o comando `route` para adicionar a rota padrão:

```
route add default gw 200.150.233.1 dev eth0
```

Por fim, o endereço do servidor de nomes deverá ser configurado:

```
echo "nameserver 8.8.8.8" >> /etc/resolv.conf
```

Configurando IP Dinâmico via DHCP

Os clientes de DHCP mais comuns nas distribuições de Linux são:

`dhcpcd` e o `pump`

O cliente DHCP negocia com o provedor de serviços uma espécie de empréstimo de entrega de endereço IP que será utilizado durante aquela conexão. Enquanto a conexão é estabelecida também é configurado, máscara de sub-rede, default Gateway e servidor de nomes.

Usando o PUMP:

```
pump -i eth0 -status
```

Para devolver um endereço IP

```
pump -i eth0 -release
```

Para solicitar um endereço IP

```
pump -i eth0
```

O cliente `dhcpcd` é mais simples. Sua utilização básica resume em solicitar um endereço IP e devolver um endereço.

Para solicitar um endereço IP

```
dhcpcd eth0
```

Para devolver um endereço IP

```
dhcpcd -k eth0
```

Podemos utilizar também o dhclient (padrão no debian)

```
dhclient eth0
```

Configurando IP dinâmico usando PPPoX

A maioria das conexões de alta velocidade no Brasil utiliza modems DSL, como VIRTUA, VELOX, SPEED dentro de outros produtos.

Estes modems costumam aceitar um dos dois tipos de mecanismos para estabelecer conexão: Bridged/DHCP ou PPPoX. Ambos têm a tarefa de obter um endereço IP e outras configurações de rede para que a conexão seja feita. O importante é sabermos qual tipo de configuração o modem utiliza para podermos instalar o software cliente apropriado.

Para configurar uma conta PPPoE que necessita autenticação (Geralmente e-mail e senha do provedor) instale o pppoeconf

```
apt-get install pppoeconf
```

Caso esteja utilizando o Debian Squeeze faz-se necessário adicionar repositórios (Endereços para baixar os pacotes/aplicativos) para isto edite o arquivo /etc/apt/sources.list

Após a instalação, execute:

```
pppoeconf
```

Veja uma lista completa de repositórios para o sources.list na próxima página.

```
## Debian Squeeze sources.list
deb-src http://security.debian.org/ squeeze/updates main

## Debian security updates:
deb http://security.debian.org/ squeeze/updates main contrib non-free
deb-src http://security.debian.org/ squeeze/updates main contrib non-free

## Debian.org:
deb http://ftp.debian.org/debian/ squeeze main contrib non-free
deb-src http://ftp.debian.org/debian/ squeeze main contrib non-free

## Debian Official Repository Mirror squeeze:
deb ftp://debian.oregonstate.edu/debian/ squeeze main contrib non-free
deb-src ftp://debian.oregonstate.edu/debian/ squeeze main contrib non-free
deb ftp://debian.oregonstate.edu/debian/ squeeze-proposed-updates main
contrib non-free
deb-src ftp://debian.oregonstate.edu/debian/ squeeze-proposed-updates main
contrib non-free

## Debian NL mirror:
deb http://ftp.nl.debian.org/debian squeeze main contrib non-free
deb-src http://ftp.nl.debian.org/debian squeeze main contrib non-free

## Debian DE mirror:
deb http://ftp.de.debian.org/debian squeeze main contrib non-free
deb-src http://ftp.de.debian.org/debian squeeze main contrib non-free

## Debian US mirror:
deb ftp://ftp.us.debian.org/debian/ squeeze main contrib non-free
deb-src ftp://ftp.us.debian.org/debian/ squeeze main contrib non-free

## Tor:
deb http://deb.torproject.org/torproject.org squeeze main
#Then add the gpg key used to sign the packages by running
#gpg --keyserver keys.gnupg.net --recv 886DDD89
#gpg --export A3C4F0F979CAA22CDBA8F512EE8CBC9E886DDD89 | sudo apt-
key add -

## EFTel mirror updates
deb http://mirror.eftel.com/debian squeeze main non-free contrib
deb-src http://mirror.eftel.com/debian squeeze main non-free contrib

## Debian.hands.com squeeze:
deb http://debian.hands.com/debian/ squeeze main non-free contrib
deb-src http://debian.hands.com/debian/ squeeze main non-free contrib

## Debian Multimedia squeeze:
deb http://mirror.home-dn.net/debian-multimedia squeeze main
```

Configurar dispositivo USB

O Bus Serial Universal (Universal Serial BUS, ou USB) é um tipo de interface usada para se conectarem vários tipos de periféricos, que vão desde teclados e mouses até discos rígidos, scanners, câmeras digitais e impressoras.

O barramento USB pode operar a baixa velocidade, 1,5 megabits por segundo (Mbps) e a alta velocidade, 12 Mbps. O primeiro modelo é geralmente usado por dispositivos como mouse, teclado e Joystics. As velocidades mais altas são usadas por equipamentos como scanners e impressoras.

O primeiro passo é determinar qual o chip é utilizado em sua placa mãe:

```
lspci -vt
```

Caso sua placa-mãe possua um chipset INTEL ou VIA será necessário subir o módulo USB-uhci. Se possuir um chipset da OPTi, SIS, Ali, Compaq, Apple será necessário subir o módulo USB-ohci

```
modprobe usb-uhci
```

ou

```
modprobe usb-ohci
```

Topologia USB

Os dispositivos USB são conectados a um host em uma árvore, através de um número qualquer de dispositivos hub. O comando `lsusb` pode ser usado para ver como os dispositivos se encontram conectados fisicamente a um sistema Linux.

```
lsusb -t
```

Controlador USB

Existem três tipos de controladores host USB:

- Interface de controlador de Host Aberta (Open Host Controller interface, OHCI)
- Interface de controlador de Host Universal (Universal Host Controller Interface, UHCI)
- Interface de controlador de Host Aprimorada (Enhanced Host Controller Interface, EHCI)

Os controladores OHCI e UHCI são ambos da especificação USB 1.1, capaz de transmitir dados na velocidade máxima de 12 Mbps. Os controladores EHCI são

USB 2.0, Capaz de um máximo teórico de 480 Mbps. Para conseguir velocidade acima do USB 1.1, você precisa ter um controlador USB 2.0, bem como dispositivos, hubs e cabos USB 2.0. Um dispositivo USB 2.0 conectado a um hub USB 1.1 só conseguirá rodar com velocidades USB 1.1.

Ferramentas para configuração de dispositivo USB

Algumas distribuições possuem ferramentas para configuração automática de dispositivos USB, como o `usbmgr` e o `hotplug`.

O `usbmgr` possui o arquivo `/etc/usbmgr.conf` que contém uma vasta lista de dispositivos e fabricantes que são automaticamente reconhecidos e configurados. Você não precisa alterar este arquivo.

O `hotplug` foi introduzido a partir do Kernel 2.4 para reconhecer e configurar automaticamente dispositivos USB, PCI Cardbus (PCMCIA), IEEE 1394 (Firewire) e docking stations para laptops.

Ele é responsável por carregar os módulos apropriados, ajustar permissões e executar tarefas diversas sempre que um novo dispositivo é conectado.

O `hotplug` tornou-se componente padrão para detecção de hardware USB a partir do Kernel 2.6. Em algumas distribuições você inicia o serviço:

```
/etc/init.d/hotplug start
```

Processo de Boot

*"Eu só posso lhe mostrar a porta.
Você tem que atravessá-la"
Filme: Matrix*

Processo de Boot

Qualquer computador PC quando ligado inicia uma série de ações complexas de teste de hardware programada por uma memória especial chamada BIOS (Basic Input Output System).

Durante o boot, o BIOS realiza uma série de testes, cuja função é determinar os componentes de hardware existentes no sistema. Este teste é chamado de POST (Power-on Self Test). É através do POST que o computador busca informações dos números e dos tipos de placas, drivers de disquete, HDs, portas seriais, paralelas e USBs, monitor, mouse e teclado.

Depois do POST a BIOS procura no dispositivo de disco rígido, disquete, pendrive, ou CD/DVD-Rom um endereço especial chamado setor de boot.

A MBR

A MBR (Master Boot Record) ocupa o primeiro setor do disco (512 bytes). Este primeiro setor contém a tabela de partições e o carregador de boot. Terminado os procedimentos da BIOS, o carregador de boot é disparado, que por sua vez procura na tabela de partições uma partição ativa e carrega o primeiro setor dessa partição.

No Linux os gerenciadores mais comuns são o:

LILO e o GRUB

O LILO não é mais abordado na prova LPI, por este motivo, não vamos abordar neste livro.

Após a carga do Kernel, este inicia um processo especial chamado INIT. Este processo é o pai de todos os processos e responsável pelo restante da carga do boot do Linux.

Depois da carga do boot, o init chama outro programa especial chamado getty que é responsável pela autenticação dos usuários e pelo início do processo de Shell.

É importante que você entenda que cada programa é pelo menos um processo e que cada processo possui alguns atributos, tais como:

Process ID (PID) ou identificação do processo:

Cada processo possui um número de identificação único. O primeiro processo INIT sempre terá o PID 1 e para o restante dos processos este número é incrementado à medida que novos processos são executados.

User ID e Group ID (ID do usuário e ID do grupo)

Os processos precisam ser executados com os privilégios de uma conta de usuário e do grupo associados a eles. Isto é importante por quê assim o sistema pode determinar e gerenciar o acesso aos recursos.

Processo pai:

Todos os processos no sistema Linux, com exceção do INIT, possuem um processo pai, que é responsável pela sua execução. O atributo ParentID grava o PID do processo pai. Caso o processo pai termine sua execução antes do processo filho, o processo filho é “apadrinhado” pelo init. Ganhando o parent ID igual a 1;

Variáveis de ambiente:

Cada processo herda do processo pai algumas variáveis de ambiente que simplesmente guardam alguns valores que podem ou não ser importantes para o processo em execução. É possível que durante sua execução um processo altere, incremente ou apague uma variável de ambiente.

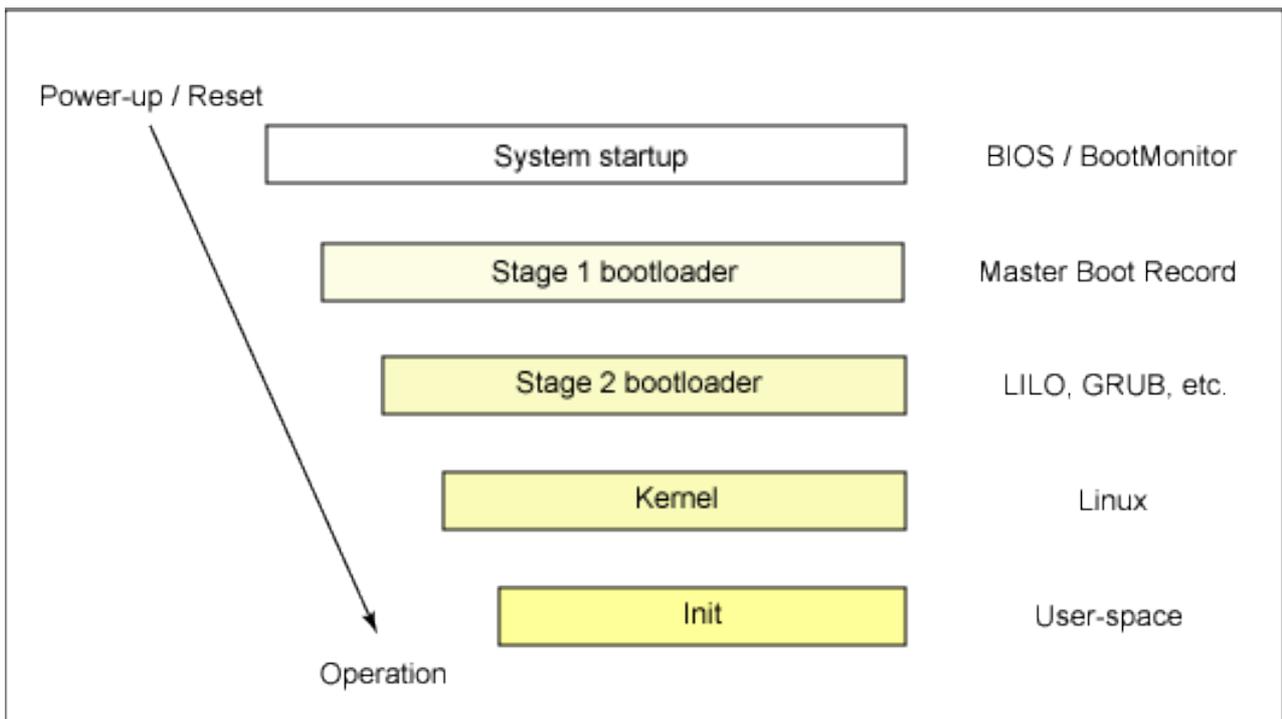
Diretório de trabalho

Os processos também são associados a um diretório de trabalho, onde podem fazer a leitura e a escrita do disco.

Temporizadores:

O kernel mantém registros da hora em que os processos são criados bem como o tempo de CPU que eles consomem durante a sua execução.

Boot do Linux:



GRUB

O grub (Grand Unified Bootloader) é uma alternativa ao lilo. Também é instalado na MBR; pelo comando `/sbin/grub` ou pelo `/sbin/grub-install`, que obtém as instruções de `/boot/grub/grub.conf`.

Principais seções de `/boot/grub/grub.conf`

No debian foi criado arquivo `/etc/default/grub` (para fazer alterações no `grub.conf`)

Após realizar as alterações utilize o comando:

```
update-grub
```

Principais informações do arquivo de configuração:

global

- Default - imagem de boot padrão (começar por 0)
- Timeout - tempo de espera para iniciar o boot, em segundos

imagem

- title - nome para a imagem
- root - localização do carregador de segundo estágio e do kernel (`hd0,0` = `/dev/hda`)
- kernel- é o caminho para o kernel à partir de root
- ro - Read-only
- initrd - caminho para a imagem `initrd`

Diferente do lilo, o comando `grub-install` não precisa ser executado toda vez que forem feitas alterações ao arquivo `/boot/grub/grub.conf`. Uma vez instalado, o carregador de boot do grub lê o arquivo `/boot/grub/grub.conf` diretamente ao reiniciar o sistema.

Para que estas alterações sejam feitas sem reiniciar execute:

```
update-grub
```

Configurando o GRUB

O arquivo no qual ficam armazenadas as entradas de todos sistemas operacionais instalados é o `grub.cfg`, que fica em `/boot/grub/`. Para visualizá-lo você precisa rodar o comando:

```
sudo gedit /boot/grub/grub.cfg
```

Logo será aberto um editor de texto com o arquivo onde na primeira linha estará escrito:

DO NOT EDIT THIS FILE

Ou seja, esse arquivo "teoricamente" não pode ser editado, porém se você quiser editar a ordem das entradas e suas escritas nada o impede.

O arquivo o qual você personalizará suas entradas no grub2 é o `40_custom`, que fica em `/etc/grub.d/`.

Primeiro atualize a lista de seus sistemas instalados com o comando:

```
sudo update-grub
```

Agora abra o arquivo `grub.cfg`:

```
sudo gedit /boot/grub/grub.cfg
```

Nele você terá todas entradas de seus sistemas instalados, onde abaixo segue um exemplo de entradas de Linux e Windows:

```
menuentry 'Ubuntu, with Linux 2.6.35-23-generic' --class ubuntu --class gnu-
linux --class gnu --class
os {
    recordfail
    insmod part_msdos
    insmod ext2
    set root='(hd0,msdos5)'
    search --no-floppy --fs-uuid --set f8f54563-c324-41a5-8412-275cb97418e2
    linux /boot/vmlinuz-2.6.35-23-generic root=UUID=f8f54563-c324-41a5-8412-
275cb97418e2 ro
    quiet splash
    initrd /boot/initrd.img-2.6.35-23-generic
}
menuentry "Windows 7 (loader) (on /dev/sda1)" {
    insmod part_msdos
    insmod ntfs
    set root='(hd0,msdos1)'
```

```
search --no-floppy --fs-uuid --set 6020fefa20fed64e
chainloader +1
}
```

Agora para facilitar a edição abra em outro terminal o arquivo 40_custom:

```
sudo gedit /etc/grub.d/40_custom
```

Nele você terá um arquivo assim:

```
#!/bin/sh
exec tail -n +3 $0
```

This file provides an easy way to add custom menu entries. Simply type the # menu entries you want to add after this comment. Be careful not to change # the 'exec tail' line above.

Copie as entradas encontradas no arquivo grub.cfg para o 40_custom, onde as entradas começam em "menuentry {" e terminam em "}".

Após copiar suas entradas do grub.cfg para o arquivo 40_custom, ele deve ficar mais ou menos assim:

```
#!/bin/sh
echo "Adding 40_custom menu entries." >&2
exec tail -n +3 $0
menuentry 'Ubuntu, with Linux 2.6.35-23-generic' --class ubuntu --class gnu-
linux --class gnu --class
os {

    recordfail
    insmod part_msdos
    insmod ext2
    set root='(hd0,msdos5)'
    search --no-floppy --fs-uuid --set f8f54563-c324-41a5-8412-275cb97418e2
    linux /boot/vmlinuz-2.6.35-23-generic root=UUID=f8f54563-c324-41a5-8412-
275cb97418e2 ro
    quiet splash
    initrd /boot/initrd.img-2.6.35-23-generic
}
menuentry "Windows 7 (loader) (on /dev/sda1)" {
    insmod part_msdos
    insmod ntfs
    set root='(hd0,msdos1)'
    search --no-floppy --fs-uuid --set 6020fefa20fed64e
    chainloader +1
}
```

Agora você poderá editar o texto que será exibido no boot, como por exemplo trocando de:

```
menuentry 'Ubuntu, with Linux 2.6.35-23-generic' --class ubuntu --class gnu-  
linux --class gnu --class os
```

Para:

```
menuentry " Ubuntu Maverick 10.10 Kernel 2.6.32 "
```

Isso é só um exemplo, ficando a seu critério o conteúdo escrito.

Após editar o arquivo 40_custom basta salvar e fechar ele.

Para definir a ordem de boot, ou seja, para definir qual sistemas vai iniciar após a contagem no boot esgotar, é preciso editar o arquivo grub que fica em /etc/default

```
sudo gedit /etc/default/grub
```

GRUB_DEFAULT=0 <<< configura a entrada padrão sendo "0" para 1º entrada, "1" para 2º entrada e assim por diante.

GRUB_TIMEOUT=3 <<< aqui você irá escolher o tempo em segundos para contagem no boot

Depois de editar esse arquivo basta salvar e fechar.

Agora rode esse comando:

```
sudo chmod -x /etc/grub.d/10_linux /etc/grub.d/30_os-prober
```

Esses dois arquivos por padrão vem em modo executáveis, ou seja, a cada update-grub que você der ele irá inserir as entradas no seu grub mas como você já editou elas no arquivo 40_custom elas ficariam duplicada e rodando esse comando você retira essas duplicações.

Obs.: A cada nova instalação de um Linux ou Windows ou a cada atualização ou instalação de um novo kernel você precisará voltar esses arquivos "10_Linux e 30_os-prober" para o estado de executáveis para que eles possam inserir as novas entradas no grub.cfg e assim você as copia para o arquivo 40_custom. Para retornar o modo executável desses arquivos faça:

```
$ sudo chmod +x /etc/grub.d/10_linux /etc/grub.d/30_os-prober << trocando -x  
por +x
```

Feito isso rode um:

```
sudo update-grub
```

Vá até o arquivo grub.cfg e copie as novas entradas para o 40_custom e depois retire o modo executável dos arquivos:

```
sudo chmod -x /etc/grub.d/10_linux /etc/grub.d/30_os-prober
```

E novamente rode:

```
$ sudo update-grub
```

Caso queira retirar o memtest também rode esse comando:

```
sudo chmod -x /etc/grub.d/20_memtest86+
```

E rode um:

```
$ sudo update-grub
```

E pronto! Pode reiniciar seu PC para ver o resultado.

Notação do Linux	Notação do grub
/dev/hda	(hd0)
/dev/hda1	(hd0,0)
/dev/hda2	(hd0,1)
/dev/hdb	(hd1)
/dev/hdb1	(hd1,0)
/dev/hdb2	(hd1,1)
/dev/sda	(hd0)
/de/sda1	(hd0,0)
/dev/sda2	(hd0,1)
/dev/sdb	(hd1)
/dev/sdb1	(hd1,0)
/dev/sdb2	(hd1,1)

Alterando Runlevels, shutdown e reboot

O Linux possui três grandes sistemas especiais de boot de serviços responsáveis pela iniciação dos processos:

- System V Init Daemon
- Systemd
- Upstream

O system V foi herdado do Unix, que recebeu o nome de System V init Daemon ou simplesmente de init.

Durante o processo de boot (GRUB) carrega a imagem do Kernel para a memória. Assim que o Kernel assume o controle da máquina, ele carrega um programa chamado init e que sempre tem o PID 1.

Você pode ver essa relação com o comando pstree:

```
pstree
```

Runlevel

No Linux existe sete níveis de execução, definidos de 0 à 6. Os servidores que cada runlevel pode executar vão depender da distribuição de Linux e da configuração que o administrador do sistema efetuou.

Runlevel	Descrição
0	O nível zero define um desligamento rápido do sistema
1	O nível um é utilizado para manutenção do sistema. Também chamado monousuário (single user) somente o essencial é executado
2	Modo multiusuário com compartilhamento de arquivos NFS desabilitado
3	Modo multiusuário com todos os serviços habilitados mas sem interface gráfica para login no sistema.
4	Não utilizado
5	Modo multiusuário com todos os serviços habilitados, interface gráfica x11 e login em modo gráfico
6	O nível 6 é parecido com o nível zero, mas executa o reboot da máquina.

O administrador do sistema pode a qualquer momento alterar o nível de execução através do comando init seguido do número do runlevel desejado.

```
init 5
```

Ou

```
telinit 6
```

telinit é um link para o init.

Quando o Linux inicia, uma série de scripts no diretório `/etc/rc.d` é executado para dar o boot do sistema. O arquivo `rc.sysinit` é executado pelo processo `init` durante o boot do sistema e habilita as funções essenciais. Por exemplo, montar partições em disco. O arquivo `rc.local` é executado pelo script `rc.sysinit`. Ele é utilizado pelos administradores para eles modificarem os serviços na carga do sistema. O arquivo `rc` é utilizado para transições entre os runlevels. Os níveis de serviço são definidos através dos diretórios `/etc/rc.<n>` onde `<n>` pode variar de 0 a 6. Correspondendo ao runlevel. Dentro de cada diretório existem links simbólicos para os scripts de serviços.

- A Letra K, de Kill, indica que o script é para terminar um determinado serviço
- A letra S, de Start, indica que o script é para iniciar um determinado serviço

Definindo Runlevel Padrão

O arquivo que define qual o runlevel que o sistema irá iniciar é:

```
/etc/inittab
```

Neste arquivo basta editar a linha:

```
id:n:initdefault
```

O “n” definirá o runlevel pelo seu número de 0 à 6.

O comando `runlevel` somente informa o nível de execução, para alterar é necessário o comando `init`.

Exemplo:

```
runlevel
```

shutdown

O comando `shutdown` é utilizado para desligar ou reiniciar o sistema com horários determinados, exemplo:

```
shutdown {opções} horário {mensagem}
```

Para reiniciar o sistema imediatamente:

```
shutdown -r now
```

Para reiniciar o sistema as 06 da manhã e enviar mensagem no terminal do usuário:

```
shutdown -r 06:00 "Haverá troca de hardware"
```

Para desligar o sistema em 10 minutos:

```
shutdown -h 10
```

Instalação e gerenciamento de pacotes

*"Não acredito em destino. Por que não gosto da ideia de não
poder controlar minha vida"*

Filme: Matrix

Instalação e Gerenciamento de Pacotes do Linux - Peso 11

Para acessar recursos em disco o sistema utiliza um mecanismo chamado montagem. Em sistemas UNIX, significa anexar um disco a um diretório, chamado ponto de montagem. Para o usuário, os recursos aparecem como uma árvore de diretórios e subdiretórios.

A raiz da árvore de diretórios é representada por uma barra “ / “. É necessariamente o primeiro diretório a ter um dispositivo anexado. Depois de montada a raiz, os diretórios contidos neste dispositivo poderão ser pontos de montagem para outros dispositivos.

Processo de montagem:

- O carregador de boot carrega o kernel e transmite as informações sobre a localização do dispositivo raiz.
- Os demais dispositivos são montados conforme as instruções encontradas em `/etc/fstab`.

Etapas da criação do layout de disco:

- Criar partições de tamanho específico
- Escolher o sistema de arquivos
- Determinar um ponto de montagem para cada partição

Duas partições é o mínimo exigido em sistemas GNU/Linux, uma que será a raiz “ / “ e outra que será a partição de troca swap. Muitas vezes, pode haver uma terceira partição pequena, no início do disco, apenas para armazenar o kernel e o carregador de boot secundário. Fora essas, não há regras inflexíveis quanto à criação de partições, devendo ser avaliado o melhor esquema para a função que o sistema desempenhará.

A partição raiz deve ser do tipo Linux Native, cujo código é 83 (0x83). Tudo no sistema poderá existir diretamente no dispositivo raiz. No entanto, certos tipos de arquivos são processados de formas bem distintas de outros arquivos. Em certos casos, é interessante criar uma partição distinta para certos diretórios, principalmente em servidores que são muito exigidos. Essa estratégia também impede que os dados no disco se fragmentem muito.

Outra questão é o backup. Enquanto certos arquivos não necessitam de backup, outros exigem-no. Mesmo dentre os quais é realizado o backup, alguns são pouco alterados durante um determinado período enquanto que outros são constantemente alterados ou criados. A criação de partições diferentes para cada diretório que comporte um tipo distinto de arquivos facilita as operações de criação e recuperação de backup.

Sugestões de diretórios que podem estar em outros dispositivos/partições:

/var

Este diretório contém os arquivos de email e impressão, que são muito manipuladas. Há também os arquivos de log, cujo conteúdo estão em constante alteração e crescimento.

/usr

Programas, códigos fonte e documentação. O ciclo de alterações desses arquivos é longo.

/tmp

Espaço temporário utilizado por programas. Uma partição distinta para /tmp impedirá que dados temporários ocupem todo o espaço no diretório raiz, causando travamento do sistema. Não necessita de backup.

/home

Diretórios pessoais do usuário. Uma partição distinta ajuda a limitar o espaço disponível para usuários comuns.

/boot

Ponto de montagem para as partições do kernel e do carregador de boot. Necessário apenas caso o sistema exija que o kernel esteja antes do cilindro 1024 do disco.

A Partição Swap

Uma partição swap é identificada pelo código 82 (0x82), atribuído quando da sua criação. Geralmente, a partição swap é do mesmo tamanho que o montante de memória RAM no sistema. Para servidores, esse número pode ser maior. Uma partição swap muito maior que isso provavelmente não terá o espaço excedente utilizado. É possível haver mais de uma partição de swap.

É preferível criar partições de swap nos dispositivos mais rápidos, se possível em dispositivos distintos daqueles que tem seus dados muito acessados pelo sistema.

Também é possível criar grandes arquivos como área de swap, o que é geralmente feito em situações emergenciais, quando o sistema ameaça ficar sem memória disponível:

Criar o espaço:

```
dd if=/dev/zero of=emerg.swp bs=1024k count=32
```

Definir como swap:

```
mkswap emerg.swp
```

Ativar a swap:

```
swapon emerg.swp
```

Verificar:

```
cat /proc/swaps
```

Compressão e Descompressão de Arquivos

A maioria dos programas distribuídos em código fonte apresenta-se na forma de arquivos tar (tape archiver) comprimidos. Arquivos tar são vários arquivos aglutinados em um só, o que facilita a sua distribuição.

Tipos de compactação

Compressão	Descompressão	Descompressão "cat"	Extensão
compress	uncompress	zcat	.z
gzip	gunzip	zcat	.gz
gzip2	bunzip2	bzcat	.bz2

Os comandos zcat e bzcat descomprimem para a saída padrão, ou seja, jogam o conteúdo descomprimido na tela do terminal.

Essas ferramentas de compressão não concatenam arquivos, por isso são utilizadas junto com o comando tar. Para facilitar esse procedimento, a compressão e descompressão podem ser feitas diretamente pelo comando tar, através dos argumentos:

- Z - compress
- Z - gzip
- j - bzip2

Criar um arquivo tar "programa.tar.bz2" contendo o diretório ./programa/ e seu conteúdo, compactando com bzip2:

```
tar -cjf programa.tar.bz2 ./programa/
```

Extrair esse arquivo:

```
tar -xjf programa.tar.bz2
```

Como mostrado no exemplo, o argumento `c` indica criação de arquivo, `x` extração de arquivo e `f` o nome do arquivo.

Compilar e Instalar Programas

O primeiro passo para instalar um programa em código fonte distribuído no formato tar é extraí-lo:

```
tar -xjvf sylpheed-2.0.4.tar.bz2
```

Este comando criará o diretório `./sylpheed-2.0.4/`, contendo o código fonte do programa e as ferramentas de configuração.

A configuração pré-compilação é feita dentro desse diretório, por um script chamado `configure`. Este script coleta informações sobre a arquitetura do sistema, caminhos de comandos, bibliotecas compartilhadas, características de funcionamento do programa, etc.

Em geral, o `configure` pode ser invocado sem argumentos na forma:

```
./configure
```

Uma opção bastante utilizada é `--prefix`, que informa em qual diretório base o programa deverá ser instalado. Por exemplo:

```
./configure --prefix=/opt
```

Indica que o diretório base da instalação será `/opt`. O script `configure` possui muitas opções de personalização da instalação. A lista descritiva completa das opções pode ser vista usando

```
./configure --help
```

As informações coletadas pelo script `configure` são armazenadas em um arquivo no mesmo diretório chamado `makefile`. Este arquivo pode ser editado para alterar as opções de instalação, como a variável `prefix`, que desempenha a mesma função do argumento `--prefix` do script `configure`, e outras variáveis que indicam a localização de bibliotecas, comandos, características do programa, etc.

Terminada a configuração, o programa pode ser compilado através do comando `make`. `Make` criará as bibliotecas e arquivos executáveis conforme as opções existentes no `makefile`. Após o término da compilação, que pode levar algum tempo dependendo do tamanho e tipo do programa, o programa estão pronto para ser instalado

```
make install
```

Se o diretório base de instalação não estiver no diretório pessoal do usuário, esse comando deverá ser executado com permissões de super usuário (root). Mesmo estando o diretório base de instalação fora do diretório pessoal do usuário, É recomendado executar ./configure e make como usuário comum.

As bibliotecas Compartilhadas

Quando um programa é compilado no Linux, muitas das funções requeridas pelo programa são vinculadas a partir de bibliotecas de sistemas que lidam com os discos, memórias e outras funções.

Por exemplo, quando printf() é usado em um programa, o programador não fornece o código-fonte de printf() mas, em vez disso, espera que o sistema já possua uma biblioteca contendo esse tipo de função. Quando o compilador precisa vincular o código para printf(), esse código poderá ser encontrado em uma biblioteca do sistema e compilado para dentro do executável. Chamamos de código “estaticamente vinculado” porque se trata de um programa independente, que não precisa de nenhum outro código no momento da execução.

1. Programas estaticamente vinculados são grandes
2. Desperdiçam memória quando muitos programas diferentes executam as mesmas funções de bibliotecas

Dependências de Bibliotecas Compartilhadas

Qualquer programa que esteja dinamicamente vinculado, precisará de pelo menos algumas, poucas bibliotecas compartilhadas. Se as bibliotecas não existirem ou não puderem ser encontradas, o programa não poderá rodar. Instalaras bibliotecas corretas dever ser o suficiente para eliminar tais problemas. O utilitário ldd pode ser usado para saber quais bibliotecas são necessárias por um executável específico.

ldd programa

Exemplo:

ldd /bin/bash

```
linux-vdso.so.1 (0x00007ffffaf194000)
libreadline.so.6 => /lib64/libreadline.so.6 (0x00007fa41b110000)
libtinfo.so.5 => /lib64/libtinfo.so.5 (0x00007fa41aedc000)
libdl.so.2 => /lib64/libdl.so.2 (0x00007fa41acd8000)
libc.so.6 => /lib64/libc.so.6 (0x00007fa41a929000)
/lib64/ld-linux-x86-64.so.2 (0x00007fa41b358000)
```

O programa encarregado de carregar a biblioteca e ligar ao programa que dela depende é o **ld.so**. Para que o **ld.so** possa localizar a biblioteca da qual um programa depende, esta deverá estar mapeada em **/etc/ld.so.cache**. As

localidades comuns de bibliotecas de sistema são **/lib** e **/usr/lib**. Para acrescentar um diretório ao `ld.so.cache`, o arquivo **/etc/ld.so.conf** é usado:

Exemplo de `/etc/ld.so.conf`

```
/usr/local/lib
/usr/X11R6/lib
/usr/i486-slackware-linux/lib
/usr/lib/qt/lib
```

Para atualizar o `/etc/ld.so.cache` após as alterações em `/etc/ld.so.conf`, é utilizado o comando **ldconfig**. A execução do `ldconfig` é fundamental para que as alterações em `/etc/ld.so.conf` repercutam no funcionamento do `ld.so`.

Outra maneira de deixar uma localidade de biblioteca ao alcance do `ld.so` é adicionar seu o respectivo caminho à variável de ambiente **LD_LIBRARY_PATH**

```
export LD_LIBRARY_PATH=/usr/local/lib
```

Esse método, porém, garante apenas o acesso temporário do `ld.so` ao diretório em questão. Não funcionará fora do escopo da variável de ambiente e quando a variável deixar de existir, mas é um método útil para usuários sem permissão para atualizar o `/etc/ld.so.cache` ou para execução pontual de programas.

ldconfig

Uso:

```
ldconfig (opções) diretório de bibliotecas
```

O utilitário `ldconfig` cria os links e refaz o índice das bibliotecas dinâmicas do arquivo `/etc/ld.so.cache`. Ele procura por bibliotecas nos diretórios `/usr/lib` e `/lib`, assim como nos diretórios listados em `/etc/ld.so.conf`. Bem como o diretório informado na linha de comando.

As opções mais comuns são:

-p : lista o conteúdo do cache `/etc/ld.so.cache`

-v: Mostra o progresso da atualização do cache

-f: arquivo informa um outro arquivo de configuração diferente do padrão `/etc/ld.so.conf`

Procure por uma entrada de uma biblioteca específica no cache:

```
ldconfig -p | grep ncurses
```

Neste caso à referência “ncurses”.

Gerenciamento de pacotes do Debian (DEB)

O sistema de gerenciamento de pacotes do Debian é um dos sistemas de instalação de pacotes mais versátil e automatizado entre os sistemas linux.

Cada pacote do Debian contém arquivos de programas e de configuração, documentação e indicações de dependências em relação a outros pacotes. Os nomes dos pacotes do Debian possuem três elementos em comum, incluindo:

Nome do Pacote

Geralmente curto e descritivo: kernel-sources, telnet

Número da versão:

Cada pacote tem uma versão. A maioria das versões dos pacotes tem o mesmo número que o software que ela contém. Por padrão todos os pacotes do debian terminam com a extensão .deb

pacote	-	1.0	-	2	-	deb
Nome do pacote		versão do software		versão do pacote		extensão

dpkg

Sintaxe:

dpkg (opções) ação

O comando dpkg mantém as informações dos pacotes instalados basicamente nos arquivos /var/lib/dpkg/available e /var/lib/dpkg/status, respectivamente.

Available - É a lista de todos os pacotes disponíveis

Status - Contém atributos dos pacotes, tais como se um determinado pacote está instalado ou se está marcado para remoção.

Opções frequentemente utilizadas

-E

Instrui o comando a não sobrescrever um pacote, previamente instalado, da mesma versão.

-G

Instrui o comando a não sobrescrever um pacote, previamente instalado, com uma versão mais antiga do que a do pacote instalado.

-R (Também **-recursive**)

Processa recursivamente arquivos de pacotes em subdiretórios especificados, tabalha com **-i**, **--install**, **--unpack**, e assim por diante.

--configure nome_do_pacote

Configura um pacote desempacotado. Requer o setup de arquivos de configuração.

-i nome_do_pacote (também **-install nome_do_pacote**)

Instala o pacote.

--purge nome_do_pacote

Remove tudo à respeito do pacote

-r nome_do_pacote

Remove tudo, exceto os arquivos de configuração.

-S nome_do_pacote (Também **-status nome_do_pacote**)

Relata o status do pacote

Exemplo: Instalando um pacote com DPKG

```
dpkg -i hdparm 3 3 3.deb
```

Lembre-se que o comando de instalação de pacotes só funciona como super usuário (root).

Exemplo: Removendo um pacote com DPKG

```
dpkg -purge hdparm
```

dpkg-reconfigure

```
dpkg-reconfigure nome_do_pacote
```

O comando `dpkg-reconfigure` reconfigura um pacote já instalado no sistema. Ele tem o mesmo efeito da re-instalação do pacote.

apt-get

O `apt-get` busca em um servidor específico (local ou internet) programas debian e realiza sua instalação de modo automatizado. O primeiro passo é verificar o endereço dos servidores de repositórios de programas que fica no endereço:

```
/etc/apt/sources.list
```

Após adicionar ou remover endereço de repositórios no endereço acima é necessário executar o comando de atualização do `apt-get`:

```
apt-get update
```

Comandos frequentemente utilizados:

-d

Faz o download do arquivo mas não instala.

-s

Simula os passos em uma modificação de pacotes, mas não modifica realmente o sistema.

-y

Responda “yes” (“Sim”) automaticamente a todos os prompts, em vez de esperar que você responda, durante a instalação/remoção de pacotes.

dist-upgrade

Faz upgrade automaticamente para novas versões do Debian Linux

install

Instala ou faz upgrade de um ou mais pacotes, por nome

remove

Remove os pacotes especificados

update

Obtém uma lista dos pacotes atualmente disponíveis. Isso normalmente é feito antes que quaisquer modificações sejam feitas aos pacotes existentes.

upgrade

Faz upgrade do conjunto completo de pacotes de um sistema para as versões mais atualizadas de forma segura.

aptitude

O utilitário aptitude é uma interface para facilitar o uso do APT. Ele mostra uma lista de pacotes e permite que o usuário selecione os pacotes para instalação ou desinstalação de forma interativa.

dselect

O comando dselect é um utilitário para instalar e desinstalar pacotes através de menus interativos. Ele funciona como uma interface amigável para o dpkg.

alien

Converte ou instala um pacote não-debian. Os tipos de pacotes suportados incluem o .rpm do Red Hat, o .slp do stampede, o .tgz do Slackware, além de arquivos .tar.gz genéricos. RPM precisa estar também instalado no seu sistema para converter os pacotes RPM em pacotes .deb. O comando alien produz um pacote de saída no formato do Debian, por padrão, após a conversão.

Sintaxe:

-i

Instala automaticamente o pacote de saída e remove o arquivo de pacote convertido.

-r

Converte o pacote para o formato RPM

-t

Converte o pacote em um arquivo gzip tar.

unpack

```
unpack apache.tar.gz
```

o utilitário unpack é um script perl utilizado para extrair pacotes tarball em um determinado diretório. Geralmente ele utiliza o diretório HOME do usuário ou o diretório temporário /tmp.

Apt-cache

O utilitário apt-cache é para manipular e obter informações sobre os pacotes no cache do apt.

Ele deve ser utilizado em conjunto com alguns comandos, a seguir:

```
apt-cache add
```

Adiciona um pacote ao cache do apt

```
apt-cache gencaches
```

Gera o cache do apt

```
apt-cache showpkg
```

Mostra algumas informações sobre um determinado pacote

```
apt-cache stats
```

Mostra alguma estatísticas

```
apt-cache check
```

Verifica a sanidade do cache

```
apt-cache search
```

Procura na lista de pacotes por uma determinada ocorrência

```
apt-cache depends
```

Mostra as dependências de software de um determinado pacote

```
apt-cache pkgnames
```

Lista o nome de todos os pacotes

Gerenciamento de pacotes RPM (Red Hat)

O gerenciador de pacotes do Red Hat está entre os métodos mais populares de distribuição de software para o Linux e é instalado pôr padrão na memória das distribuições.

O Red Hat Package Manager (RPM) também possibilita construir pacotes a partir de arquivos fonte, verificar a assinatura digital e simular instalação. Cada pacote RPM contém programas binários, arquivos de configuração, documentação e informações de como eles devem ser instalados e suas dependências.

Os pacotes RPM acompanham a seguinte nomenclatura:

pacote - 1.0 - 2 - i386 - RPM
Nome - versão - Distribuição - Arquitetura - Extensão

Obs: Este padrão pode mudar dependendo do desenvolvedor do pacote.

Opções mais utilizadas do comando rpm:

- i** → Ou **--install**. Instala o pacote
- U** → Ou **--update**. Atualiza ou instala o pacote
- F** → Ou **--freshen**. Atualiza o pacote apenas se o mesmo estiver instalado
- V** → Ou **--verify**. Verifica o tamanho, MDB, permissões, tipo, integridade, etc.
- q** → Ou **--query**. Investiga pacotes e arquivos
- e** → Ou **--erase**. Desinstala o pacote

Exemplo de instalação de um pacote:

```
rpm -i gcc-2.96-i386.rpm
```

Excluir um pacote:

```
rpm -e glibc-devel
```

Por padrão o RPM só desinstala um pacote se nenhum outro programa for dependente dele.

--nodeps

RPM pula a verificação de dependências com esta opção habilitada. Este comando deve ser evitado, uma vez que ele torna inconsistente o banco de dados de dependências.

--test

Esta opção passa por todas as etapas, mas sem realmente desinstalar nada, é útil para verificar se um pacote pode ser desinstalado corretamente, sem quebrar outras dependências, antes de se fazer a desinstalação real. Repare que as opções verbose e hash não podem ser usadas com -test, mas -vv pode.

Determinar a versão do software

```
rpm -qpi openssh-3.4.i386.rpm | grep Version
```

Listar os arquivos do pacote:

```
rpm -qlp openssh-3.4.i386.rpm
```

Listar os arquivos de documentação do pacote

```
rpm -qd openssh-3.4.i386.rpm
```

Listar os scripts de configurações de um pacote:

```
rpm -qc openssh-3.4.i386.rpm
```

Listar pacotes instalados no sistemas

```
rpm -qa
```

[São listados centenas de pacotes]

Arquivos e Diretórios Pertinentes ao rpm

/etc/rpmrc

É o arquivo de configuração do rpm e do rpmbuild. Contém informações sobre a arquitetura do sistema e os caminhos para macros e diretórios utilizados no manejo de pacotes. Este arquivo é também encontrado em `/usr/lib/rpm/`

/usr/lib/rpm/*

Diretório contendo as macros necessárias para o manejo de pacotes.

/var/lib/rpm/

Diretório onde se situam as bases de dados do rpm

Comandos RPMS que geralmente aparecem no exame 101:

Instalar um pacote:

```
rpm -ivh xyz.rpm
```

Instalar todos pacotes iniciados por xyz

```
rpm -ivh xyz*
```

Atualizar um pacote:

```
rpm -Uvh xyz.rpm
```

Desinstalar pacote chamado xyz

```
rpm -e xyz
```

Mostra os arquivos contidos no pacote

```
rpm -qlp xyz.rpm
```

Mostra que pacote instalou o arquivo abc.efg

```
rpm -qf abc.efg
```

Gerenciador de pacotes YUM

O YUM (Yellow Dog Updater) é uma ferramenta utilizada para gerenciar a instalação e remoção de pacotes que utilizam o sistema RPM. O YUM assemelha-se ao APT, ele possui um arquivo de configuração que especifica os repositórios que o YUM deve utilizar e baixar os pacotes RPM.

Configurando repositórios

O arquivo de configuração está em:

```
/etc/yum.conf
```

Exemplo do arquivo:

```
[ol6_u2_base]
```

```
name=Oracle Linux 6 U2 - $basearch - base
baseurl=http://public-yum.oracle.com/repo/OracleLinux/OL6/2/base/$basearch
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY
gpgcheck=1
enabled=1
```

Alguns comandos do YUM:

yum list

Lista todos os pacotes disponíveis

yum check-update ou yum list updates

Verifica se há pacotes disponíveis para um update

yum update

Faz o update do seu sistemas

yum install <package>

Instala um pacote específico e suas dependências

yum info <package>

Apresenta informações básicas de um determinado pacote.

Comandos GNU e UNIX

"Tu te tornas eternamente responsável por aquilo que cativas"
Antoine de Saint-Exupéry

Comandos GNU e UNIX - Peso 26

Embora haja utilitários gráficos disponíveis para gerenciar quase tudo em uma distribuição de Gnu/Linux, você precisa ter um conhecimento sólido sobre os utilitários de linha de comando para se preparar melhor para trabalhar com qualquer distribuição compatível com LSB.

Sobre o Shell

O shell é um interpretador de comandos que analisa o texto digitado na linha de comandos e os executa produzindo algum resultado. Além de interpretador de comandos o shell também é um poderoso ambiente de programação capaz de automatizar praticamente tudo em um sistema linux.

Uma vez iniciado o processo do shell, ele irá preparar o prompt de comandos, indicando que está pronto para receber instruções. No linux utilizamos normalmente o BASH (ou Borne Again Shell) que é uma derivação do CSH. A prova LPI usa o BASH como padrão.

Variáveis do shell

Durante a execução, bash mantém um conjunto de variáveis de shell, as quais contém informações importantes para a execução do bash. A maioria destas variáveis é definida na inicialização do bash.

Variável PS1

A variável PS1 que significa: Prompt String, armazena o conteúdo do prompt de comando que é exibido quando bash está pronto para aceitar comandos.

Variável PS2

A variável PS2 é utilizada quando o bash precisa de entrada com múltiplas linhas para completar um comando.

Você pode exibir facilmente o conteúdo de PS1 ou de qualquer outra variável do shell, usando o comando "echo" com o nome da variável precedido pelo \$:

```
echo $PS1
```

Geralmente o Bash vem no formato: \u@\h:\w\ \$

Sendo que cada um destes caracteres é interpretado de forma especial. O \u é utilizado para representar o nome do usuário, o \h é utilizado para representar o nome do sistema (hostname) e o \w é o diretório corrente. Um exemplo deste prompt é:

```
grubelilo@enterthematrix:home$
```

Variável PATH

A variável PATH guarda uma lista de diretórios que contém os programas que você poderá utilizar sem passar para a linha de comando o endereço completo

```
echo $PATH  
  
/  
home/juliano/bin:/usr/local/bin:/usr/bin:/bin:/usr/bin/X11:/usr/X11R6/bin:/usr/g  
ames:/usr/lib/mit/bin
```

Quando executei o comando `echo $PATH` o sistema retornou os endereços de onde estão meus executáveis. Observe que ele apontou para diversos diretórios. Eu possuo o navegador firefox instalado no diretório `/usr/local/bin` então para rodar o aplicativo não preciso digitar:

```
/usr/local/bin/firefox
```

Mas somente:

```
firefox
```

Já que a variável PATH já está configurada.

Você pode verificar uma lista completa de variáveis do shell utilizando o comando:

```
set
```

Você também pode alterar ou criar uma variável do shell com os comandos:

```
MATRIX="Siga o coelho branco"  
export MATRIX
```

Para visualizar o seu conteúdo:

```
echo $MATRIX
```

Quando uma variável é exportada para o ambiente comando (`export`), ela ficará disponível para todos os programas executados pelo seu shell.

Os comandos emitidos no shell de um sistema Linux geralmente consistem de quatro componentes:

- 1 - Um comando válido (interno do shell, um programa, ou um script encontrado em um dos diretórios listados PATH, ou então um programa explicitamente definido).
- 2 - Opções do comando, em geral precedidas por um hífen
- 3 - Argumentos
- 4 - Aceitação da linha (ou seja, o pressionamento da tecla Enter), o que estará sempre presumido nos exemplos.

Cada comando tem a sua sintaxe própria, embora a maioria siga uma forma relativamente padronizada. O mínimo necessário é o comando:

ls

Esse comando simples lista os arquivos do diretório atual. Ele requer nem opções, nem argumentos. Em geral, as opções são letras ou palavras precedidas por um **hífen** simples ou duplo e são adicionadas após o comando, sendo separadas dele por um espaço:

ls -l

A opção **-l** modifica o comportamento de **ls** listando os arquivos em um formato mais longo e mais detalhado. Na maioria dos casos, as opções com hífen simples podem ser ou combinadas, ou especificadas separadamente. Para ilustrar isso, considere estes dois comandos equivalentes:

ls -l -a

ls -la

Adicionando a opção **-a**, la exibe os arquivos que começam com um ponto (os quais são escondidos por padrão). Adicionar essa opção especificando-se **-la** gera o mesmo resultado. Alguns comandos oferecem formas alternativas para a mesma opção. No exemplo anterior, a opção **-a** pode ser substituída por **--all**.

ls -a --all

Essas opções com hífen duplo e uma palavra completa são frequentemente encontradas em programas do projeto GNU. Elas não podem ser combinadas da forma como fizemos para as opções de hífen simples. Ambos os tipos de opções podem ser misturados livremente. Embora as opções estilo GNU sejam mais longas e requeiram mais digitação, elas são mais fáceis de lembrar e também mais fáceis de ler em scripts do que as opções que usam apenas uma letra.

Adicionar um argumento refina ainda mais o comportamento do comando:

ls -l *.c

Agora o comando fornecerá uma listagem detalhada apenas dos arquivos com código-fontes de programas **.C**. Ou seja, apenas extensões em **.C** se houver algum no diretório em questão. Usar o asterisco ***.c** faz com que qualquer arquivo que termine com a extensão **.c** seja incluído.

Em alguns casos, as opções e os argumentos podem ser misturados em qualquer ordem :

ls -all *.c -l

Neste caso, **ls** foi capaz de determinar que **-l** é uma opção e não outro descritor de arquivos.

Alguns comandos como **tar** e **ps**, não requerem o hífen que precede as opções porque, para esses comandos, espera-se ou requer-se pelo menos uma opção. Para ser específico, **ps** não requer um hífen quando está funcionando como o **ps** do **BSD**. Uma vez que a versão Linux de **ps** foi elaborada para ser o mais compatível possível com as diversas outras versões de **ps**, ela às vezes precisa do hífen para distinguir-se entre opções conflitantes. (Como exemplo, experimente o **ps -e** e **ps e**).

Além disso, uma opção frequentemente instrui o comando de que o item subsequente na linha de comando é um argumento específico. Por exemplo:

tar cf mytarfile file1 file2 file3
tar -cf mytarfile file1 file2 file3

Esses comandos equivalentes usam **tar** para criar um arquivo chamado **mytarfile** e colocar três arquivos (**file1**, **file2**, e **file3**) dentro dele. Neste caso, a opção **f** diz a tar que o arquivo chamado **mytarfile** segue logo após a opção.

Assim como qualquer linguagem natural contém exceções e variações, o mesmo ocorre com a sintaxe usada para os comandos GNU e Unix. Você não deverá ter problemas para aprender a sintaxe essencial dos comandos que precisará usar com mais frequência. As capacidades do conjunto de comandos oferecidos no Linux são amplas, fazendo com que seja altamente improvável que você vá memorizar todas as sintaxes dos comandos de que precisará. A maioria dos administradores de sistemas estão sempre aprendendo mais sobre recursos que jamais usaram, de comandos que usam regularmente. É uma prática padronizada consultar regularmente as páginas de manual (**manpages**) ou infopages, bem como outras documentações sobre comandos que você use. Portanto, sintá-se à vontade para explorar e aprender à medida que trabalha com os comandos.

E por fim, se você desejar digitar mais de um comando em uma única linha no terminal, use o sinal de ponto e vírgula para tal. Exemplo:

ls ; ps

No exame LPIC-1 a sintaxe dos comandos e o uso da linha de comando são assuntos muito importantes. Preste bastante atenção no uso de opções e argumentos e, em como eles são diferenciados. Esteja ciente também de que alguns comandos esperam que as opções estejam precedidas por um hífen, enquanto que outros não. **Os exames LPI não se concentram nas opções, portanto não pense que você precisa memorizar cada opção obscura de cada comando para os exames.**

Os shells modernos, como o bash, incluem um importante conjunto de recursos chamados de histórico, expansão e edição dos comandos. Usando estes recursos é fácil referir-se de volta a comandos anteriores e a sua sessão interativa no shell se torna muito mais simples e efetiva.

A primeira parte desse conjunto de comandos é o histórico. Quando o bash é executado interativamente, ele fornece acesso a uma lista de comandos digitados anteriormente. Os comandos são armazenados no histórico antes de qualquer interpretação pelo shell. Ou seja, são armazenados antes de os coringas serem expandidos ou de as substituições de comandos serem feitas. A lista do histórico é controlada pela variável de shell **HISTSIZE**.

Por padrão, **HISTSIZE** é definida para 500 linhas, mas você pode controlar esse número, bastando ajustar o valor de **HISTSIZE**. Além dos comandos digitados na sua sessão atual do bash, os comandos de sessão anteriores são armazenados por padrão em um arquivo chamado **~/.bash_history** (ou então o arquivo nomeado na variável de shell **HISTSIZE**).

Observação: Se você usar diversos shells em um ambiente de janelas (como praticamente todo mundo faz), o último shell a ser terminado escreverá o seu histórico em **~/.bash_history**. Por essa razão, você poderá desejar usar uma só chamada a shell para a maior parte do seu trabalho.

Para visualizar seu histórico de comandos no shell, digite o comando `history`. Note que haverá um número de linhas a cada comando. Esse número poderá ser usado posteriormente na expansão do seu histórico. Vejamos como isso funciona logo a baixo:

Designador	Descrição
<code>!!</code>	Pronunciado como "bang bang", este comando se refere ao comando mais recente. O ponto de exclamação é frequentemente chamado de "bang" em sistemas Linux e Unix.
<code>!n</code>	Refere-se ao comando número n do histórico. Use o comando <code>history</code> para exibir esses números.
<code>! -n</code>	Refere-se ao comando atual menos n do histórico.
<code>!string</code>	Refere-se ao comando mais recente que comece com <code>string</code> .
<code>!?string</code>	Refere-se ao comando mais recente que contenha <code>string</code> .
<code>^string1^string2</code>	Substituição rápida. Repete o último comando, substituindo a primeira ocorrência de <code>string1</code> por <code>string2</code> .

Embora o uso da substituição do histórico possa ser útil para se executar comandos repetitivos, a edição do histórico de comandos é muito mais interativa. Para visualizar o conceito de edição do histórico de comandos, pense em todo o seu histórico do bash (incluindo a parte obtida a partir do seu arquivo `~/.bash_history`) como conteúdo buffer de um editor. Nesse cenário, o prompt atual é a última linha de um buffer de edição, e todos os comandos anteriores no seu histórico estarão acima dele. Todos os recursos típicos de edição estão disponíveis com a edição do histórico de comandos, incluindo realizar movimentação dentro do "buffer", buscar, cortar, colar, e assim por diante.

Depois que você tiver se acostumado a usar o histórico de comando em forma de edição, tudo o que tiver feito na linha de comando se torna disponível em forma de texto recuperável e reutilizável para comandos subsequentes. Quanto mais você se familiarizar com esse conceito, mais útil ele será. Por padrão, bash usa vinculações de teclas parecidas com aquelas encontradas no editor Emacs para edição do histórico de comandos. A baixo algumas teclas atalho para te ajudar a manipular melhor o prompt shell:

Tecla	Descrição
C-p	Linha anterior (equivalente: seta para cima)
C-n	Próxima linha (equivalente: seta para baixo)
C-b	Voltar um caracter (equivalente: seta para esquerda)
C-f	Avançar um caracter (equivalente: seta para direita)
C-a	Começo da linha
C-e	Fim da linha
C-l	Limpa a tela (tipo o comando clean)
M-<	Início do histórico
M->	Fim do histórico
C-d	Apaga o caractere à direita
C-k	Apaga o texto do cursor até o fim da linha
C-y	Cola o texto anteriormente apagado com C-k
M-d	Apaga palavra
C-rtext	Busca reversa por texto
C-stext	Busca adiante por texto

O bash oferece uma útil capacidade de se fazer substituições de comandos. Esse recurso permite substituir o resultado de um comando por um script. por exemplo, sempre que **\$(comando)** for encontrado, a sua saída será substituída. Essa saída poderia ser atribuída a uma variável, como no número das linhas do arquivo .bashrc :

RCSIZE=\$(wc -l ~/.bashrc)

Uma outra forma de substituição de comandos é `comando`. O resultado é o mesmo, exceto pelo fato de que a sintaxe com aspas reversas tem algumas regras especiais referentes a meta-caracteres que a sintaxe \$(comando) evita.

Se você precisar repetir um comando ao longo de todas as ramificações de uma árvore de diretórios, a execução recursiva é bastante útil, mas pode também ser perigosa. Ela dá a um único comando interativo o poder de operar sobre uma área muito mais ampla, no seu sistema, do que o seu diretório atual, e é necessário exercer os cuidados apropriados. Pense duas vezes antes de usar esses recursos, particularmente quando estiver operando como o super usuário.

Alguns comando do GNU em sistema Linux possuem capacidades internas na forma de opções. Por exemplo, **chmod modifica permissões sobre arquivo do diretório atual:**

chmod g+w *.c

Neste exemplo, todos os arquivos com a extensão .c, no diretório atual, são modificados com a permissão de escrito de grupo. Entretanto, poderá haver uma série de diretórios e arquivos em hierarquias que requeiram essa modificação. O **chmod** contém a opção **-R** (com letra maiúscula mesmo pois, o shell é case sensitive ou seja, diferencia minúsculas de maiúsculas e você também pode usar ao invés de **-R** o comando **--recursive**) o qual instrui o comando a operar não somente nos arquivos e diretórios especificados na linha de comando, mas também em todos os arquivos e diretórios contidos sob os diretórios especificados. Por exemplo, este comando fornece a permissão de escrita de grupo a todos os arquivos em uma árvore de códigos-fontes chamada src:

chmod -R g+w src

Este comando descerá até cada um dos subdiretórios de **src** e adicionará a permissão requisitada a cada arquivo e diretório que encontrar. Outros exemplos de comandos com essa capacidade incluem **cp** (copiar), **ls** (listar arquivos) e **rm** (remover arquivos).

Uma abordagem mais geral para a execução recursiva através de um diretório é usar o comando **find**. O **find** é inerentemente recursivo e foi elaborado para descer através dos diretórios, para buscar arquivos com certos atributos ou para executar comandos. Na sua forma mais simples, find exibe uma hierarquia inteira de diretórios quando você simplesmente digita um comando com um diretório-alvo. (Se não for passado nenhum argumento para o **find**, ele exibirá todos os arquivos que encontrar pela frente).

find src -name "*.c" - Os arquivos em .c serão listados recursivamente

Explore a manpage do find e outros comandos citados anteriormente para aprender a usa-los com seus devidos argumentos e definições.

Arquivos interessantes de configuração do bash:

~/profile

O bash também possibilita que uma lista de comandos possa ser executada quando um login é feito no sistema. Cada usuário pode ou não possuir o arquivo .profile no seu diretório HOME.

~/bashrc

o .bashrc pode também conter uma lista de comandos, mas sua execução ocorre toda vez que o bash é executado.

Obtendo ajuda

O diretório /usr/share/doc possui grande parte da documentação de uma distribuição Linux. O subdiretório /usr/share/doc/howto possui quase 2000 páginas de documentação detalhando passo a passo diversas tarefas. No subdiretório /usr/share/doc/packages você pode encontrar informações sobre quase todos os pacotes de software instalados no sistema.

O comando info

O comando info foi desenvolvido pelo projeto GNU e cobre grande parte das ferramentas. Para ter acesso à esta documentação você utiliza “info” antes do comando, exemplo:

```
info ls
```

Para passar para à próxima página do conteúdo digite “N” (next) e para voltar digite “P” (Previous). A tecla “Q” (quit) sai do info.

O comando man

O comando man (Manual) acompanha quase todos os programas Gnu/Linux. Ele traz uma descrição básica do comando e detalhes sobre o seu funcionamento. Exemplo:

```
man ls
```

Os documentos de manual são divididos em sessões de acordo com o assunto abordado. As sessões são numeradas de 1 a 9, a seguir:

1. Programas executáveis ou comandos de shell
2. Chamadas de sistema (funções fornecidas pelo Kernel)
3. Chamadas de Biblioteca (funções fornecidas pelas bibliotecas)
4. Arquivos especiais, especialmente aqueles localizados em /dev
5. Formato de arquivos e convenções
6. Jogos
7. Pacotes de macro
8. Comandos administrativos
9. Rotinas do Kernel

Rodando o comando:

```
man passwd
```

Você irá visualizar a página de manual do comando (1), se quiser, visualizar um manual do arquivo de configuração do comando `passwd` utilize:

```
man 5 passwd
```

Neste caso ele retorna o manual de `/etc/passwd`

Utilizando o Help On-line

O recurso de ajuda rápida é útil para se saber quais opções podem ser usadas com um determinado comando. Quase todos os programas GNU/Linux oferecem este recurso para consultas rápidas. Para acionar o help on-line, digite: comando `-help`

Utilizando filtros de Texto

O Linux tem diversas ferramentas para trabalhar e transformar arquivos de texto puro (sem formatação especial). Estas ferramentas são úteis quando trabalhamos com scripts, verificação de logs, etc.

Cut

Exibe na saída padrão, ou seja, no terminal, colunas ou campos de um ou mais arquivos. O arquivo original não é modificado. É útil quando se precisa de uma fatia vertical de um arquivo. O delimitador padrão é o **tab**

Sintaxe:

```
cut opções [arquivos]
```

Opções mais utilizadas:

-b lista-bytes

Mostra apenas os *bytes* nas posições listadas em *lista-bytes*. Tabs e backspaces são tratados como qualquer outro caractere, já que ocupam 1 *byte*.

-c lista-caracteres

Mostra apenas os *caracteres* nas posições listadas em *lista-caracteres*. Por enquanto, seu efeito é idêntico ao de `-b`, mas a internacionalização e o uso de caracteres **unicode** vai mudar isso, já que caracteres internacionais podem ocupar mais de 1 *byte*, mas continuam sendo apenas 1 *caractere*.

-f lista-campos

Mostra apenas os campos listados em *lista-campos*. O delimitador padrão para os campos é o caractere **tab**.

-d delimitador

Deve ser usado em conjunto com a opção *-f*. Define o delimitador de campos.

-n

Não quebrar caracteres com mais de 1 *byte*. Esta opção não é usada no momento. Terá efeito apenas com a internacionalização.

lista-bytes, *lista-caracteres* e *lista-campos* pode ser um número apenas, uma lista de números separada por vírgula ou um intervalo separado por hífen (-).

fmt

Formata o texto em uma largura de, no máximo, um dado número de caracteres (75 por padrão). Se for informado mais de um arquivo, eles serão concatenados na saída.

Sintaxe:

```
fmt opções [arquivos]
```

Opções mais utilizadas:

-u

Usa espaçamento uniforme. Um espaço entre palavras e dois entre sentenças.

-w largura

Define a *largura* para um número diferente do padrão, 75.

head

Exibe as primeiras (10, por padrão) linhas de um arquivo. Se mais de um arquivo for informado, as primeiras linhas de cada um serão exibidas separadamente.

Sintaxe:

```
head opções [arquivos]
```

Opções mais utilizadas:

-c n

Exibe os primeiros *n* bytes de cada arquivo. Pode-se usar os sufixos *k* ou *m*, significando:

kilobytes e *megabytes*, respectivamente.

-n

Exibe as primeiras *n* linhas de cada arquivo. O padrão é 10.

Exemplo:

```
head -2 /etc/passwd /etc/protocols
==> /etc/passwd <==
root:x:0:0:root:/root:/bin/bash
grubelilo:x:100:100:grubelilo:/home/grubelilo:/bin/bash
==> /etc/protocols <==
ip 0 IP
icmp 1 ICMP
Join
```

join

O comando join une as linhas de ambos os arquivos que tenham um índice comum. O comando join poderá ser utilizado como um banco de dados simples.

As opções frequentemente utilizadas são:

- j1 número:** Escolhe o campo número como índice para o arquivo1
- j2 número:** Escolhe o campo número como índice para o arquivo2

Para exemplificar, crie um arquivo com o conteúdo:

```
1 cor 1
2 cor 2
3 cor 3
```

Salve como arquivo1, crie outro arquivo com o conteúdo:

```
1 branco
2 preto
3 laranja
```

Salve como arquivo2.

Agora execute o comando:

```
join -j 1 arquivo1 arquivo2
```

O resultado do comando:

```
1 cor 1 branco
2 cor 2 preto
3 cor 3 laranja
```

nl

Numera as linhas de um ou mais *arquivos*, concatenando-os na saída. Pode ser usada uma marcação especial para delimitação de cabeçalho, corpo e rodapé. Cabeçalho e rodapé são por padrão excluídos da numeração.

A numeração é feita para cada *página lógica*, definida por ter um cabeçalho, um corpo e um rodapé.

A marcação especial é a seguinte:

- **\:\:** para o cabeçalho
- **\:** para o corpo
- **\:** para o rodapé

Sintaxe:

```
nl [opções] [arquivos]
```

Exemplo de criação de um arquivo para o nl:

```
\:\:  
Cabeçalho
```

```
\:  
linha1  
linha2
```

```
\:  
rodapé
```

Usando o comando:

```
nl arquivo1
```

Obtemos o seguinte resultado:

```
Cabeçalho
```

```
1 linha1  
2 linha2
```

```
rodapé
```

pr

Converte um arquivo de texto em uma versão paginada, com cabeçalhos (contendo o nome do arquivo, data e hora e número de página). Pode ser bem útil para preparar arquivos texto para impressão. O número de linhas padrão de cada página é 66.

Sintaxe:

```
pr [opções] [arquivo]
```

Opções mais usadas:

-d

Duplo espaço entre linhas

-h *cabeçalho*

usa *cabeçalho* ao invés do nome do arquivo no cabeçalho de cada página

-l *linhas*

Define o número de linhas de cada página. O padrão é 66

-o *largura*

Define a margem esquerda para *largura* colunas

split

O comando split é utilizado para dividir grandes arquivos em arquivos menores de acordo com sua definição na entrada do comando. Exemplo:

Digamos que você possua o seguinte arquivo de nome frutas1:

```
1 banana  
2 maça  
3 laranja  
4 melancia  
5 uva  
6 tangerina
```

Se você executar o comando:

```
split -3 fruta1 saida1
```

O resultado será 2 arquivos gerados de nomes:

```
saida1aa  
saida1ab
```

E o seguinte conteúdo, no “saida1aa”:

```
1 banana
2 maça
3 laranja
```

E no “saida1alb”:

```
4 melancia
5 uva
6 tangerina
```

Ou seja, você criou dois arquivos com 3 linhas.

tac

O comando **tac** mostra o conteúdo de um ou mais arquivos de trás para frente.
Exemplo:

```
tac nome_do_arquivo
```

tail

O comando **tail** visualiza as últimas 10 linhas de um arquivo. Diferente do comando **head** que visualiza as 10 primeiras linhas de um arquivo.

Exemplo:

```
tail -n 50 /var/log/messages
```

Mostra as últimas 50 linhas do arquivo.

A opção **-f** mostra as últimas linhas finais de um arquivo continuamente enquanto outro processo grava mais linhas, muito útil para arquivos de log, exemplo:

```
tail -f /var/log/messages
```

tr

Traduz caracteres de string1 para os caracteres de string2. Exemplos:

```
cat arquivo1 | tr a-z A-Z
```

Deixa todos os caracteres de minúsculos para maiúsculos.

```
cat arquivo1 | tr -d a
```

O comando **tr** apaga a letra “a “

wc

Não, não é o banheiro. O comando `wc` conta as linhas, palavras e caracteres de um ou mais arquivos. Se mais de um arquivo for passado como argumento, ele irá apresentar as estatísticas de cada arquivo e também o total.

As opções mais frequentes são:

-c

Conta o número de caracteres de um ou mais arquivos

-l

Conta o número de linhas de um ou mais arquivos

-l

Exibe apenas a contagem de linhas

-w

Exibe apenas a contagem das palavras

Exemplo:

```
wc /etc/apt/sources.list
```

od

O comando `od` é utilizado para visualizarmos o conteúdo de um arquivo nos formatos hexadecimal, octal, ASCII e nome de caracteres.

As opções frequentemente utilizadas são:

-t tipo

Especifica o tipo de saída que o comando `od` deve gerar, exemplo:

```
od -t x arquivo.txt
```

Exibe o conteúdo de `arquivo.txt` em hexadecimal, os tipos são:

- a - Nome de caractere
- c - ASCII
- o - Octal
- x - Hexadecimal

hexdump

O comando hexdump substitui o comando od, imprimindo a saída padrão o conteúdo de um arquivo nos formatos ASCII, hexadecimal e octal.

Para exemplificar vamos criar um arquivo:

```
echo "Grubelilo" > apelido
```

Agora vamos visualizar o conteúdo do arquivo apelido (grubelilo) em formato byte:

```
hexdump -b apelido
```

O resultado:
0000000 147 162 165 142 145 154 151 154 157 012
000000^a

Imprime o arquivo em caractere de um byte:

```
hexdump -c apelido
```

imprime o arquivo em hexadecimal e em ASCII

```
hexdump -C apelido
```

uniq

Mostra o conteúdo de arquivos suprimindo linhas sequenciais repetidas.

paste

Concatena arquivos lado a lado. Exemplo:

```
paste texto_simples texto_simples
```

texto simples
criado com cat

texto simples
criado com cat

sort

Ordena alfabeticamente. Com a opção **-n** ordena numericamente e **-r** inverte o resultado.

```
sort texto_simples
```

criado com cat
texto simples

Diretórios e Arquivos

Arquivos podem ser acessados tanto por seu caminho **absoluto** quanto seu caminho **relativo**. Caminhos absolutos são aqueles iniciados pela barra da raiz (/), e caminhos relativos são aqueles que tomam por referência o diretório atual. O ponto “.” refere-se ao diretório atual, e “..” refere-se ao diretório contendo o diretório atual.

ls

O comando **ls** é usado para listar arquivos e conteúdo de um diretório. A opção **-l** exibe detalhes sobre o(s) arquivo(s), **-s** mostra o tamanho em *kilobytes* e **-d** mostra o diretório, e não seu conteúdo.

Exemplo de saída de ls -l:

```
ls -l /etc/X11/xinit/
```

```
total 20
-rw-r--r--    1 root root 321 2006-01-14 17:33 README.Xmodmap
lrwxrwxrwx    1 root root  15 2006-03-20 22:31 xinitrc -> xinitrc.fluxbox
-rwxr-xr-x    1 root root 556 2003-03-16 19:59 xinitrc.blackbox
-rwxr-xr-x    1 root root 560 2006-03-07 03:32 xinitrc.fluxbox
-rwxr-xr-x    1 root root 799 2006-01-14 17:30 xinitrc.twm
-rwxr-xr-x    1 root root 788 2005-07-21 15:27 xinitrc.wmaker
```

A primeira coluna mostra o tipo e as permissões do arquivo, a segunda coluna mostra o número de links físicos (hard links) para o arquivo, a terceira e a quarta mostram o dono e o grupo aos quais o arquivo pertence, a quinta mostra o tamanho em *bytes*, a sexta e a sétima mostram a data e a hora da última modificação do arquivo e a oitava coluna mostra o nome do arquivo. Se o arquivo for um link simbólico, uma seta mostra o arquivo para o qual ele aponta.

cp

O comando **cp** é utilizado para copiar arquivos. Suas opções principais são:

- i** → Modo interativo. Pergunta antes de sobrescrever um arquivo.
- p** → Copia também os atributos do arquivo original.
- r** → Copiar recursivamente o conteúdo do diretório de origem.

É importante saber que quando copiando um diretório recursivamente, o uso da barra “/” no final do diretório de origem fará com que apenas o conteúdo do diretório seja copiado/movido para o destino e o não uso da barra fará com que o diretório de origem e seu conteúdo sejam copiados no destino.

Exemplo de cp:

```
cp tux.xcf icons/
```

Copia o arquivo “tux.xcf” para o diretório “icons” no diretório atual.

mv

O comando **mv** move e renomeia arquivos. Usado com a opção **-i** pede por confirmação antes de sobrescrever um arquivo de destino.

Exemplo de mv:

```
mv imagem.jpg ../fotos/
```

Move o arquivo “imagem.jpg” para o diretório “fotos” que está no mesmo diretório onde está contido o diretório atual (um nível acima).

touch

Para alterar a data de um arquivo, utiliza-se o comando **touch**. Usado sem argumentos, touch altera a data e a hora de criação e modificação de um arquivo para os valores atuais do sistema. Para alterar apenas a data de modificação, usa-se a opção **-m**, e para alterar apenas a data de acesso, usa-se a opção **-a**. O argumento passado com **-t** usa outro valor de tempo:

Mudar a data e hora para janeiro, 01 - 00:01

```
touch -t '01010001' texto_simples
```

pwd

No bash, o comando para retornar o diretório atual é o **pwd**. O comando **cd** muda para o diretório especificado ou vai para o diretório pessoal quando nenhum diretório for especificado.

mkdir

O comando **mkdir** cria diretórios. Para criar uma árvore de diretórios recursivamente, usa-se a opção **-p**:

```
mkdir -p caminho/completo/para/dir
```

Para alterar as permissões do diretório no ato da criação, as mesmas são transmitidas ao mkdir com a opção **-m**.

Criar diretório sem qualquer tipo de acesso para o grupo ou outros:

mkdir -m 700 exclusivo

rmdir

Diretórios vazios podem ser apagados pelo comando **rmdir**. Para apagar uma árvore de diretórios vazios, usa-se a opção **-p**. Para apagar diretórios com conteúdo, usa-se **rm -r**, e para forçar a remoção, a opção **-f** é utilizada.

find

Encontrando Arquivos com o Comando find

Sintaxe do find:

```
find diretório critério [-exec comando {} \;]
```

O argumento *diretório* indica onde o find deve iniciar a busca e *critério* pode ser o nome do arquivo /diretório a ser procurado e/ou uma regra para a busca.

Critérios comuns para o find:

-type

Tipo do arquivo (**d** para diretório, **f** para arquivo comum e **l** para *link*)

-name *nome*

Nome do arquivo

-user *usuário*

Dono do arquivo

-atime *-/+n*

Data de último acesso ao arquivo, menor ou maior que *n*. *n* corresponde a *n**24 horas.

-ctime *-/+n*

Data de criação do arquivo, menor ou maior que *n*. *n* corresponde a *n**24 horas.

-mtime *-/+n*

Data de modificação do arquivo, menor ou maior que *n*. *n* corresponde a *n**24 horas.

-amin *-/+n*

Data de último acesso ao arquivo, menor ou maior que *n*. *n* corresponde a minutos.

-cmin *-/+n*

Data de criação do arquivo, menor ou maior que *n*. *n* corresponde a minutos.

-mmin *-/+n*

Data de modificação do arquivo, menor ou maior que *n*. *n* corresponde a minutos.

-newer *arquivo*

O arquivo procurado foi criado/modificado mais recentemente que *arquivo*.

-perm *modo*

O arquivo procurado tem permissão *mode*

-perm *-modo*

O arquivo procurado tem todos as permissões listadas em *mode*
Exemplo: Encontrar todos os arquivos do tipo link em /usr/lib criados há menos de 24 horas:

```
find /usr/lib -type l -ctime -1
```

```
/usr/lib/libssl.so
/usr/lib/libcrypto.so
/usr/lib/libssl.so.0
/usr/lib/libcrypto.so.0
```

Caracteres Coringa (file globbing)

As operações com arquivos e diretórios permitem o uso de caracteres coringa, que são padrões de substituição de caracteres.

O caracter “ * ” substitui qualquer sequência de caracteres:
 Exemplo:

```
ls /etc/host*
```

```
/etc/host.conf
/etc/hosts
/etc/hosts.allow
/etc/hosts.deny
/etc/hosts.equiv
```

O caracter “ ? ” substitui apenas um caractere:

```
ls /dev/fd?
```

```
/dev/fd0 /dev/fd1 /dev/fd2 /dev/fd3
```

O uso de colchetes “[] ” indica uma lista de caracteres:

```
ls /dev/hd[abc]
```

```
/dev/hda /dev/hdb /dev/hdc
```

Chaves “ { } ” indicam uma lista de termos separados por vírgula:

```
ls /dev/{hda,fd0}
```

```
/dev/fd0 /dev/hda
```

O uso de exclamação antes de um coringa o exclui da operação:

```
ls /dev/fd[!01]
```

```
/dev/fd2 /dev/fd3
```

Coringas precedidos de barra invertida “ \ ” não desempenham função substitutiva:

```
ls /dev/fd[!\01\]
```

ls: /dev/fd[!01]: Arquivo ou diretório não encontrado

Entre aspas duplas, apenas os caracteres especiais “|”, “/” e “\$” têm efeito. Entre aspas simples apenas o caractere especial “\” tem efeito.

Empacotadores e compressores de dados

O comando tar (tape-Archive) lê arquivos e diretórios e salva em fita de arquivo. Juntamente com os dados ele salva informações importantes como a última modificação, permissões de acesso e outros.

O comando tar recebe dois argumentos, são eles:

Fonte: Pode ser um arquivo, diretório ou dispositivo a ser copiado

Destino: Destino para os dados

As opções mais frequentes são:

-c

Cria um novo arquivo .tar

-t

Lista o conteúdo de um arquivo .tar

-x

Extrai os arquivos .tar

-u

Adiciona mais arquivos ao arquivo.tar somente se estes forem novos ou modificados.

-r

Adiciona os arquivos especificados no final do arquivo .tar

-g

Cria um backup incremental

-j

Utiliza o bzip2 para compactar e descompactar arquivos .tar

-z

Utiliza o gzip para compactar e descompactar arquivos .tar

-p

Extrai os arquivos com as mesmas permissões de criação

-v

Lista todos os arquivos processados

-f

Indica que o destino é um arquivo em disco e não uma unidade de fita

-T

Cria um pacote .tar a partir de uma lista de arquivos e diretórios

Exemplo:

Para salvar o diretório /var/lib/mysql em um arquivo utilize:

```
tar -cvzf /var/lib/mysql.tar.gz /var/lib/mysql
```

Para extrair este pacote criado

```
tar -xvzf /var/lib/mysql.tar.gz
```

Os compressores

Para maior eficiência e economia de discos de backup existe o recurso de compressão de dados, no linux vamos tratar do gzip.

Para compactar um arquivo, utilize:

```
gzip arquivo
```

Para descompactar um arquivo

```
gzip -d arquivo.gz
```

ou

```
gunzip arquivo.gz
```

O bunzip2 também é uma poderosa ferramenta de compactação de arquivos.

Para compactar um arquivo:

```
bzip2 -d arquivo
```

Para descompactar um arquivo

```
bzip2 -d arquivo.bz2
```

ou

```
bunzip2 arquivo.bz2
```

“ O bzip2 é considerado o melhor em compressão de dados “.

Utilizando o comando dd

O comando dd copia um arquivo, discos e partições para um arquivo, discos e dispositivos.

Para gerar uma imagem ISO a partir de um CD/DVD, utilize:

```
dd if=/dev/cdrom of=imagem.iso
```

Para gravar a imagem ISO em um CD/DVD

```
dd if=imagem.iso of=/dev/cdrom
```

Para clonar um HD:

Suponhamos dois Hds, um instalado no primary master (/dev/hda) e o outro no primary slave (/dev/hdb)

```
dd if=/dev/hda of=/dev/hdb
```

Como a cópia é feita bit a bit, não importa qual o sistema operacional, nem o sistema de arquivos usado no HD de origem. A cópia é completa, incluindo a tabela de partição do HD e o setor de boot.

Backup da MBR

Criar o backup

```
dd if=/dev/sda of=mbr.backup bs=512 count=1
```

Restaurar o backup

```
dd if=mbr.backup of=/dev/sda bs=512 count=1
```

Executando e eliminando processos

Em linhas gerais, um processo é um programa em execução. Cada processo possui um número único de identificação chamado **PID**. Este número pode ser usado para mudar a prioridade de um processo ou para finalizá-lo. O processo que inicia um outro processo é chamado o **processo pai** (parent process) do segundo, assim como o segundo é chamado **processo filho** (child process) do primeiro.

pstree

Mostra processos ativos em formato de árvore genealógica (processos filhos ligados aos respectivos processos pais).

Opções comuns:

- p → Inclui PIDs dos processos.
- h → Lista apenas os processos do usuário atual.

ps

Mostra os processos de maneira detalhada.

Opções comuns:

- ux** → Mostra todos processos do usuário.
- aux** → Mostra todos processos no sistema.
- u → Mostra apenas processos cujo dono seja o indicado pela opção -u.
- g → Mostra apenas processos cujo grupo seja o indicado pela opção -g.

top

Monitora continuamente os processos, mostrando informações como uso de memória e CPU de cada processo. A tecla “h” fornece ajuda sobre o uso do programa. Pode ser usado para alterar a prioridade de um processo

pidof

Retorna o número PID para o programa solicitado, se o programa estiver em execução. PID do daemon crond:

```
pidof crond
```

```
2404
```

kill

Envia sinais de controle para processos. O sinal padrão quando nenhum sinal é informado é **SIGTERM**, de valor numérico **15**, que pede ao programa em questão para finalizar. O processo não necessariamente obedece ao sinal, a menos que o sinal seja **SIGKILL**. Em alguns casos, o sinal **SIGHUP** pode ser interpretado como ordem para o processo reler seu(s) arquivo(s) de configuração.

Exemplo de kill:

```
kill -SIGTERM 4902
```

Envia o sinal *SIGTERM* para o processo de PID 4902

Sinais comuns:

```
SIGHUP
```

Termina ou reinicia o processo. Valor numérico **1**.

```
SIGINT
```

Interrompe o processo, igual a Ctrl+c. Valor numérico: **2**.

```
SIGQUIT
```

Fecha o processo. Valor numérico: **3**.

```
SIGKILL
```

Força a finalização o processo. Valor numérico: **9**.

```
SIGTERM
```

Pede ao processo para finalizar. Valor numérico: **15**.

killall

Tem função igual a de kill, porém usa o *nome* do processo no lugar do PID. Com a opção **-l** lista os sinais possíveis.

```
killall -SIGHUP cupsd
```

Neste exemplo, o daemon cupsd irá reler seus arquivos de configuração.
Tarefas em Primeiro e Segundo Plano

Após iniciado um programa no shell, este na maioria das vezes assumirá o controle de *stdin* e *stdout*, ou seja, ficará em *primeiro plano*. Para interromper o programa e voltar ao prompt do shell, usa-se a combinação de teclas **Ctrl+z**.

Feito isso, para continuar a execução do programa em *segundo plano*, ou seja, mantendo o prompt do bash em primeiro plano, usa-se o comando **bg** (*background*).

Para continuar a execução do programa em primeiro plano usa-se o comando **fg** (*foreground*).

```
ogg123 -q 03-geraldo_vandré-fica_mal_com_deus.ogg
```

O número mostrado antes do comando identifica a tarefa. Este número é passado para `fg` e `bg` para especificar a tarefa desejada. Se houver apenas uma tarefa na sessão atual, `fg` e `bg` podem ser usados sem argumentos.

Para usar o nome do programa no lugar de seu número de tarefa, basta precedê-lo por `%?`:

```
bg %?ogg123
```

jobs

O comando **jobs** lista as tarefas existentes na sessão atual do bash. É especialmente útil quando há mais de uma tarefa em andamento.

É possível iniciar programas diretamente em segundo plano, adicionando o símbolo “&” ao seu final.

```
ogg123 -q 10-geraldo_vandré-berimbau.ogg &  
[2] 5375
```

Os números mostrados correspondem ao número da tarefa e ao PID, respectivamente.

Quando um usuário sai do sistema, um sinal **SIGHUP** é enviado a todos os processos à ele pertencentes. Para que esse sinal não interrompa o processo do usuário após este sair do sistema, o comando deve ser invocado através do

nohup:

```
nohup wget ftp://transferência/muito/demorada.tar.bz2 &  
nohup: appending output to `nohup.out'
```

As saídas *stdout* e *stderr* serão redirecionadas para o arquivo **nohup.out**, criado no mesmo diretório em que o comando foi executado.

Prioridade de execução

O kernel do linux possui como uma de suas atividades críticas o escalonador de processos. O escalonador é um algoritmo especial que coloca em fila todos os processos em execução e decide qual programa processo será executado e durante quanto tempo. O escalonador é o que permite que o computador possa executar mais de um processo em concorrência, dividindo a CPU em fatias de tempo de execução.

top

O comando top mostra os processos em execução como o comando ps, mas atualizando a tela. Este recurso é útil para monitorarmos como um ou mais processos agem no sistema.

Este comando ordena os processos que utilizam mais memória no topo da tela.

-b

Executar em modo batch ou arquivo em lote. Utilizado para direcionarmos a saída do comando para outros processos ou arquivo-fonte

-d n

Determina o tempo das atualizações da tela em segundos (n).

-n num

Mostra o número de vezes na tela dos processos em execução e depois termina.

-q

Executar com atualizações em tempo real. Esta opção deve ser utilizada com cautela. Pois pode consumir muita quantidade de CPU.

-u

Monitora os processos em execução de um determinado usuários

Exemplo:

```
top -u grubelilo
```

Modificando a prioridade de execução dos processos

É possível alterar a prioridade de execução dos processos, através dos comandos “nice” e “renice”. Estes comandos são extremamente úteis em ambientes multiusuários, onde é preciso dar mais ou menos fatias de processamento aos diversos programas dos usuários. Para manter um ambiente estável e organizado.

O ajuste de prioridade possível vai de -20 (Maior prioridade) até o 19 (menos prioridade).

Exemplo:

```
nice -n -10 tar czf home_backup.tar.gz /home
```

Se não for passado nenhum valor de ajuste o comando nice ajustará a prioridade para +10 diminuindo o tempo do processo:

```
nice tar czf home_backup.tar.gz /home
```

renice

O comando `renice` ajusta a prioridade de execução de processos que já estão rodando. Por padrão o comando `renice` recebe como parâmetro o PID de um determinado processo. O ajuste de prioridade é um número inteiro que vai do -20 (maior prioridade) ao +20 (executar qualquer coisa antes deste processo).

-p

Recebe um PID para alterar sua prioridade

-u

Recebe um nome de usuário para alterar a prioridade de todos os seus processos

-g

Recebe um nome de grupo para alterar todos os processos pertencentes a este grupo.

Exemplo:

```
renice -1 987 -u root -p 32
```

Neste exemplo o Processo de PID 987 e PID 32 além de todos processos de root terão prioridade -1.

Expressões Regulares e grep

Expressões regulares são elementos de texto, palavras chave e modificadores que formam um padrão, usado para encontrar e opcionalmente alterar um padrão correspondente. Muitos programas suportam o uso de expressões regulares. O comando **grep** é o mais comum para realizar buscas por expressões regulares em textos.

Caracteres especiais em expressões regulares:

^ → Começo de linha

\$ → Fim de linha

. → Qualquer carácter

***** → Qualquer sequência de zero ou mais caracteres

[] → Qualquer carácter que esteja presente nos colchetes

Exemplo: Mostrar apenas as linhas que começam com o carácter “ #” do arquivo /etc/lilo.conf.

```
grep '^#' /etc/lilo.conf
```

Mostrar linhas de `/etc/lilo.conf` que contenham o termo “hda” ou “hdb”:

```
grep 'hd[ab]' /etc/lilo.conf
```

Mais opções de `grep`:

- c → Conta as linhas contendo o padrão.
- i → Ignora a diferença entre maiúsculas e minúsculas.
- f → Usa a expressão regular contida no arquivo indicado por essa opção.
- n → Procurar somente na linha indicada por essa opção.
- v → Mostra todas as linhas exceto a que corresponder ao padrão.

Exemplo: Mostrar apenas linhas contendo caracteres (inverter a busca por linhas que não possuam caracteres)

```
grep -v '^$' /etc/lilo.conf
```

sed

O comando `sed` é mais utilizado para procurar e substituir padrões em textos, mostrando o resultado em `stdout`. Sintaxe do `sed`:

```
sed [opções] 'comando e expressão regular' [arquivo original]
```

No `sed`, a expressão regular fica circunscrita entre barras “/”.

Exemplo:

```
sed -e '/^#/d' /etc/lilo.conf
```

Mostra o arquivo `/etc/lilo.conf` sem linhas começadas por “#” (linhas de comentário). A letra “d” ao lado da expressão regular é um comando `sed` que indica a exclusão de linhas contendo o respectivo padrão.

Exemplo: substituir o termo “hda” por “sdb”:

```
sed -e 's/hda/sdb/g' /etc/lilo.conf
```

Exemplo: Entre as linhas 14 e 21, substituir a palavra “runlevel” por “rl”:

```
sed -n -e '14,21s/runlevel/rl/p' /etc/inittab
```

Opções comuns de sed:

- e → Executa a expressão e comando a seguir.
- f → Lê expressões e comandos do arquivo indicado pela opção.
- n → Não mostrar as linhas que não correspondam a expressão.
- s → Substituir.
- d → Apagar a linha.
- r → Insere o conteúdo do arquivo indicado na ocorrência da expressão.
- w → Escreve a saída no arquivo indicado.
- g → Substitui todas ocorrências da expressão na linha atual.

Editor de texto VI

Na maioria das distribuições o **vi** é o editor de textos padrão. O vi localiza-se em /bin. Há três modos básicos no vi:

Modo de comando

É o modo inicial do vi. Serve para navegação e edição. Geralmente, os comandos são letras únicas. Se precedido por número, o comando será repetido correspondentemente ao valor desse número.

Modo de última linha ou coluna

Acessível ao apertar a tecla “:” no modo de comando. Usado para fazer buscas, salvar, sair, executar comandos no shell, alterar configurações do vi, etc. Para retornar ao modo de comando, usa-se o comando “visual”.

Modo de inserção

A maneira mais comum de entrar no modo de inserção é apertando a tecla “i” ou “a”. É o modo mais intuitivo, usado para digitar texto no documento. A tecla ESC sai do modo de inserção e volta para o modo de comando.

Algumas teclas comuns usadas no modo de comando:

- O , \$** → Início e fim de linha
- 1G , G** → Início e fim de documento
- b , e** → Início e fim de palavra
- (,)** → Início e fim de sentença
- { , }** → Início e fim de parágrafo
- w , W** → Pular palavra e pular palavra contando com a pontuação
- h,j,k,l** → Esquerda, abaixo, acima, direita
- / , ?** → Busca para frente e para trás
- i** → Entra no modo de inserção na posição atual do cursor
- a , A** → Entra no modo de inserção depois do cursor ou no fim da linha
- o , O** → Adiciona linha e entra no modo de inserção depois ou antes do cursor
- s , S** → Apaga item ou linha e entra no modo de inserção
- c** → Modifica um item através de inserção de texto
- r** → Substitui um único caracter
- x** → Apaga um único caracter
- y , yy** → Copia um item ou toda linha
- p , P** → Cola o conteúdo copiado depois ou antes do cursor
- u** → Desfazer

ZZ → Fecha e salva se necessário
ZQ → Fecha e não salva
 Comandos do modo de última linha ou coluna
:! → Chama um comando do shell
:quit ou **:q** → Fecha
:quit! ou **:q!** → Fecha sem gravar
:wq → Salva e fecha
:exit ou **:x** ou **:e** → Fecha e grava se necessário
:visual → Volta para o modo de comando

O editor VIM é uma evolução do VI

Busca

/texto: Procura pela palavra “texto” do início para o fim do arquivo
?texto: Procura pela palavra “texto” do fim para o início do arquivo
/Mari[oa]: Procura por “Mario” ou “Maria”
^<pal: Procura por expressões que comecem com “pal” como, “palavra” ou “palíndromo”
/ismo\>: Procura por expressões que terminem com “ismo” como, “autismo”
^<...\>: Procura por todas as palavras com 3 letras
/maria\|joao: Procura por maria ou joao
^<\d\d\d\d\>: Procura exatamente por 4 dígitos numéricos
/^\n{3}: Procura por três linhas em branco
:bufdo /palavra/: Procura “palavra” em todos os arquivos abertos

Substituição

:%s/antigo/novo/g: Substitui todas as ocorrências de “antigo” por “novo” no arquivo
:%s/antigo/novo/gw: Substitui todas as ocorrências com confirmação
:2,35s/antigo/novo/g: Substitui todas as ocorrências entre as linhas 2 e 35
:5,\$s/antigo/novo/g: Substitui todas as ocorrências da linha 5 até o fim da linha
:%s/^/legal/g: Substitui o começo de cada linha com “legal”
:%s\$/Oh/g: Substitui o fim de cada linha por “Oh”
:%s/antigo/novo/gi: Substitui “antigo” por “novo” desconsiderando maiúsculas e/ou minúsculas
:%s/ *\$/g: Apaga todos os espaços em branco
:g/palavra/d: Apaga todas as linhas contendo “palavra”
:v/palavra/d: Apaga todas as linhas que não contém “palavra”
:s/maria/joao/: Substitui a primeira ocorrência de “maria” por “joao” na linha corrente
:s/maria/joao/g: Substitui todas as ocorrências de “maria” por “joao” na linha corrente

:%s/maria/joao/g: Substitui “maria” por “joao” em todo o arquivo
:%s#<[^>]\+>##g: Apaga tags HTML mas mantém o texto
:%s/^\(.*\)\n\1\$/\1/: Apaga linhas repetidas
Ctrl+a: Incrementa o número sob o cursor

Ctrl+x: Decrementa o número sob o cursor

ggVGg?: Muda o texto usando Rot13

Minúsculo/Maiúsculo

Vu: Torna todos os caracteres da linha minúsculos

VU: Torna todos os caracteres da linha maiúsculos

g~~: Inverte os caracteres do texto inteiro

vEU: Coloca as letras da palavra em maiúsculas

vE~~: Inverte os caracteres da palavra selecionada

guG: Coloca todo o texto em minúsculas do ponto em que está até o fim do arquivo

gUG: Coloca todo o texto em maiúsculas do ponto em que está até o fim do arquivo

:set ignorecase: Ignora minúsculos/maiúsculos nas buscas

:set smartcase: Ignora minúsculos/maiúsculos em buscas exceto quando uma letra maiúscula é usada

:%s/\<./\u&/g: Coloca a primeira letra de cada palavra em maiúscula

:%s/\<./\l&/g: Coloca a primeira letra de cada palavra em minúscula

:%s/.*\u&: Coloca a primeira letra de cada linha em maiúscula

:%s/.*\l&: Coloca a primeira letra de cada linha em minúscula

Lendo/Gravando arquivos

:1,10 w arquivo: Salva as linhas de 1 a 10 em “arquivo”

:1,10 w >> arquivo: Adiciona as linhas de 1 a 10 em “Arquivo”

:r arquivo: Insere o conteúdo de “arquivo” no atual

:23r arquivo: Insere o conteúdo de “arquivo” a partir da linha 23

Explorando arquivos

:e .: Abre o gerenciador de arquivos integrado do Vim

:Sex: Divide a janela e abre o gerenciador de arquivos integrado

:browse e: Abre o gerenciador de arquivos integrado na janela corrente

:ls: Lista os buffers carregados

:cd ..: Move para a pasta superior

:args: Lista os arquivos

:args *.php: Abre lista de arquivos

:grep expressao *.php: retorna uma lista de arquivos .php que contenham a expressão informada

gf: Abre o arquivo sob o cursor

Interação com o Linux

!:pwd: Executa o comando “pwd” e retorna para o Vim

!!pwd: Executa o comando “pwd” e insere a saída no buffer

:sh: Retorna temporariamente para o shell

exit: Retorna para o Vim

Alinhamento

:%!fmt: Alinha todas as linhas

!}fmt: Alinha todas as linhas a partir da posição corrente

5!!fmt: Alinha as próximas 5 linhas

Abas

:tabnew: Cria uma nova aba

gt: Mostra a próxima aba

:tabfirst: Mostra a primeira aba

:tablast: Mostra a última aba

:tabm n(posicao): Reorganiza as abas

:tabdo %s/foo/bar/g: Executa um comando em todas as abas

:tab ball: Coloca todos os arquivos abertos em abas

Divisão da janela do Vim

:e arquivo: Edita “arquivo” na janela corrente

:split arquivo: Divide a janela e abre “arquivo”

ctrl-w “seta para cima”: Coloca o cursor na janela do topo

ctrl-w ctrl-w: Coloca o cursor na próxima janela

ctrl-w_: Maximiza a janela corrente

ctrl-w=: Coloca todas as janelas com o mesmo tamanho

10 ctrl-w+: Adiciona 10 linhas de tamanho na janela corrente

:vsplit arquivo: Divide a janela verticalmente

:sview arquivo: O mesmo que split, mas em modo somente-leitura

:hide: Fecha a janela corrente

:b 2: Abre #2 na janela corrente

Auto-completion do texto

Ctrl+n Ctrl+p (em modo de inserção): Completa palavra

Ctrl+x Ctrl+l: Completa linha

:set dictionary=dict: Define dict como o dicionário atual

Ctrl+x Ctrl+k: Completa usando o dicionário

Dispositivos e sistemas de arquivos

*“Triste época! É mais fácil desintegrar um átomo do que um preconceito.”
Albert Einstein*

Dispositivos, Sistemas de Arquivos e o FileSystem Hierarchy Standard

O sistema de arquivos do Linux permite que o usuário crie e mantenha arquivos em diferentes partições, dispositivos e computadores remotos. O bom gerenciamento do sistema de arquivos está entre as atividades mais importantes para manter um sistema linux estável. Como já é de conhecimento geral o Linux permite sua instalação e execução em Discos IDE, SCSI, CD-ROM, CD-RW, Pen-Drivers, Zip-Drivers, Jaz-Drivers, Memória Flash, disquete entre outros, já deve ter linux rodando em torradeiras.

Antes de tudo vamos conhecer os padrões IDE e SCSI. Os discos IDE (Integrated Device Electronics) são os mais utilizados pelos usuários caseiros e oferecem uma boa performance por um custo baixo. As placas mãe comuns possuem duas interfaces IDE chamadas de Primária e Secundária. E cada conexão permite à instalação de dois discos. Dependendo de qual interface o disco está conectado ele vai ser configurado como mestre primário, escravo primário, mestre secundário e escravo secundário. Os discos SCSI (Small Computer System Interface) oferecem desempenho melhor que os discos IDE. Geralmente são mais caros e utilizados em servidores. Comparado ao IDE, SCSI oferece um excelente desempenho, menor utilização da CPU e um esquema de conexão mais flexível, capaz de lidar com 15 discos em um mesmo bus.

Por padrão o Linux define os arquivos de dispositivos IDE da seguinte forma:

Nome Lógico ou dispositivo	Dispositivo Físico (Discos)
/dev/hda	Discos IDE conectado na interface primária mestre
/dev/hdb	Disco IDE conectado na interface escravo
/dev/hdc	Disco IDE conectado na interface secundária
/dev/hdd	Disco IDE conectado na interface secundária escravo
/dev/sda	Disco SCSI conectado no primeiro canal
/dev/sdb	Disco SCSI conectado no segundo canal
/dev/sdc	Disco SCSI conectado no terceiro canal

Partições Primárias

Cada disco pode conter no máximo quatro partições primárias(4) e doze(12) partições lógicas. Totalizando quinze (15) partições possíveis. Você deve estar pensando: $12 + 4 = 16$, porém uma (1) partição é chamada de "Container" que serve para abrigar as partições lógicas e não é utilizável para gravar dados.

Caso você crie as quatro partições primárias no disco IDE, você terá:

```
/dev/hda1
/dev/hda2
/dev/hda3
/dev/hda4
```

Agora digamos que você queira criar a partição /dev/hda2 como estendida para criar outras partições, então ficaria assim:

/dev/hda1	- Primária
/dev/hda2	- Estendida (Container - Informação das partições lógicas)
/dev/hda5	- Lógica
/dev/hda6	- Lógica

Observe no esquema acima que o sistema pulou de /hda2 para /hda5, isto por que as partições /dev/hd3 e /dev/hd4 são exclusiva para partições primárias.

Partições Estendidas

Uma partição estendida é uma variante da partição primária, mas não é capaz de conter um sistema de arquivos. Em vez disso, ela contém partições lógicas. Apenas uma partição estendida pode existir em um mesmo disco físico.

Partições Lógicas

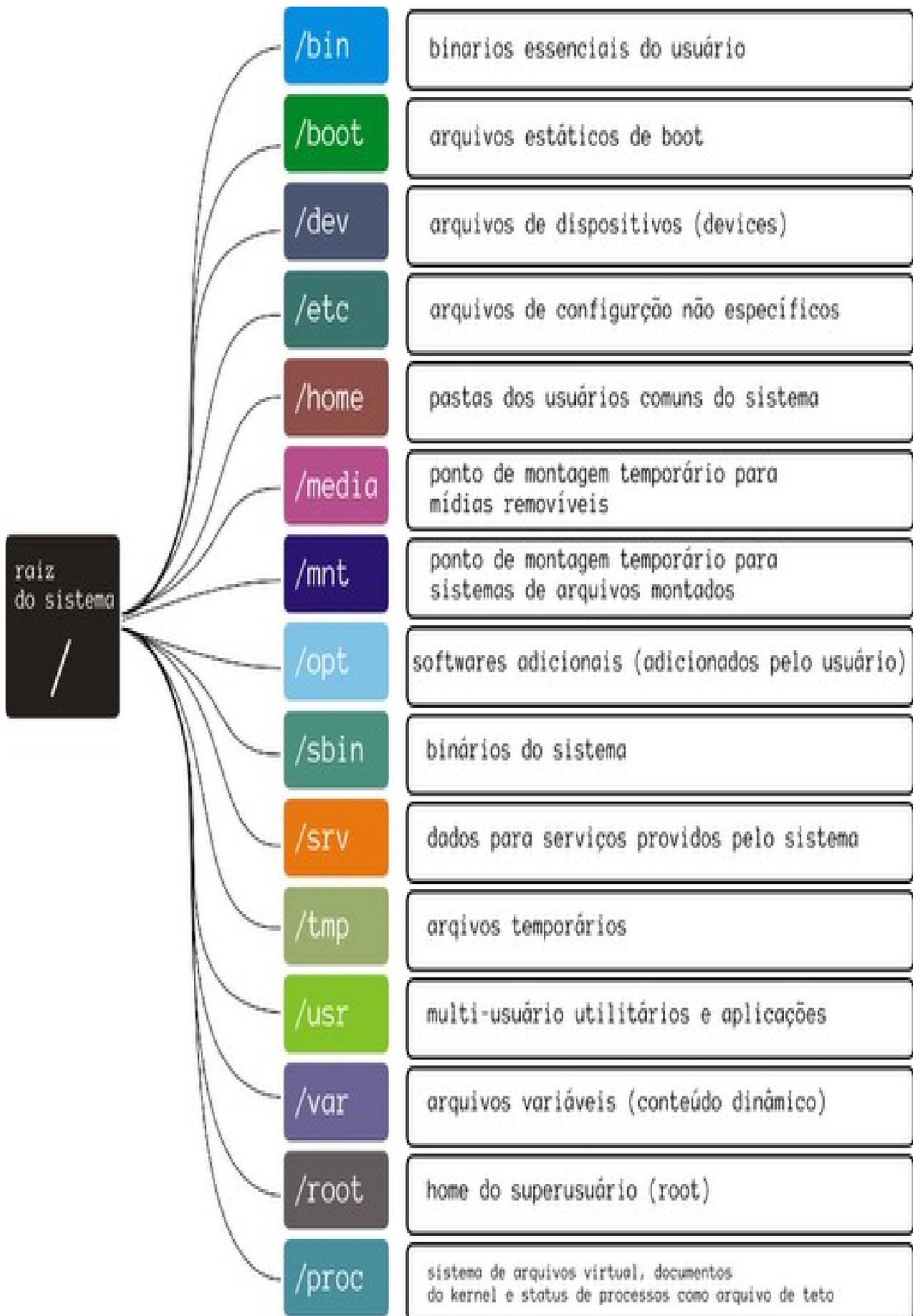
As partições lógicas existem dentro da partição estendida. As partições lógicas são numeradas de 5 a 16. As partições em um disco com uma partição primária, uma estendida e quatro lógicas seriam numeradas da seguinte forma:

/dev/hda1	- Primária
/dev/hda2	- Estendida (Container - Informação das partições lógicas)
/dev/hda5	- Lógica
/dev/hda6	- Lógica
/dev/hda7	- Lógica
/dev/hda8	- Lógica

FHS - File Hierarchy System

FHS - em português Hierarquia de Sistemas de Arquivos é um projeto para que desenvolvedores Unix e desenvolvedores de pacotes de implementações do sistema tivessem um padrão de diretórios. A versão atual é a 2.3 que teve seu anúncio em 29 de janeiro de 2004.

Na próxima página segue uma imagem que define toda esta estrutura.



Quando se faz o boot de um sistema linux, o primeiro sistema de arquivos que se torna disponível é chamado de root (raíz) identificado com uma barra ("/"). O sistema de arquivos root "/" não deve ser confundido com o usuário administrador do linux (chamado de root) ou com o diretório deste usuário que está em /root. Você pode instalar todo o sistema Linux definindo uma partição como "/" na instalação, porém e principalmente em servidores isto não é conveniente. Caso o sistema root ficasse com a sua capacidade de armazenamento lotada o sistema poderia ter falhas. Em vez disto, definimos diversas partições, cada uma contendo um sub-diretório do sistema que está em / (root).

Segue uma sugestão para partições individuais, note que não existe um padrão para este tipo de configuração sendo uma opção pessoal.

/ (root)

Uma vez que o único sistema de arquivos montado no inicio do processo de boot é /, certos diretórios precisam fazer parte dele para estarem disponíveis para o processo de boot. Entre eles, incluem-se:

/bin e /sbin

Contêm programas binários de sistema requeridos

/dev

Contêm arquivos de dispositivos

/etc

Contêm informação de configurações usadas no boot

/lib

Contêm as bibliotecas compartilhadas

Estes diretórios sempre fazem parte da partição "/".

/boot

Este diretório armazena arquivos estáticos usados pelo carregador de boot, incluindo imagens de kernel. Em sistemas onde a atividade de desenvolvimento do kernel ocorre regularmente, tornar /boot uma partição separada elimina a possibilidade de / ficar lotado com imagens de kernel e arquivos associados durante o seu desenvolvimento.

/home

Os arquivos dos usuários são geralmente colocados em uma partição separada. Essa é frequentemente a maior partição do sistema, e poderá se localizar em um disco físico ou array de disco separado.

/tmp

Este diretório frequentemente é uma partição separada usada para impedir que arquivos temporários lotem o sistema de arquivos root.

/var

Os arquivos e log são armazenados aqui. Este caso é semelhante ao /tmp, onde os arquivos do usuário podem preencher qualquer espaço disponível no caso de algo dar errado, ou se os arquivos não utilizados não forem removidos periodicamente.

/usr

Este diretório armazena uma hierarquia de diretórios contendo comandos do usuário, códigos-fonte e documentação. Ele frequentemente, é bastante grande, o que torna um bom candidato a ter a sua própria partição.

Gerenciando partições

Criando partições com o fdisk do linux

O fdisk é uma ferramenta ótima para partições ele consegue criar mais de 60 tipos de partições fora as do linux (native linux e linux swap), ele só não permite mudar o espaço de uma partição como no fips ou Partition magic.

Nesse tutorial vão ser usados exemplos somente para as partições do linux.

Vamos supor que eu tenho um HD de 3.2G, e já tenhamos uma partição do windows (fat 32) com 1.2G.

Se você der somente fdisk, ele vai por default na /dev/hda1 (primary master), se caso você tenha outro HD no /dev/hdb por exemplo, e quiser particioná-lo, faça:

```
fdisk /dev/hd?
```

Onde ? É o número da sua partição.

OBS: o fdisk só pode ser usado pelo root.

Rodando o comando:

```
disk
Using /dev/hda as default device!
Ex.: fdisk /dev/hda (para o primeiro disco IDE);
ou : fdisk /dev/sdc (para o terceiro disco SCSI );
ou : fdisk /dev/eda (para a primeira unidade ESDI PS/2);
ou : fdisk /dev/rd/c0d0 ou : fdisk /dev/ida/c0d0 (para dispositivos RAID)
Command (m for help):
```

De o comando '**m**' e teremos a seguinte tabela:

Command (m for help): **m**

Command action	Comando - ação
a toggle a bootable flag	a alterna a opção "inicializável"
b edit bsd disklabel	b edita rótulo BSD no disco
c toggle the dos compatibility flag	c alterna a opção "compatibilidade"
d delete a partition	d exclui uma partição
l list known partition types	l lista os tipos de partição conhecidos
m print this menu	m mostra este menu
n add a new partition	n cria uma nova partição
o create a new empty DOS partition table	o cria uma nova tabela de partições DOS vazia
p print the partition table	p mostra a tabela de partições
q quit without saving changes	q sai sem salvar as alterações
s create a new empty Sun disklabel	s cria um novo rótulo de disco Sun vazio
t change a partition's system id	t altera a identificação da partição para o sistema
u change display/entry units	u altera as unidades das entradas mostradas
v verify the partition table	v verifica a tabela de partições
w write table to disk and exit	w grava a tabela no disco e sai
x extra functionality (experts only)	x funcionalidade adicional (somente para usuários avançados)

Command (m for help):

Para visualizar as partições do HD digite 'p'.

Command (m for help): **p**

Disk /dev/hda: 128 heads, 63 sectors, 781 cylinders

Units = cylinders of 8064 * 512 bytes

Device Boot Start End Blocks Id System

/dev/hda1 * 1 305 1229759+ b Win95 FAT3

Device = a partição no caso /dev/hda1
Boot = a partição ativa
Start = o cilindro em que a partição começa
End = o cilindro em que a partição termina
Blocks = sera o espaço que a partição esta ocupando
Id = numero das lista de partições do fdisk 'b'= fat32
System = o tipo da partição

Neste exemplo iremos adicionar uma Native Linux e uma Linux Swap.

Para adicionar uma nova partição digite 'n', veja:

```
Command (m for help): n
Command action
e extended
p primary partition (1-4)
```

Aqui você digite 'p', (Partição primária) veja :

```
Partition number (1-4):
```

Escolha o *numero da partição*:

1 =/dev/hda1, 2 =/dev/hda2, 3 =/dev/hda3, 4=/dev/hda4, se você quiser ter ainda mais partições você terá que fazer uma partição estendida, isso será explicado mais a frente.

Então digite 2, porque 1 já existe, que é a partição do windows, ficará assim:

```
Partition number (1-4): 2
First cylinder (306-781, default 306):
```

Aqui você terá que digitar o cilindro de inicio, pressione [enter], pois você pode ver que o cilindro de inicio no padrão já é o "306" (veja acima quando demos o comando 'p' que o END da fat32 era 305, então o cilindro de inicio dessa será 306, e ficará assim:

```
First cylinder (306-781, default 306):
Using default value 306
Last cylinder or +size or +sizeM or +sizeK (306-781, default 781):
```

Coloque o espaço da partição por diferentes tipos:

```
+size = valor em bytes
+sizeM = valor em MegaBytes
+sizeK = valor em kbytes
```

Ou por cilindros, vamos adicionar por MegaByte, digite +1780M que ficará:

```
First cylinder (306-781, default 306):  
Using default value 306  
Last cylinder or +size or +sizeM or +sizeK (306-781, default 781): +1780M  
Command (m for help):
```

Agora nós precisamos especificar de qual tipo é a partição, então digite 't', que ficará :

```
Command (m for help): t  
Partition number (1-4):  
  
Coloque o número da partição, no caso 2:  
Command (m for help): t  
Partition number (1-4): 2  
Hex code (type L to list codes):
```

Digite 'l', aparecerá os tipos de partições suportadas pelo fdisk:

```
Hex code (type L to list codes): l  
  
0 Empty 16 Hidden FAT16 61 SpeedStor a6 OpenBSD  
1 FAT12 17 Hidden HPFS/NTFS 63 GNU HURD or Sys a7 NeXTSTEP  
2 XENIX root 18 AST Windows swa 64 Novell Netware b7 BSDI fs  
3 XENIX usr 24 NEC DOS 65 Novell Netware b8 BSDI swap  
4 FAT16 <32M 3c PartitionMagic 70 DiskSecure Mult c1 DRDOS/sec (FAT-  
5 Extended 40 Venix 80286 75 PC/IX c4 DRDOS/sec (FAT-  
6 FAT16 41 PPC PReP Boot 80 Old Minix c6 DRDOS/sec (FAT-  
7 HPFS/NTFS 42 SFS 81 Minix / old Lin c7 Syrix  
8 AIX 4d QNX4.x 82 Linux swap db CP/M / CTOS / .  
9 AIX bootable 4e QNX4.x 2nd part 83 Linux e1 DOS access  
a OS/2 Boot Manag 4f QNX4.x 3rd part 84 OS/2 hidden C: e3 DOS R/O  
b Win95 FAT32 50 OnTrack DM 85 Linux extended e4 SpeedStor  
c Win95 FAT32 (LB 51 OnTrack DM6 Aux 86 NTFS volume set eb BeOS fs  
e Win95 FAT16 (LB 52 CP/M 87 NTFS volume set f1 SpeedStor  
f Win95 Ext'd (LB 53 OnTrack DM6 Aux 93 Amoeba f4 SpeedStor  
10 OPUS 54 OnTrackDM6 94 Amoeba BBT f2 DOS secondary  
11 Hidden FAT12 55 EZ-Drive a0 IBM Thinkpad hi fe LANstep  
12 Compaq diagnost 56 Golden Bow a5 BSD/386 ff BBT  
14 Hidden FAT16 <3 5c Priam Edisk
```

Digite '83' ("83 Linux"):

```
Hex code (type L to list codes): 83
```

```
Command (m for help):
```

Agora digite 'p' e veja como ficou:

```
Command (m for help): p
```

```
Disk /dev/hda: 128 heads, 63 sectors, 781 cylinders  
Units = cylinders of 8064 * 512 bytes
```

```
Device Boot Start End Blocks Id System  
/dev/hda1 * 1 305 1229759+ b Win95 FAT32  
/dev/hda2 306 758 1826496 83 Linux
```

```
Command (m for help):
```

Agora iremos fazer a swap com o restante do espaço que sobrou:

Digite 'n' para fazer a partição:

```
Command (m for help): n  
Command action  
e extended  
p primary partition (1-4)
```

Digite 'p' e depois ponha o número da partição , no caso 3 (/dev/hda3):

```
Command action  
e extended  
p primary partition (1-4)  
p  
Partition number (1-4): 3  
First cylinder (759-781, default 759):
```

Digite [enter], pois o default já ta em 759:

```
First cylinder (759-781, default 759):  
Using default value 759  
Last cylinder or +size or +sizeM or +sizeK (759-781, default 781):
```

Como vamos usar o resto do espaço no HD para fazer a swap, nem digite +sizeM, pois você pode ver que o último cilindro do HD 781 já está no default, se você digitar [enter] o fdisk irá fazer a swap com o resto de espaço que sobrou no HD (o que no nosso caso a swap ficará com uns 120mb +-).

Agora digite 't' para especificar o tipo da partição, ponha o numero da partição (no caso 3) e depois 'l' para visualizar os tipos:

```
Command (m for help): t
Partition number (1-4): 3

Hex code (type L to list codes): l

0 Empty                16 Hidden FAT16       61 SpeedStor         a6 OpenBSD
1 FAT12                17 Hidden HPFS/NTFS   63 GNU HURD or Sys   a7 NeXTSTEP
2 XENIX root           18 AST Windows swa    64 Novell Netware    b7 BSDI fs
3 XENIX usr            24 NEC DOS             65 Novell Netware    b8 BSDI swap
4 FAT16 <32M          3c PartitionMagic     70 DiskSecure Mult   c1 DRDOS/sec
(FAT-
5 Extended             40 Venix 80286        75 PC/IX             c4 DRDOS/sec (FAT-
6 FAT16                41 PPC PReP Boot      80 Old Minix          c6 DRDOS/sec (FAT-
7 HPFS/NTFS           42 SFS                81 Minix / old Lin    c7 Syrix
8 AIX                  4d QNX4.x             82 Linux swap         db CP/M / CTOS / .
9 AIX bootable         4e QNX4.x 2nd part    83 Linux              e1 DOS access
a OS/2 Boot Manag     4f QNX4.x 3rd part    84 OS/2 hidden C:     e3 DOS R/O
b Win95 FAT32         50 OnTrack DM         85 Linux extended     e4 SpeedStor
c Win95 FAT32 (LB     51 OnTrack DM6 Aux    86 NTFS volume set   eb BeOS fs

e Win95 FAT16 (LB     52 CP/M               87 NTFS volume set   f1 SpeedStor
f Win95 Ext'd (LB     53 OnTrack DM6 Aux    93 Amoeba             f4 SpeedStor
10 OPUS                54 OnTrackDM6         94 Amoeba BBT         f2 DOS
secondary
11 Hidden FAT12        55 EZ-Drive           a0 IBM Thinkpad hi    fe LANstep
12 Compaq diagnost    56 Golden Bow         a5 BSD/386            ff BBT
14 Hidden FAT16 <3    5c Priam Edisk
```

Digite '82' (82 Linux swap), e depois 'p' para visualizar como ficou:

```
Hex code (type L to list codes): 82
```

Changed system type of partition 3 to 82 (Linux swap)

```
Command (m for help): p
```

```
Disk /dev/hda: 128 heads, 63 sectors, 781 cylinders  
Units = cylinders of 8064 * 512 bytes
```

```
Device Boot Start End Blocks Id System  
/dev/hda1 * 1 305 1229759+ b Win95 FAT32  
/dev/hda2 306 758 1826496 83 Linux  
/dev/hda3 759 781 92736 82 Linux swap
```

```
Command (m for help):
```

Pronto suas partições já estão feitas.

Se você quiser mudar a partição ativa digite 'a' depois o numero da partição, como a do windows que esta ativa primeiro você vai ter que digitar 'a' e depois 1, assim ela será desativada, depois digite 'a' e 2, para escolher a Native Linux, digite 'p' para ver como ficou:

```
Command (m for help): p
```

```
Disk /dev/hda: 128 heads, 63 sectors, 781 cylinders  
Units = cylinders of 8064 * 512 bytes
```

```
Device Boot Start End Blocks Id System  
/dev/hda1 1 305 1229759+ b Win95 FAT32  
/dev/hda2 * 306 758 1826496 83 Linux  
/dev/hda3 759 781 92736 82 Linux swap
```

Para fazer mais que quatro partições com o fdisk você vai ter que fazer partição 4 (/dev/hda4) extensível e dela fazer as outras, o procedimento é quase igual suponhamos que você tenha mais espaço no HD e já tenha 3 partições então, digite 'n' depois 'e', veja:

```
Command (m for help): n
```

```
Command action
```

```
e extended
```

```
p primary partition (1-4)
```

```
e
```

```
Partition number (1-4): 4
```

```
First cylinder (781-915, default 915):
```

A partir daí você já conhece, ponha o cilindro inicial e no tipo escolha 85 (85 Linux extended) consulte a lista. Para sair e salvar as alterações digite 'w', e para sair e não salvar nada digite 'q'.

Sistemas de arquivos

Ext2

O ext2 é uma evolução das limitações impostas pelo ext. O ext2 (Second Extended file System) é um sistema de arquivos para dispositivos de blocos (Discos Rígido, disquete, pend-drive).

Ext3

O ext3 (third extended file system) é um sistema de arquivos que acrescenta o recurso de journaling ao ext2.

O journaling consiste em um registro (log ou journal) de transações cuja finalidade é recuperar o sistema em caso de desligamento não programado, falha no disco ou queda de energia.

Ext4

O fourth extend filesystem é um sistema de arquivos também com journaling que a partir do kernel 2.6.28 ficou disponível de forma estável no sistema. Ele trabalha com volumes de até 1 exbibyte e é totalmente compatível com ext2 e ext3. Com performance superior.

Reiserfs v3

O reiserfs é um dos sistemas mais utilizados em sistemas linux, ele possui suporte de até 1 EiB de tamanho e o acesso a árvore de diretórios é um pouco mais rápido que o ext3. O reiserfs também possui Journaling. Uma desvantagem é que seu consumo de CPU é elevado.

VFAT

O vfat (virtual file allocation table) é uma extensão para os sistemas de arquivos fat16 e fat32 incluído a partir do Windows 95 com suporte no linux. Este sistema é amplamente utilizado em cartões de memória e pen-drivers. Não se deve usá-lo em sistemas de arquivos linux, pois possui muitas limitações.

Formatando a partição

O comando mkfs formata a partição criada pelo fdisk com o sistema de arquivos. Você define o tipo de arquivos usando a opção -t e são os formatos nativos ext2, ext3 ou msdos. Os comandos mke2fs e mkdosfs são variações do mkfs.

As opções possíveis do comando mkfs são:

-c

Verifica a existência de bad blocks (defeitos) no disco

-L

Configura o nome do dispositivos

-n

Configura o nome do dispositivo para o formato msdos

-q

Faz com que o mkfs trabalhe com o mínimo de saída no vídeo possível

-v

Faz com que o mkfs trabalhe com o máximo de saída possível no vídeo.

Exemplo:

```
mkfs -t ext2 -L backup /dev/hda4
```

mkswap

O comando mkswap prepara o dispositivo para ser usado como área de memória virtual (swap). A partição da swap tem que ser do tipo 82 (linux swap).

Comando:

```
mkswap /dev/hda6
```

Arquivo de Swap

É possível criar uma swap a partir de um arquivo, este recurso é interessante quando precisamos de um recurso emergencial ao sistema.

A princípio, cria-se um arquivo de **256 MB**. Esse valor pode ser modificado de acordo com suas necessidades e com a observação do desempenho do seu sistema.

```
dd if=/dev/zero of=/meu_arquivo_swap bs=1024 count=262144
```

No comando, *bs* é o tamanho do bloco em *bytes* e *count* é o número de blocos a ser usado. Dessa forma, o tamanho do arquivo é: ****1024*262144 = 256 MB****

Altera-se a permissão de leitura e escrita para o *root* e nada para os demais usuários e grupos:

```
chmod 600 /meu_arquivo_swap
```

Prepara-se o arquivo para ser usado como SWAP:

```
mkswap meu_arquivo_swap
```

Inicia-se o uso do arquivo de SWAP:

```
swapon meu_arquivo_swap
```

Checando o Sistema de Arquivos

O comando **fsck** deve ser executado em partições que apresentarem erros ou em dispositivos que foram desligados incorretamente. A partição deverá estar desmontada ou montada como somente-leitura (**ro**) para a verificação.

Como o comando **mkfs**, o **fsck** possui a opção **-t** para especificar o tipo do sistema de arquivos e um comando específico para cada partição:

fsck.ext2 ou **e2fsck**, **fsck.ext3**, **fsck.xfs**, **reiserfsck** e **dosfsck**.

Examinando e corrigindo o Sistema de Arquivos

debugfs

Depurador interativo de sistemas de arquivos. Examina sistemas **ext2** e **ext3**. Muda diretórios, examina dados de inodes, apaga arquivos, cria links, mostra o log de **journaling ext3**, etc. É usado em casos extremos, geralmente após o **fsck** ter falhado.

dumpe2fs

Mostra informações de grupo de blocos e de *superblocks*.

tune2fs

Configura parâmetros ajustáveis em sistemas de arquivos **ext2**, como rótulo e limites de montagem antes de checar automaticamente.

Uso do Disco

df

Mostra o espaço disponível em cada dispositivo. A análise é feita diretamente no dispositivo. Por padrão, mostra o espaço em unidades de 1kb. A opção **-h** usa medidas apropriadas para tornar a saída mais inteligível.

Exemplo:

```
df -h
```

du

Mostra o espaço ocupado. Sem argumentos, mostra o uso de cada diretório no sistema. Um diretório específico pode ser indicado através da opção **-s**. A opção **-h** usa medidas apropriadas para tornar a saída mais inteligível.

Exemplo:

```
du -s /home/juliano/Downloads/
```

Montando e desmontando sistemas de arquivos**/etc/fstab**

Durante os procedimentos de carregamento do sistema, é o arquivo **/etc/fstab** que determina os pontos de montagem dos dispositivos. Cada linha corresponde a um dispositivo, contendo os seguintes termos separados por tabulações ou espaços:

1. Dispositivo
2. Ponto de montagem (“swap” se for uma partição de *swap*)
3. Tipo de sistema de arquivos
4. Opções
5. Dump (0 ou 1). Determina se dispositivo deverá ser considerado pelo comando *dump*. Se ausente, 0 é considerado.
6. fsck (1 ou 2). Determina a ordem da checagem feita pelo fsck durante o boot. Para a partição raiz (/), deve ser 1. Se ausente, 0 é presumido e a checagem não será feita no boot.

Para permitir que usuários comuns montem e desmontem dispositivos – geralmente, o caso de dispositivos removíveis – deve-se incluir a opção “users” para o respectivo dispositivo.

Exemplo:

```
/dev/cdrom /mnt/cdrom auto noauto,users,ro 0 0
```

A palavra “auto” na posição referente ao sistema de arquivos indica que o sistema de arquivos deve ser identificado automaticamente.

mount

O comando **mount** usado sem argumentos mostra os dispositivos montados e outros detalhes, como ponto de montagem e tipo do sistema de arquivos. Também é usado para montar dispositivos manualmente:

```
mount -t iso9660 -o ro /dev/cdrom /mnt/cdrom
```

O comando do exemplo monta o dispositivo **/dev/cdrom** contendo mídia com sistema de arquivos **iso9660** no diretório **/mnt/cdrom**, com permissão somente-leitura (**ro**).

Para montar manualmente um dispositivo que conste em **/etc/fstab**, basta fornecer para o comando **mount** a localização do dispositivo ou do diretório alvo. Para desmontar um dispositivo, o comando **umount** é utilizado tendo como argumento o dispositivo ou o diretório alvo a ser desmontado. Usado com a opção **-a**, **mount** monta todos os dispositivos em **/etc/fstab** (exceto os marcados com a opção **noauto**).

Opções de montagem

As opções de montagem são as mesmas para **/etc/fstab** e para **mount** diretamente. Se mais de uma opção for fornecida, deverão ser separadas por vírgula.

rw ou **ro**

Gravável ou somente leitura

noauto

Não é montado automaticamente

users

O dispositivo poderá ser montado e desmontado por usuários comuns

user

Apenas quem montou terá permissão de desmontar

owner

As permissões do dispositivo montado se adequarão ao usuário que o montou

usrquota

Habilita o uso de quotas de disco para usuário

grpquota

Habilita o uso de quotas de disco para grupo

remount

Remonta um dispositivo montado com outras opções. Útil para remontar um dispositivo como somente leitura, por exemplo.

Opções mais frequentes:

Nos casos acima, o comando mount irá ler as informações que necessita no arquivo /etc/fstab

-a

Monta todos os dispositivos especificados no arquivo /etc/fstab que não tem a opção noauto selecionada

-r

Monta o sistema de arquivos do dispositivo como somente leitura.

-w

Monta o sistema de arquivos do dispositivo para leitura e gravação.

-o

Especifica as opções de montagem

-t

Especifica o tipo de sistema de arquivos do dispositivo. Exemplo: ext2, ext3, ext4, iso9660, msdos etc.

umount

O comando umount é utilizado para desmontar dispositivos montados pelo comando mount.

-a

Desmonta todos os dispositivos listados no arquivo /etc/mstab que é mantido pelo comando mount como referência de todos os dispositivos montados:

-t

Desmonta somente os dispositivos que contenham o sistema de arquivos especificado no tipo.

Exemplos:

```
umount /dev/cdrom
```

ou

```
umount /cdrom
```

Administrando quotas de Disco

Em sistemas multiusuários, servidores de web, servidores de arquivos, servidores e servidores de e-mail é de suma importância habilitarmos cotas de discos. Com isto, podemos definir a quantidade de espaço em disco para cada usuário ou grupo de usuários.

Limites de cotas

Cada sistema de arquivos possui até cinco tipos de limites de cotas que podem ser aplicados a ele. Esses limites são especificados em blocos de disco de 1.024 cada um.

Limite hard por usuário

O limite hard é a quantidade máxima de espaço que um usuário individual pode ter em um sistema. Assim que o usuário atingir o limite de sua cota, não poderá escrever mais arquivos no disco.

Limite Soft por usuários

Cada usuário é livre para armazenar dados no sistema de arquivos até atingir o seu limite soft. Quando o usuário ultrapassa o limite soft ele recebe um aviso do sistema.

Limite hard por grupo

O limite físico por grupo define o máximo de espaço em disco que um grupo de usuários possa ter. Se este limite for alcançado, os usuários do grupo não poderão mais efetuar gravações no disco.

Limite Soft por grupo

O mesmo conceito do “soft” por usuário porém atuando com mensagens de aviso para todo o grupo de usuários.

Ainda podemos configurar no gerenciamento de cotas um período de graça (grace period) para que um usuário ou grupo que tenha atingido o limite possa apagar alguns arquivos e ficar dentro do estabelecido.

Para trabalhar com cotas é necessário editar o arquivo `/etc/fstab` adicionando o gerenciamento de quota para usuários:

```
/dev/hda2 /home ext3 defaults,usrquota,grpquota 1 2
```

Agora crie o arquivo `quota.user` e o `quota.group` no diretório `/home` e configure as permissões de leitura e escrita somente para o superusuário. (você aprende mais sobre permissão na próximo capítulo deste livro). Estes dois arquivos serão banco de dados para as quotas dos usuários.

```
touch /home/quota.user
touch /home/quota.group
chmod 600 /home/quota.user
chmod 600 /home/quota.group
```

Execute o comando **quotacheck-avug** para iniciar o banco de dados recém criado.

Verifique se o banco de dados foi criado, os arquivos não devem ter tamanho zero.

```
ls -lga /home
```

Habilite o serviço de quotas

```
quotaon -a
```

Para iniciar o serviço de cotas no boot, caso não existe, crie o arquivo **/etc/rc.d/quotas** com o conteúdo:

```
#!/bin/bash
/sbin/quotaon -avug
```

Altere as permissões deste arquivo:

```
chmod 755 /etc/rc.d/quotas
```

Crie o link simbólico para o arquivo de quotas para o runlevel 3 e o runlevel 5.

```
ln -s /etc/rc.d/quotas /etc/rc.d/rc3.d/S20quotas
ln -s /etc/rc.d/quotas /etc/rc.d/rc5.d/S20quotas
```

Configure para uma checagem das cotas uma vez por semana:

```
crontab -e
03**0 /sbin/quotacheck -avug
```

Comandos das cotas

quota

O comando quota mostra as cotas de espaço em disco configuradas para um determinado usuário ou grupo e o seu espaço utilizado.

Suas opções frequentes são:

-u

Mostra a quota definida para um determinado usuários

-g

Mostra a quota definida para um determinado grupo.

-q

Mostra somente quanto a quota for excedida

-v

Mostra as quotas mesmo que nenhuma esteja definida

Exemplo:

```
quota -v grubelilo
```

quotaon

O comando quotaon habilita o gerenciamento de quotas em um dispositivo previamente configurado.

As opções mais frequentes são:

-a

Habilita o gerenciamento de quotas para todos os dispositivos configurados para controle de quotas no arquivo /etc/fstab

-g

Habilita o gerenciamento de quotas para grupos em um determinado dispositivo

-u

Habilita o gerenciamento de quotas para usuários em um determinado dispositivos

-v

Mostra todos os dispositivos onde o gerenciamento de quotas foi habilitado

Exemplo:

```
quotaon -avug
```

Habilita quotas de usuários e grupos para todos os dispositivos configurados em **/etc/fstab**.

quotaoff

Desabilita as cotas de disco em um ou mais sistemas de arquivos

-a

Desativa as cotas para todos os sistemas de arquivos em **/etc/fstab**

-g

Desativa as cotas dos grupos. Esta opção não é necessária quando se usa a opção **-a**, a qual inclui tanto as cotas dos usuários quanto as do grupo

-u

Desativa as cotas dos usuários, está é o padrão

-v

Habilita o modo verbose, para exibir uma mensagem indicando cada sistema de arquivos em que as cotas tenham sido desativadas.

```
quotaoff -av
```

quotacheck

Este comando examina os sistemas de arquivos e compila bancos de dados referentes às cotas.

Opções mais frequentes:

-a

Faz a varredura de todos os dispositivos configurados para o gerenciamento de cotas no arquivo **/etc/fstab** e contrói a base de dados para usuários e grupos.

-g

Constrói a base de dados de cotas para um determinado grupo

-u

Contrói a base de dados de cotas

-v

Mostra todos os procedimentos executados pelo **quotacheck**

Exemplo:

```
quotaoff -a
quotacheck -aug
quotaon
```

edquota

Este comando edita cotas para usuários e grupos. Ele utiliza o editor de texto VI ou outro editor de texto padrão para editar as configurações de cota do disco.

As opções são:

-g

Se a opção -g for especificada, o edquota deverá receber como parâmetro o nome do grupo

-n

Esta opção é padrão, para editar as cotas do usuários

-u

Esta é a opção padrão, para editar as quotas dos usuários

-p

Esta opção é chamada de proto-usuário. Você poderá utilizar esta opção para copiar as configurações de um usuário padrão para um outro usuário.

-t

Esta opção é para configurar o período de graça para os usuários ou determinado grupo. Deve ser combinada com a opção -u ou -g

Exemplo:

```
edquota -u grubelilo
```

```
Quotas for user grubelilo:
```

```
/dev/sda5: blocks in use: 87, limits (soft = 99900, hard = 100000)
```

```
inodes in use: 84, limits (soft=0, hard=0)
```

```
/dev/sda1: blocks in use: 0, limits (soft =0, hard = 0)
```

```
inodes in use: 0, limits (soft = 0, hard = 0)
```

```
-  
-
```

```
"/temp/EDP.auHTZJO" 5 lines, 241 characters
```

Aqui o usuário grubelilo recebeu um limite de 99.9000 blocos (o que, em um sistema padrão ext2 ou ext3 no linux com blocos de 4k, significa 390MB), um limite hard de 100.000 blocos (apenas 400kb a mais do que o limite soft) e nenhum limite sobre o número de arquivos em /dev/sda5. O usuário não tem limites em /dev/sda1

repquota

Comando usado para relatar sobre o status das cotas. Na primeira forma, repquota exibe um relatório resumido sobre as cotas do sistema de arquivos especificado.

Opções frequentemente utilizadas:

-a

Faz relatórios sobre todas as cotas para os sistemas de arquivos de leitura-escrita mencionados em /etc/fstab

-g

Relata as cotas para grupo

-u

Relata as cotas para usuários, esta ação é padrão

-v

Habilita o modo verbose, o qual adiciona um cabeçalho descritivo à saída.

Exemplo:

```
repquota -v /home
```

Permissões de arquivos Linux

Há três níveis de permissão para arquivos e diretórios: **usuário dono do arquivo (u)**, **grupo dono do arquivo (g)** e **outros (o)**.

Exemplo de permissões de arquivos:

```
ls -l /etc/X11/xdm/
total 80
-rwxr-xr-x 1 root root 385 2006-01-14 17:30 GiveConsole
-rwxr-xr-x 1 root root 244 2006-01-14 17:30 TakeConsole
-r--r--r-- 1 root root 3618 2006-01-14 17:30 Xaccess
-rwxr-xr-x 1 root root 191 2006-01-14 17:30 Xreset
-r--r--r-- 1 root root 2559 2006-01-14 17:30 Xresources
-r--r--r-- 1 root root 487 2005-09-12 02:56 Xservers
-rwxr-xr-x 1 root root 4697 2006-01-14 17:33 Xsession
-rwxr-xr-x 1 root root 4545 2005-09-12 03:01 Xsession.orig
-rwxr-xr-x 1 root root 471 2006-01-14 17:33 Xsetup_0
-rwxr-xr-x 1 root root 145 2006-01-14 17:30 Xsetup_0.orig
-rwxr-xr-x 1 root root 189 2006-01-14 17:30 Xstartup
-rwxr-xr-x 1 root root 303 2006-01-14 17:30 Xwilling
lrwxrwxrwx 1 root root 20 2006-03-19 09:36 authdir -> ../../../../var/lib/xdm
-rwxr-xr-x 1 root root 14696 2006-01-14 17:35 chooser
drwxr-xr-x 2 root root 104 2006-01-14 17:30 pixmaps
-r--r--r-- 1 root root 1533 2006-01-14 17:30 xdm-config
```

A primeira letra representa o tipo do arquivo, podendo ser:

- d** → diretório
- l** → link simbólico
- c** → dispositivo especial de caracteres
- p** → canal fifo
- s** → socket
- → arquivo convencional

As demais letras são divididas em grupos de três, determinando as permissões para o dono do arquivo, o grupo do arquivo e demais usuários, respectivamente.

As permissões são alteradas com o comando **chmod**, podendo ser **leitura (r)**, **escrita (w)** e **execução (x)**.

Exemplo: O grupo à que pertence o arquivo texto_simples terá apenas acesso de leitura ao mesmo e para os demais usuário será retirado a permissão de leitura:

```
chmod g=r,o-r texto_simples
```

Inclui permissão de escrita para o grupo do arquivo texto_simples:

```
chmod g+w texto_simples
```

Apesar de possuírem o mesmo modelo de permissões, arquivos e diretórios reagem de maneiras diferentes. Em diretórios, a permissão “**r**” permite acessar o conteúdo do diretório, a permissão “**w**” permite criar arquivos dentro do diretório, e “**x**” permite listar o conteúdo do diretório.

Permissões Octais

Permissões podem ser manejadas mais eficientemente através de um formato numérico, chamado octal. O número octal consiste numa seqüência dígitos, cada um representando as permissões para o usuário, grupo e outros, nessa ordem.

Permissões octais (numéricas) e suas permissões correspondentes:

Dígito	Leitura	Escrita	Execução
0	-	-	-
1	-	-	sim
2	-	sim	-
3	-	sim	sim
4	sim	-	-
5	sim	-	sim
6	sim	sim	-
7	sim	sim	sim

Assim, o comando:

```
chmod 0664 texto_simples
```

Mudará as permissões do arquivo `texto_simples` para `-rw-rw-r--`, ou seja, leitura e escrita para o usuário, leitura e escrita para o grupo e somente leitura para outros.

Para mudar recursivamente todos os arquivos dentro de um diretório especificado, utiliza-se o `chmod` com a opção **-R**.

umask

umask é o filtro de permissões para criação de arquivos. As permissões para novos arquivos criados são aplicadas calculando as permissões padrão do sistema (0666 para arquivos e 0777 para diretórios) menos as permissões `umask`.

`umask` sem argumentos mostra a máscara atual de criação de arquivos. Para mudar, basta fornecer a nova máscara como argumento.

Em sistemas onde os grupos iniciais dos usuários são particulares, a máscara poderá ser 002, o que subtrairá das permissões padrão do sistema a permissão 2 (w - escrita), na categoria outros (o). Dessa forma, os arquivos serão criados com as permissões 0664.

Em sistemas onde o grupo inicial de todos os usuários é o grupo `users`, a máscara poderá ser 0022, o que subtrairá das permissões padrão do sistema a permissão 2 (w - escrita), nas categoria grupo (g) e outros (o). Dessa forma, os arquivos serão criados com as permissões 0644, limitando a permissão de escrita apenas ao usuário dono do arquivo.

suid e sgid

Todos processos são vinculados ao usuário que os iniciou. Dessa forma, um programa pode trabalhar com um arquivo apenas até onde as permissões deste arquivo permitem ao usuário. Algumas tarefas, no entanto, exigem que um usuário altere ou acesse arquivos aos quais não tem a permissão necessária. Por exemplo, alterar a própria senha exige que o arquivo `/etc/passwd` seja alterado e as permissões de `/etc/passwd` só permitem escrita ao usuário dono (root):

```
ls -l /etc/passwd
-rw-r--r-- 1 root root 814 2006-03-15 14:04 /etc/passwd
```

Para contornar essa condição, existe um bit de permissão especial chamado **suid**. Arquivos executáveis que possuam a permissão `suid` serão executados em nome do dono do arquivo, e não em nome de quem os executou.

A permissão `suid` é representada pela letra “**s**”, no campo referente ao dono do arquivo (u):

```
$ ls -l /usr/bin/passwd
-rws--x--x 1 root bin 37880 2004-06-21 16:20 /usr/bin/passwd
```

Exemplo: Incluir o modo `suid` a um arquivo executável:

```
chmod u+s meu_programa
```

De maneira semelhante, a permissão **sgid** atua em diretórios. A permissão `sgid` é uma permissão de grupo, portanto aparece no campo de permissões referente ao grupo.

Num diretório com a permissão `sgid`, todos os arquivos criados pertencerão ao grupo do diretório em questão, o que é especialmente útil em diretórios com o qual trabalham um grupo de usuários pertencentes ao mesmo grupo.

Exemplo: Tornar um diretório `sgid`:

```
chmod g+s dir_do_grupo/
```

Quando habilitadas, as permissões `suid` e `sgid` fazem aparecer a letra “**s**” no lugar da letra “**x**” nas permissões de dono do arquivo e grupo do arquivo, respectivamente. Se a permissão de execução também existir, aparecerá a letra “**s**” em minúsculo. Se apenas as permissões `suid` e `sgid` existirem, aparecerá a letra “**S**” em maiúsculo.

A Permissão **sticky**

O inconveniente em usar diretórios compartilhados é que um usuário poderia apagar algum ou todo o conteúdo inadvertidamente. Para evitar que isso aconteça, existe o modo de permissão **stick**, que inibe usuários de apagarem arquivos que não tenham sido criados por eles mesmos. É o caso do diretório `/tmp`:

```
ls -ld /tmp
drwxrwxrwt 21 root root 1312 2006-03-23 22:51 /tmp
```

A letra “**t**” nas permissões para demais usuários demonstra o uso da permissão `sticky`. Se apenas a permissão `sticky` existir, aparecerá a letra “**T**” maiúscula.

Exemplo: Usar a permissão `sticky` no diretório “comunitario”

```
chmod o+t comunitario/
```

Permissões Especiais em Formato Octal

Como as opções convencionais, as permissões especiais também podem ser manipuladas em formato octal (numérico). A permissão especial é o primeiro dos quatro dígitos da opção no formato octal.

dígito **suid** **sgid** **sticky**

0	-	-	-
1	-	-	sim
2	-	sim	-
3	-	sim	sim
4	sim	-	-
5	sim	-	sim
6	sim	sim	-
7	sim	sim	sim

Atributos **chattr**

É possível estabelecer algumas regras para o tratamento de arquivos em sistemas do tipo ext2, os chamados atributos do arquivo. Para listar esses atributos, usa-se o comando **lsattr**. Para alterar esses atributos, usa-se o comando **chattr**. Cada atributo é especificado por uma letra precedida do sinal +, - ou =, que adiciona, retira ou torna exclusivo o atributo no arquivo/diretório especificado.

Atributo	Descrição
A	Não modifica a data de acesso
a	O arquivo não poderá ter seu conteúdo sobregravado. Apenas o superusuário ou um programa capacitado por CAP_LINUX_IMMUTABLE poderão alterar essa opção.
c	O kernel automaticamente comprime os dados para gravar e descomprime para ler
D	Um diretório com essa função grava imediatamente os dados ao disco, como se a opção de montagem dirsync fosse aplicada a um subconjunto de arquivos
d	Essa opção exclui o arquivo de um backup via dump
i	Não poderá ser alterado, apagado ou renomeado. Nenhum link poderá tê-lo como alvo. Apenas o superusuário ou um programa capacitado por CAP_LINUX_IMMUTABLE poderão alterar essa opção.
j	Todos os dados serão gravados ao jornal ext3 antes de serem propriamente gravados
s	Se o arquivo for apagado, todos seus blocos são zerados no disco
S	Alterações no arquivo são gravadas imediatamente no disco
u	Após a remoção do arquivo, o conteúdo do arquivo é preservado, possibilitando recuperação.

Os atributos “c”, “s” e “u” não são de fato suportados.

Exemplo: Atribuir “i” ao arquivo “texto_simples”

```
chattr +i texto_simples
```

Listas de Controle de Acesso - ACL

Listas de controle de acesso (ACL) são usadas para obter maior controle sobre permissões de acesso à arquivos e diretórios. Uma ACL consiste em um conjunto de entradas, que determinam condições de acesso para um usuário em particular ou um grupo de usuários, permissões de leitura, escrita e execução/leitura.

Uma entrada ACL contém o tipo da entrada, opcionalmente um argumento para o tipo e as permissões.

Tipos de entrada:

ACL_USER_OBJ

Permissões para o dono do arquivo

ACL_USER

Permissões para os usuários especificados pelo argumento

ACL_GROUP_OBJ

Permissões de acesso para o grupo do arquivo

ACL_GROUP

Permissões para os grupos especificados pelo argumento

ACL_MASK

As maiores permissões possíveis para ACL_USER, ACL_GROUP_OBJ e ACL_GROUP

ACL_OTHER

Permissões para processos que não se encaixem em nenhum dos outros itens.

Exemplo de ACL textual:

```
user::rw-  
user:lisa:rw-  
group::r--  
group:toolies:rw-  
mask::r--  
other::r--
```

As permissões ACL tem correspondência com o modelo convencional de permissões. As permissões para o dono do arquivo correspondem à entrada ACL_USER_OBJ. As permissões para o grupo do arquivo correspondem à entrada ACL_GROUP_OBJ, somente se não houver ACL_MASK. Se a ACL contiver uma entrada ACL_MASK, as permissões para o grupo do arquivo corresponderão à entrada ACL_MASK. Permissões para demais usuários corresponde m à ACL_OTHER.

Para alterar dono e grupo de arquivos e diretórios, usa-se **chown** e **chgrp**. O primeiro argumento é um nome válido de usuário ou grupo e o segundo é o arquivo/diretório para alterar. Apenas o superusuário pode usar o comando **chown**, mas qualquer usuário pode usar **chgrp** em seus arquivos/diretórios.
Mudar dono de arquivo:

```
chown luciano texto_simples
```

Mudar grupo de arquivo:

```
chgrp users texto_simples
```

Para alterar usuário e grupo simultaneamente:

```
chown luciano.users texto_simples
```

*Tanto **chown** quanto **chgrp** possuem a opção **-R**, para alterar conteúdos de diretórios recursivamente.*

Links

Hardlinks são um ou mais nomes que um inode do sistema de arquivos pode ter. Todo arquivo criado é, necessariamente, um hardlink para seu inode correspondente. O mesmo só será apagado após seu último hardlink remanescente ser apagado e nenhum programa estar mantendo o arquivo aberto. Novos hardlinks são criados usando o comando **ln**:

```
ln texto_simples outro_texto_simples
```

A opção **-li** do **ls** mostra o número inode dos arquivos:

```
ls -li texto_simples outro_texto_simples
5412 outro_texto_simples 55412 texto_simples
```

Ambos “**texto_simples**” e “**outro_texto_simples**” são hardlinks para o mesmo inode 55412. Hardlinks para o mesmo inode possuem mesma permissão, donos tamanho e data, pois essas permissões são registradas diretamente nos inodes.

```
ls -li texto_simples
-rw-r--r-- 2 luciano users 29 2006-01-01 00:01 texto_simples
```

O número “2” na segunda coluna de informações demonstra que há 2 hardlinks para o inode correspondente à **texto_simples**. Hardlinks só podem ser criados dentro de um mesmo sistema de arquivos. Não é possível criar hardlinks para diretórios. Os arquivos especiais “.” e “..” são hardlinks para diretório criados exclusivamente pelo próprio sistema.

Softlinks (Links Simbólicos)

Links simbólicos apontam para um caminho, ou mais especificamente, para um hardlink. Links simbólicos podem apontar para qualquer alvo, inclusive em outros e diferentes sistemas de arquivos.

Para criar um link simbólico, usa-se ln com a opção **-s**.

```
ln -s texto_simples link_texto_simples
```

Detalhes do link:

```
ls -l mais_texto_simples  
lrwxrwxrwx 1 luciano users 13 2006-03-24 05:11 mais_texto_simples ->  
texto_simples
```

Um link é indicado pela letra “l” no início das permissões, que neste caso são sempre rwxrwxrwx. O tamanho do arquivo de link é exatamente a quantidade de bytes (caracteres) do caminho alvo. A seta ao lado do nome do link simbólico indica o caminho alvo.

Um link simbólico para um caminho relativo será quebrado se o alvo ou o próprio link for movido. Um link simbólico para um caminho absoluto só será quebrado se o alvo for movido ou apagado.

Para atualizar a informação de alvo de um link simbólico existente, recria-se o link com a opção **-f**.

Funções comuns para links simbólicos são indicar caminhos longos frequentemente usados, criar nomes mais simples para executáveis e nomes adicionais para bibliotecas de sistema.

Localizando arquivos no Linux

which

O comando wich recebe como argumento o nome de um comando e trás como resultado a localização no disco deste comando.

Exemplo:

```
which mkdir
```

find

O comando find localiza a partir de um caminho ou diretório recursivamente uma expressão diretamente no sistema de arquivos.

Para localizar arquivos pelo nome:

```
find / -name bash
```

Para localizar pelo nome desconsiderando maiúsculas e minúsculas, utiliza-se a opção `-iname`

```
find / -iname bash
```

Para os links simbólicos de um arquivo, utilizamos a expressão `-lname`

```
find / -lname bash
```

Para localizar arquivos que foram alterados nos últimos 30 dias, utilizamos:

```
find / -ctime 30
```

Para localizarmos arquivos com pelo menos 30k utilizamos a expressão `-size`

```
find / -size 30k
```

Nos comandos acima estamos buscando com `find` no diretório raiz `/` você pode alterar `/` por outro diretório qualquer.

locate

O comando `locate` busca a localização de arquivos e diretórios em um banco de dados criado com o comando `updatedb`, sua busca é mais rápida que o `find` porém é necessário sempre executar o comando `updatedb` para atualizar o campo de pesquisa.

```
locate passwd
```

updatedb

O comando `updatedb` atualiza a base de dados com o nome dos arquivos e diretórios. A base de dados fica em `/var/lib/slocate/slocate`

whatis

Procura no banco de dados `whatis` por uma palavra exatamente

```
whatis bash
```

apropos

Procura no banco de dados do `wathis` por parte de palavras.

```
apropos tou
ktouch (1) - um tutoria de digitação para o KDE
FcUcs4ToUtf8 (3) - convert UCS4 to UTF-8
FcUtf16ToUcs4 (3) - convert UTF-16 to UCS4
FcUtf8ToUcs4 (3) - convert UTF-8 to UCS4
```

Quer contribuir para o desenvolvimento?

Obrigado pela leitura, espero sinceramente que tenha gostado e que de algum modo este livro ajude-o no seu desenvolvimento pessoal ou profissional.

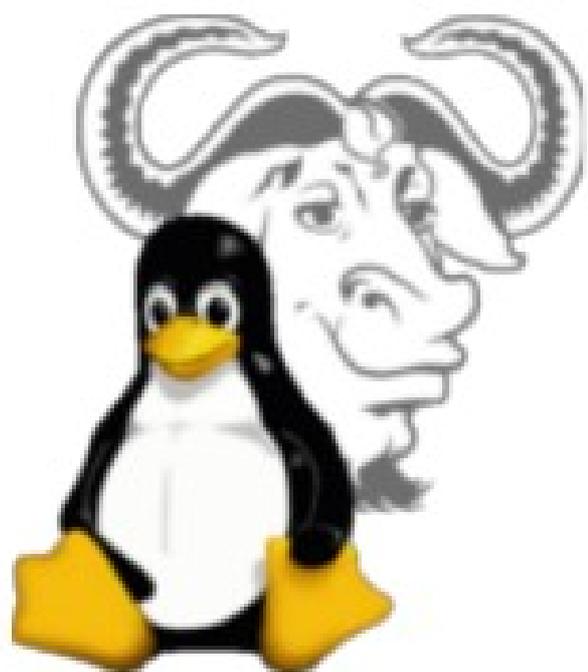
Estou desenvolvendo neste momento o Segundo Livro que vai abordar o exame 102 além de corrigir e aperfeiçoar este que coloquei como versão 0.0.1. Se você quiser ajudar no desenvolvimento de conteúdo, ou enviar correções serei muito grato.

Mas se você não tem tempo, ainda poderá ajudar este projeto fazendo uma doação de qualquer valor no endereço:

<http://lcenter.com.br/linux/arquivos/1437>

Você deve ser cadastrado em nossa rede.

Vamos nos falando - Abraços - Juliano Ramos



Certificação Linux

Este livro é um grande recurso de estudo para a certificação LPIC-1 - 101, CompTIA Linux+ e Novell Linux Administrator.

Porém o livro não é só para quem pretende a certificação, ele aborda o sistema operacional Gnu/Linux de um modo mais amplo para qualquer usuário administrar o sistema.

O linux tem conquistado cada vez mais espaço nas empresas. Até pouco tempo o sistema rodava apenas nos servidores, hoje porém, já é amplamente utilizado nas estações de trabalho.

A tecnologia muda em média a cada seis meses e nem sempre um profissional recém formado na faculdade é capaz de atender as demandas do mercado; para preencher esta lacuna, existem as certificações específicas que garantem o bom conhecimento do profissional. Desde 2012 duas das maiores organizações de certificação resolveram unir suas forças criando uma certificação dupla: A compTIA e a LPI.

Sendo assim, o profissional que se certificar na CompTIA Linux+ Powered by LPI, além de receber o certificado da CompTIA, recebe também o certificado LPIC-1 da LPI.

