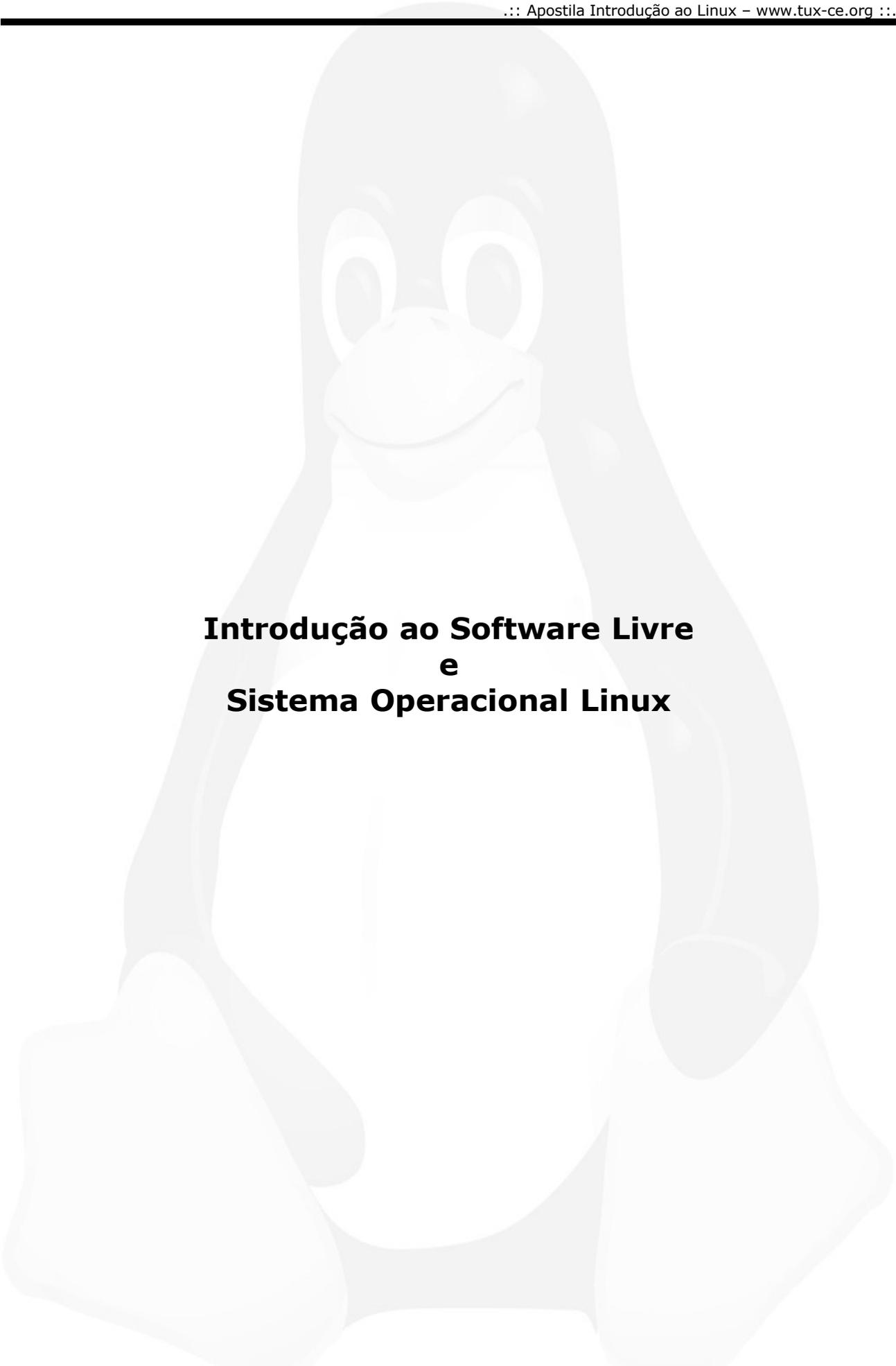


Sumário

Título	02
Nota do Autor.....	03
Parte 1 – Introdução ao Software Livre.....	04
1.1 De onde vem o Software Livre.....	04
1.2 O que é o sistema GNU Linux.....	06
1.3 Porque usar a plataforma Linux.....	07
Principais vantagens da utilização de Software Livre.....	07
1.4 Quem faz o Software Livre.....	08
1.5 Distribuições Linux.....	09
Debian.....	09
Conectiva.....	09
Mandrake.....	10
Mandriva.....	10
Slackware.....	10
Ubuntu.....	11
Suse.....	12
Red Hat.....	12
Kurumin.....	12
Parte 2 – Dispositivos e Sistemas Operacionais.....	13
2.1 Dispositivos de Entrada e Saída.....	14
2.2 Sistema Operacional.....	14
2.3 Estrutura do Sistema de Arquivos.....	16
2.4 Sistema de Arquivos.....	17
2.5 Dispositivos.....	17
Parte 3 – Introdução ao Linux.....	19
3.1 Entendendo uma Shell.....	19
3.2 Efetuando Login e Logout.....	19
3.3 Desligando e Reiniciando a máquina.....	20
3.4 Comandos de Ajuda.....	20
3.5 Comandos de Navegação.....	21
3.6 Comandos para Manipulação de Arquivos.....	22
3.7 Comandos de Terminal.....	24
3.8 Comandos de Localização de Arquivos.....	25
3.9 Comandos de Informações do Sistema.....	26
3.10 Comandos de Exibição e Edição de Arquivos.....	28
Parte 4 – Utilizando o APT.....	29
4.1 O que é o APT.....	29
4.2 Atualizando o Banco de Dados do APT.....	30
4.3 Procurando por Pacotes.....	30
4.4 Instalando Pacotes.....	30
4.5 Removendo Pacotes.....	30



**Introdução ao Software Livre
e
Sistema Operacional Linux**

Copyright (c) 2006-2007 - Comunidade de Software Livre Tux-CE.

Conteúdo licenciado sob a licença GNU FDL - Free Documentation License - sem seções invariantes.

Nota do Autor

É importante ressaltar que todo o material produzido aqui foi baseado em experiência pessoal ou consultas a materiais disponíveis na internet como o site Wikipedia de onde foram tiradas algumas descrições sobre as distribuições Linux.

Marcelo Cavalcante Rocha / Kalib

Para maiores informações:

Site: www.tux-ce.org

Blog: www.tux-ce.org/blog

Forum: www.tux-ce.org/forum

Email: marcelo@tux-ce.org

Parte 1 – Introdução ao Software Livre



Porque nós
amamos a liberdade

Gnu – Mascote do Projeto Gnu

1.1 - De onde vem o conceito de Software Livre ?

Diante de nossa atual situação tecnológica seria impossível falarmos de GNU / Linux sem antes entendermos o que é Software Livre ou de onde surgiu essa filosofia.

Durante muito tempo a tecnologia vem se desenvolvendo e tornando-se cada vez mais necessária para os dias atuais. Para tal evolução, pessoas do mundo inteiro, chamadas programadores, contribuíram de forma significativa criando programas ou softwares para customizar o funcionamento dos computadores. Em boa parte desta evolução, ao precisar de algo que outro já criou, o programador receberia prontamente deste outro para assim poder fazer suas devidas mudanças ou mesmo aperfeiçoamentos poupando-lhe esforço e tempo. O autor original como agradecimento recebia seu programa melhorado. Porque refazer o que já foi feito antes?

Desta forma se dava a troca de informações e conhecimento entre a comunidade como um todo até que uma outra idéia surgiu: Fechar o código fonte e ganhar dinheiro em cima disso.

Foi nesta época que o conhecimento passou a deixar de ser livre passando a ter um dono e uma patente em cima do mesmo. Sendo vendido em caixas com a permissão para instalação em apenas uma máquina limitando-se ao uso técnico sem poder estudar aquela tecnologia a fundo ou repassar para outros interessados. Aí mostrava-se o limite da tecnologia como uma linha que não se podia ultrapassar, já que você não mais poderia aperfeiçoar um software adaptando-o às suas necessidades por não mais ter acesso ao código fonte do mesmo. O conhecimento passou a ser então manipulado e controlado criando uma enorme dependência tecnológica.

Em meados de 1983-1984 o programador americano Richard Stallman percebeu que o senso de união e integração estava perdendo espaço para um senso de egoísmo e monopólio passando assim a iniciar um movimento ativista criando assim o Projeto GNU em 1985 defendendo a idéia do Software Livre. Por que refazer todo um programa quando você tinha programas parecidos com o que você queria desenvolver? Por que não ajudar em um programa que você achava que estava rodando lento e que possivelmente saberia como deixá-lo mais rápido contribuindo assim para o seu código fonte? Essas perguntas não faziam sentido para o Richard, e por essas e outras ele começou este projeto, que de tão óbvio que era, rapidamente caiu na simpatia de toda a comunidade. A partir daí milhares de pessoas espalhadas pelo mundo inteiro uniram-se a ele nessa luta de forma a mostrar à todos que o compartilhamento da informação é fundamental para o desenvolvimento assim como a união e a solidariedade. Para tal filosofia foram criadas algumas regras que na verdade são liberdades.

Não poderia ser diferente tratando-se de Software LIVRE. São as 4 liberdades do Software Livre listadas abaixo:

- * Liberdade de executar o programa, da forma como quiser.
- * Liberdade de estudar o funcionamento do programa podendo inclusive modificá-lo de acordo com sua necessidade.
- * Liberdade de redistribuir cópias do programa livremente.
- * Liberdade de melhorar o programa e tornar estas melhorias públicas de forma a beneficiar o próximo.

O projeto GNU não é somente desenvolvimento e distribuição de alguns softwares livres úteis. O coração do projeto GNU é uma idéia: que software deve ser *livre*, e que a liberdade do usuário vale a pena ser defendida. Se as pessoas têm liberdade mas não a apreciam conscientemente, não irão mantê-la por muito tempo. Se queremos que a liberdade dure, precisamos chamar a atenção das pessoas para a liberdade que elas têm em programas livres.

Com o passar dos anos, e baseado nessa filosofia do Software Livre, surgiu então o sistema operacional que hoje chamamos de GNU / Linux.

1.2 - O que é o sistema GNU / Linux ?



Tux – Mascote do Gnu Linux

Livre,
Poderoso e
Seguro!

Em 1989 um estudante finlandês chamado *Linus Torvalds* inicia um processo pessoal de aprimoramento do Kernel do *Minix*, um sistema operacional do tipo Unix escrito por *Andrew Tannenbaum*, chamando esta vertente de Linux como abreviação de Linus's Minix.

O que parecia um projeto acadêmico foi tomando novos ares e Linus passou a perceber as possibilidades que aquilo poderia trazer assim como suas possíveis proporções, então depois de um certo tempo de trabalho, Linus lança a seguinte mensagem na internet:

"Você sente saudade dos bons dias do minix-1.1, quando homens eram homens e escreviam seus próprios device drivers? Você está sem um bom projeto e morrendo de vontade de colocar as mãos em um sistema operacional o qual possa modificar de acordo com suas necessidades? Você acha frustrante quando tudo funciona bem no Minix? Sem mais noites em claro para fazer com que um programa funcione? Então esta mensagem pode ser exatamente para você. :-)

Como eu mencionei há um mês, estou trabalhando em uma versão livre de um sistema operacional similar ao minix para computadores AT-386. Ele finalmente alcançou o estágio onde pode ser utilizado (ou não, dependendo do que você deseja), e eu estou disposto a colocar os fontes disponíveis para ampla distribuição. Ele está apenas na versão 0.02, mas eu tenho executado nele, sem problemas, programas como bash, gcc, gnu-make, gnu-sed, compress, etc."

Esta mensagem chama a atenção de muitas pessoas, entre elas programadores, analistas ou até mesmo entusiastas do movimento de Software Livre ou GNU. Logo essas pessoas passaram a entrar em contato com Linus e então um grande time de desenvolvimento formou-se em cima do que hoje chamamos de GNU / Linux.

Com pessoas espalhadas pelo mundo inteiro trabalhando em cima do mesmo objetivo, não fica difícil imaginar porque o GNU / Linux passou a ser tão estável, seguro e maduro como sistema operacional.

Finalmente em 1991 Linus lança a primeira versão oficial do Linux, juntando-se mais tarde, em 1992, ao projeto GNU de Richard Stallman com o objetivo de produzir o sistema operacional completo como o conhecemos hoje.

1.3 - Por que usar a plataforma Livre ?

O Sistema Operacional que roda no desktop da prefeita de Fortaleza é GNU/LINUX



Luiziane – Prefeita de Fortaleza usando Gnu Linux

Imaginemos o seguinte padrão: Softwares são programados através de uma linguagem. O que é Linguagem? É um conjunto de códigos que funciona como Meio de Comunicação, seja ele entre Humano / Humano ou Humano / Máquina por exemplo. Então imaginemos que a língua portuguesa é como um Software, e precisamos dela para nos comunicar com outras pessoas, correto? Agora imagine você se os códigos da língua portuguesa fossem patenteados por alguém ou mesmo fechados. Além de você não ter livre acesso aos vários códigos que formam a língua portuguesa você ainda teria de pagar para quem a patenteou. Sim, você teria que pagar para falar!

O que isto tudo importa para a informática? Bom, infelizmente a maioria dos softwares ou programas de computador vem enfrentando esta mesma situação no Brasil, pois é através do software que eu consigo me comunicar com meu computador, e a maioria das pessoas ainda tem de pagar para falar com seu computador. O mais absurdo é que o pagamento recolhido sequer fica no Brasil, pois é entregue para uma empresa no exterior que monopoliza o mercado. O Brasil paga cerca de R\$ 1.000.000.000,00 (1 bilhão) por ano em troca desses softwares.

Num sistema fechado, também não conseguimos desenvolver nossa própria autonomia tecnológica, pois não temos como estudar o seu código e não temos segurança de acesso e envio de informações em nossas próprias máquinas. Quem garante que naquele software fechado instalado em minha máquina não há junto um programa espião vasculhando minhas contas e arquivos pessoais?

No software livre você não tem obrigatoriedade de pagar nada a ninguém, desenvolvido por milhares de programadores ao redor do mundo, voluntários ou não, que compartilham seus códigos com o mundo, no software livre você usa produtos testados por milhares de pessoas que entendem do assunto e que procuraram de todas as formas possíveis, brechas, falhas, bugs e você mesmo pode ajudar nisso, você como usuário é parte importante da comunidade do software livre, sugerindo, reportando alguma falha, opinando, discutindo, ensinando e aprendendo, ou seja, o software livre vai além do uso da ferramenta, ele vai a democratização do conhecimento.

Principais *vantagens* da utilização de Software Livre:

- * Segurança (praticamente isento de vírus, você ****sabe**** o que está instalando em sua máquina, pois seu código é aberto)
- * Economia (você pode baixar ele sem custo da internet, tanto o software quanto a sua documentação de uso)
- * Alternativa à pirataria (você não corre riscos ao ser surpreendido por fiscais cobrando por licenças)
- * Engajamento (você estará utilizando uma solução mais viável para um país em desenvolvimento como o Brasil)

* Autonomia (você tem liberdade para fazer o que quiser com este software desde que siga as 4 liberdades básicas a ele atribuídas)

** As 4 liberdades foram citadas mais acima no tópico "De onde vem o conceito de Software Livre".

SOFTWARE LIVRE! Socialmente Justo...Economicamente Viável...Tecnologicamente Sustentável.

1.4 - Quem faz o software livre ?



A resposta para essa pergunta é bem simples; Você! Contribuindo com código fonte, engajando em projetos de softwares existente, reportando bugs de programas, repassando o conhecimento adquirido, ajudando outras pessoas por meios diversos, chat, eventos ou mesmo um bate-papo na rua difundindo a cultura do software livre, você também pode engajar em uma comunidade ou diretamente em um PSL (Projeto Software Livre) que tem como fundamento principal integrar as comunidades de software livre, usuários, empresas que apóiam a iniciativa, e pessoas interessadas, unindo essas pessoas para fazer ações coletivas.

Busque informações através da rede e comece a participar de comunidades, grupos de usuários, coletivos, listas de discussão etc. e passe a interagir com quem você tem mais afinidade.

1.5 - Distribuições Linux

Por se tratar de um sistema operacional livre e de código aberto não demorou para que milhares de pessoas começassem a fazer modificações no Linux de forma a customizar ou adequá-lo às suas necessidades pessoais. Cada vez que alguém altera o Linux com personalizações e particularidades, esta alteração é chamada de Distribuição Linux, e como tal cada pessoa que o fizer pode dar um nome para sua distribuição e publicá-la na rede para que outras pessoas possam utilizar a sua distribuição.

Cada distribuição tem sua característica própria, atendendo as necessidades de seus usuários, tais como o sistema de instalação, o objetivo, a localização de programas, nomes de arquivos de configuração, etc. A escolha de uma distribuição é pessoal e depende das necessidades e finalidades de cada um.

Atualmente existem centenas de distribuições Linux, porém, como tudo na vida, algumas se destacaram. Algumas distribuições bastante conhecidas e destacadas nos dias de hoje são: Slackware, Debian, Red Hat, Fedora, Mandriva, Suse, Ubuntu e Kurumin, todas usando o SO Linux como kernel principal.

Debian



Debian é simultaneamente o nome de uma distribuição não comercial livre (gratuita e de código fonte aberto) de GNU/LINUX (amplamente utilizada) e de um grupo de voluntários que o mantêm à volta do mundo. Uma vez que o Debian se baseia fortemente no projecto GNU (e a distribuição oficial do Projeto GNU é Debian), é usualmente chamado Debian GNU/Linux. O Debian é especialmente conhecido pelo seu sistema de gestão de pacotes, chamado APT, que permite: atualizações relativamente fáceis a partir de versões realmente antigas; instalações quase sem esforço de novos pacotes e remoções limpas dos pacotes antigos.

O projecto Debian é mantido por doações através da organização sem fins lucrativos Software in the Public Interest (SPI).

O nome Debian vem dos nomes dos seus fundadores, Ian Murdock e de sua mulher, Debra. A palavra "Debian" é pronunciada em Português como Débian.

Várias distribuições comerciais baseiam-se (ou basearam-se) no Debian, incluindo: Lindows (atual Linspire), Xandros, Kurumin, Debian-BR-CDD e Libranet. O Ubuntu_Linux também se baseia em Debian.

Conectiva



A Conectiva foi a primeira distribuidora brasileira do sistema operacional Linux. Suas distribuições são fáceis de instalar e configurar. A distribuição é direcionada a usuários brasileiros, sendo que o hardware que ela suporta é aquele mais comumente encontrado no Brasil. Essa distribuição está disponível em português, espanhol e inglês. A empresa Conectiva é integrante do "United Linux".

No dia 24 de fevereiro de 2005 a MandrakeSoft, companhia número 1 em Linux na

Europa, anunciou a compra da Conectiva. Adquiriu todas as ações da Conectiva por US\$2,3 milhões. A MandrakeSoft é a empresa que distribui o Mandrakelinux, uma das mais populares distribuições Linux. A empresa passou a se chamar Mandriva.

Mandrake



Mandrakelinux é uma distribuição GNU/Linux de origem francesa baseada originalmente na distribuição Red Hat Linux (ambas usam pacotes RPM) criada em 1998 por Gaël Duval que é também co-fundador da Mandrakesoft; empresa responsável pela distribuição e manutenção dos produtos derivados do Mandrakelinux.

Sua principal característica é a facilidade no uso, pois conta com diversos assistentes gráficos que evitam o uso de comandos de linha e reconhece automaticamente diversos dispositivos de hardware.

Até 2004 a distribuição era oficialmente chamada Mandrake Linux, ou simplesmente Mandrake. A alteração no nome ocorreu devido a um problema jurídico envolvendo a King Features Syndicate dona dos direitos do personagem de quadrinhos Mandrake, o mágico

Recentemente a Mandrakesoft adquiriu a empresa brasileira Conectiva. Esperasse com isso que o Mandrakelinux consiga aumentar sua presença no mercado latino americano.

Mandriva



Mandriva é junção do nome das duas empresas MandrakeSoft e Conectiva. A Mandriva Conectiva é a operação brasileira da Mandriva, desenvolvedora e distribuidora do sistema operacional Mandriva Linux, resultado da fusão ocorrida em 24 de fevereiro de 2005 entre a MandrakeSoft, uma das principais distribuições Linux da Europa, com atuação mundial em mais de 120 países, e a Conectiva, pioneira na distribuição Linux e código aberto em português, espanhol e inglês para toda a América Latina. A Mandriva possui escritórios nos Estados Unidos, França e Brasil, tem mais de 8 milhões de usuários e uma carteira de 170 grandes clientes corporativos, além de contar com 130 funcionários. É uma das distribuições linux mais potentes e relativamente fácil de usar, seja para uso doméstico e/ou corporativo.

Encontra-se para baixar no site oficial, nas interfaces GNOME e KDE. Como qualquer outra distribuição Linux, o Mandriva pode ser usado pelo CD, sem instalar no disco rígido, método chamado de Live CD. Segundo o site oficial seus requisitos são: 256 MB de memória RAM no mínimo, recomendados 512 MB, tanto para live CD quanto instalar no HD e cerca de 3 GB para a instalação deste no HD. Gratuitamente temos o Mandriva ONE 2007 Spring, que gasta apenas 1 CD, é voltado para iniciantes e temos também o Mandriva FREE 2007 Spring, que gasta 1 DVD, voltado para usuários mais experientes.

Slackware



Slackware Linux é o nome de uma das mais antigas e conhecidas distribuições (sistema operacional e conjunto de aplicativos) do Linux; é a mais antiga, distribuição que permanece mantida, além de ser, junto com suas derivadas, a distribuição Linux mais UNIX-like existente.

Criada em meados de 1993 e mantida por Patrick Volkerding, a Slackware (ou simplesmente "Slack") tem como objetivo manter-se fiel aos padrões UNIX, rejeitando também ferramentas de configuração que escondam do usuário o real funcionamento do sistema. Além disso a Slackware é composta somente de aplicativos estáveis (e não de versões beta ou pré-releases). Nos anos 90, por um bom tempo outras distribuições Linux

foram avaliadas com base em sua compatibilidade com o Slackware. Se no começo ainda trabalhava só, ao longo dos anos Patrick Volkerding acabou aceitando a ajuda de alguns colaboradores, muito poucos por sinal, a fim de o ajudarem no desenvolvimento da distribuição.

Por sua concepção UNIX-like, o Slackware e seus derivados fazem uma abordagem bastante diferente das outras distribuições populares como Red Hat, Fedora, Debian, Gentoo, SuSE, e Mandriva. Sua política de incluir somente aplicativos estáveis e nenhuma interface gráfica de configuração específica a apenas essa distribuição. Há quem diga: "Quando você conhece o Slackware, você conhece Linux; quando conhece Red Hat, tudo que você conhece é o Red Hat"...

Simplicidade e estabilidade são duas características marcantes nesta distribuição muito comum em servidores, distribuição que procura ser uma distribuição "leve", sem enfeites e rápida, e que é muito apreciada por usuários mais experientes. Apesar de o termo "slack" ser uma gíria para preguiça em inglês, em "Slackware" o termo "slack" está relacionado à definição feita pela chamada Church of the SubGenius (Igreja dos Sub-Gênios). De acordo com esta organização, "slack" significa ter "senso de liberdade, independência e originalidade para alcançar suas metas pessoais". Dessa forma, o nome "Slackware" traduz bem a filosofia do sistema, que não possui ferramentas gráficas de configuração (apenas as do KDE, por exemplo) que fazem tudo pelo usuário, o que inibe o seu "slack". Assim sendo, as configurações do sistema são feitas a partir da edição de documentos de texto, por isso sendo a preferida entre os usuários mais experientes.

Ubuntu



Ubuntu é um sistema operativo (no Brasil, chamam-lhe sistema operacional) Linux baseado na distribuição Debian. É patrocinado pela Canonical Ltd (dirigida por Mark Shuttleworth) e o seu nome deriva do conceito sul africano Ubuntu (pronuncia-se /ùbùntú/ - u-BÚN-tu; IPA: /ubuntu/) — diretamente traduzido como "humanidade para com os outros". Diferencia-se do Debian por ser lançado semestralmente, por disponibilizar suporte técnico nos dezoito meses seguintes ao lançamento de cada versão (em inglês) e pela filosofia em torno de sua concepção, uso e distribuição.

A proposta do Ubuntu é oferecer um sistema operacional/operativo que qualquer pessoa possa utilizar sem dificuldades, independente de nacionalidade, nível de conhecimento ou limitações físicas. A distribuição deve ser constituída totalmente de software gratuito e livre, além de isenta de qualquer taxa. Actualmente uma organização cuida para que cópias sejam remetidas em CDs para todo o mundo sem custos.

A Comunidade Ubuntu ajuda-se mutuamente, não havendo distinção de novatos ou veteranos; a informação deve ser compartilhada para que se possa ajudar quem quer que seja, independentemente do nível de dificuldade.

Há vários meses a distro Ubuntu está em primeiro lugar no DistroWatch, site especializado em catalogar o desempenho e uso das milhares de distros Linux.

Suse



A SuSE, mais precisamente SUSE LINUX AG, é uma empresa alemã, que disponibiliza soluções baseadas no sistema operativo GNU/Linux, incluindo várias distribuições. Em Janeiro de 2004 a empresa foi adquirida pela Novell, uma empresa americana que na década de 1980 ficou famosa por seu sistema operacional de rede (Netware) e que perdeu fatia substancial de mercado com o advento do Windows NT da Microsoft.

Red Hat



Red Hat Linux é uma distribuição de Linux muito conhecida, líder do mercado nos EUA, criada e mantida pela Red Hat.

Um grupo de programadores na Carolina do Norte decidiu tornar o Linux mais fácil para possibilitar às pessoas uma experiência mais tranquila com o mesmo. Como muitos grupos, seu objetivo era empacotar todos os bits necessários numa distribuição coerente, facilitando aos inexperientes o contato com o novo sistema operacional.

Esta distribuição porém tinha uma característica distinta das demais. Em vez de ser uma cópia de um disco rígido que tivesse o Linux instalado, ou um conjunto de disquetes com partes diferentes do sistema operacional que podiam ser copiadas, esta distribuição foi baseada no conceito de pacotes.

Cada pacote fornece um pedaço diferente de software, configurado, completamente testado e pronto para rodar. Quer experimentar um novo editor? Carregar o pacote e instale-o. Em segundos, é possível estar executando o pacote instalado. Não gosta dele? Emita um único comando e o pacote será removido.

Além disso o conceito de pacotes traz uma vantagem adicional: Esta distribuição do Linux podia ser atualizada facilmente.

O desenvolvimento do Linux no mundo é muito rápido, provocando a geração de novas versões com freqüência. Em outras distribuições, a actualização do software tornou-se complexa, em uma atualização completa, normalmente o usuário teria que deletar tudo em seu disco rígido. Desse modo você provavelmente supõe que o grupo de programadores na Carolina do Norte é a Red Hat Software, e o pacote de distribuição é o Red Hat Linux. Desde a introdução do Red Hat Linux em 1994, o Linux e o Red Hat tiveram um crescimento extraordinário. Muito mudou: suporte para hardwares mais sofisticado, grande aumento de fiabilidade, e o crescente uso do Linux por companhias ao redor do mundo.

Kurumin



O Kurumin Linux é uma distribuição Linux baseada no Knoppix e que mantém o mesmo sistema de detecção de hardware desta distribuição. Todavia, o Kurumin foi projetado para que fosse bem mais compacto, cabendo, assim, em suas versões iniciais, em um mini-CD de

80 mm.

Inicialmente o seu desenvolvedor, Carlos E. Morimoto, deu início ao projeto apenas para fins de uso pessoal; porém, ao anunciar a sua criação no seu site, muita gente demonstrou interesse pelo projeto, o que incentivou Morimoto a levar o projeto adiante.

Segundo o sítio DistroWatch, Kurumin é a distribuição Linux mais popular no Brasil.

O nome kurumin vêm da Língua tupi-guarani, onde "curumim" significa "menino", "criança", em uma alusão a uma distribuição Linux mais leve e simples, para iniciantes no sistema. A letra k no início da palavra é uma referência ao Knoppix. O logotipo do Kurumin é um pingüim com aspecto infantil: pequeno, simpático e mais magro do que o Tux, o pingüim que representa o Linux em si. Outras características notáveis são o cocar na cabeça do mascote e uma bandeira do Brasil em sua barriga.

Tendo sido focado sempre na língua portuguesa, recentemente foi anunciada a formação de um grupo de trabalho para traduzir a versão 6.0 para as línguas inglesa e espanhola.

Parte 2 – Dispositivos e Sistemas Operacionais

2.1 - Dispositivos de Entrada e Saída

- Entrada - Permite a comunicação do usuário com o computador. São dispositivos que enviam dados ao computador para processamento.
Exemplos: *Teclado, mouse, scanner*.
O dispositivo de entrada padrão (stdin) em sistemas GNU/Linux é o teclado.
- Saída - Permite a comunicação do computador com o usuário. São dispositivos que permitem o usuário visualizar o resultado do processamento enviado ao computador.
Exemplos: *Monitor, Impressora*.
O dispositivo de saída padrão (stdout) em sistemas GNU/Linux é o Monitor.

2.2 - Sistema Operacional

O Sistema Operacional (S.O.) é um conjunto de programas responsável por alocar recursos de hardware e escalonar tarefas, controlar os dispositivos de entrada e saída (I/O), tais como vídeo, teclado, mouse, e ainda controlar os recursos internos que compõem o computador, tais como: processador, memória, arquivos, etc.

Em termos de hardware um S.O. é a porção de software que roda em modo núcleo (kernel), e tem a importante função de proteger o hardware da ação direta do usuário, isto é, estabelecer critérios de uso dos recursos, ordem de acesso aos mesmos, impedindo violação de espaço de memória de processos concorrentes e tentativas de acesso simultâneo a um mesmo recurso, ou seja, gerência e proteção dos dispositivos.

Os S.O.(s) podem ser divididos em:

- Sistemas - Operacionais Monoprogramáveis / Monotarefa
Ex: Windows 95/98.
- Sistemas Multiprogramáveis / Multitarefa
Ex: Unix, Linux, Windows NT/2000/2003.

O S.O. Linux possui várias características que diferenciam dos outros sistemas operacionais e que o aproximam do Unix, sendo um dos motivos da sua escolha em várias aplicações nas quais são necessárias estabilidades e segurança.

Antes de continuarmos, convém estabelecermos alguns termos utilizados.

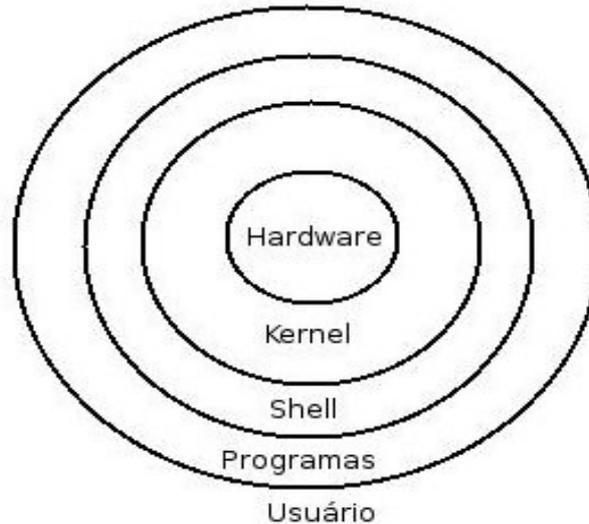
* *Kernel*

É o núcleo do sistema operacional, a parte mais próxima do nível físico (hardware). Composta de chamadas ao sistema, de acesso aos dispositivos de entrada e saída e de gerência dos recursos da máquina.

* *Shell*

Shell é o nome genérico de uma classe de programas que funcionam como interpretador de comandos e linguagem de programação script (interpretada) no Unix. Os Shell mais populares são bash, chs, tsh, zsh.
O Shell é a interface entre o usuário e o Kernel. O usuário pode escolher qual dos shells vai utilizar. O Shell padrão do Linux é o bash.

Sistema Operacional



Visão macro de um computador

* Script

Script é um arquivo que contém comandos de Shell, que, numa situação normal, poderiam ser executados em modo prompt. Esses comandos são executados sequencialmente, dependendo de situações, dependendo de condições estruturais de laço, como if ...then(se ...então) e do....while(façaenquanto). Um script também pode usar variáveis. A entrada pode ser recebida por meio do prompt de comando ou de outros arquivos.

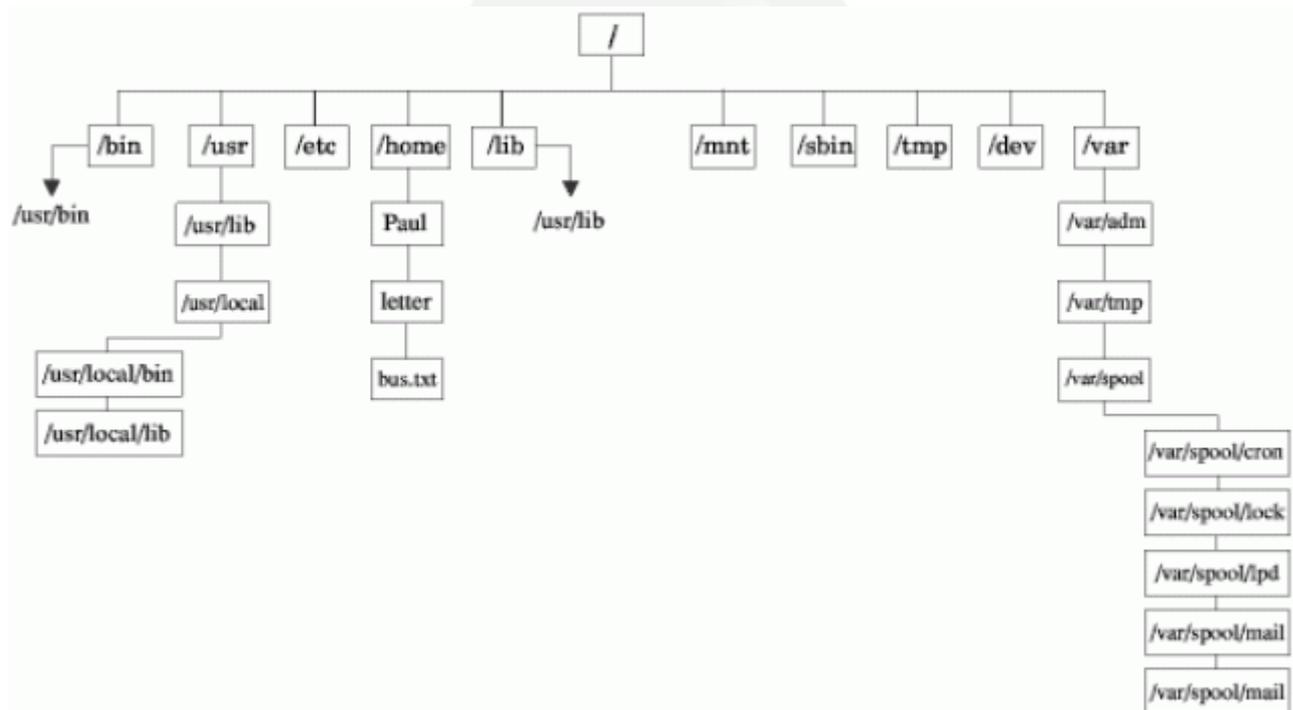
Hardware: São componentes referente a parte física do computador (impressora, disquete, placa de tv, placa mãe, pendrive, discos rígidos, etc).

Software: São os programas instalados no computador (sistema operacional, editor de texto, planilha, banco de dados, interfaces, scripts, interfaces de comando, etc).

No dito popular existe uma maneira simples de se diferenciar Hardware e Software:

Hardware é aquilo que você chuta, Software é aquilo que você xinga.

2.3 - Estrutura do sistema de arquivos



Árvore do Sistema de Arquivos Linux

O sistema GNU/Linux possui a seguinte estrutura básica de diretórios:

- /** Diretório principal (raiz).
- /bin** Contém arquivos de programas do sistema que são usados com frequência pelos usuários.
- /boot** Contém arquivos necessários para a inicialização do sistema.
- /cdrom** Ponto de montagem da unidade de CD-ROM.
- /dev** Contém arquivos usados para acessar dispositivos (periféricos) existentes no computador.
- /etc** Arquivos de configuração de seu computador local.
- /floppy** Ponto de montagem de unidade de disquetes.
- /home** Diretórios contendo os arquivos dos usuários.
- /lib** Bibliotecas compartilhadas pelos programas do sistema e módulos do kernel.
- /lost+found** Local para a gravação de arquivos/diretórios recuperados pelo utilitário fsck.ext2. Cada partição possui seu próprio diretório **lost+found**.
- /mnt** Ponto de montagem temporário.
- /proc** sistema de arquivos do kernel. Este diretório não existe em seu disco rígido, ele é colocado lá pelo kernel e usado por diversos programas que fazem sua leitura, verificam configurações do sistema ou modificar o funcionamento de dispositivos do sistema através da alteração em seus arquivos.
- /root** Diretório do usuário *root*
- /sbin** Diretório de programas usados pelo superusuário (*root*) para administração e controle do funcionamento do sistema.
- /tmp** Diretório para armazenamento de arquivos temporários criados por programas.
- /usr** Contém maior parte de seus programas. Normalmente acessível somente como leitura.
- /var** Contém maior parte dos arquivos que são gravados com frequência pelos programas do sistema, e-mails, spool de impressora, cache, etc.

2.4 - Sistema de arquivos

- *ext2* - Para partições *GNU/Linux* usando o Extended File System versão 2 (a mais comum).
- *ext3* - Para partições *GNU/Linux* usando o Extended File System versão 3, com suporte a journaling.
- *reiserfs* - Para partições reiserfs, com suporte a journaling.
- *vfat* - Para partições *Windows 95* que utilizam nomes extensos de arquivos e diretórios.
- *msdos* - Para partições *DOS* normais.
- *iso9660* - Para montar unidades de *CD-ROM*. É o padrão.

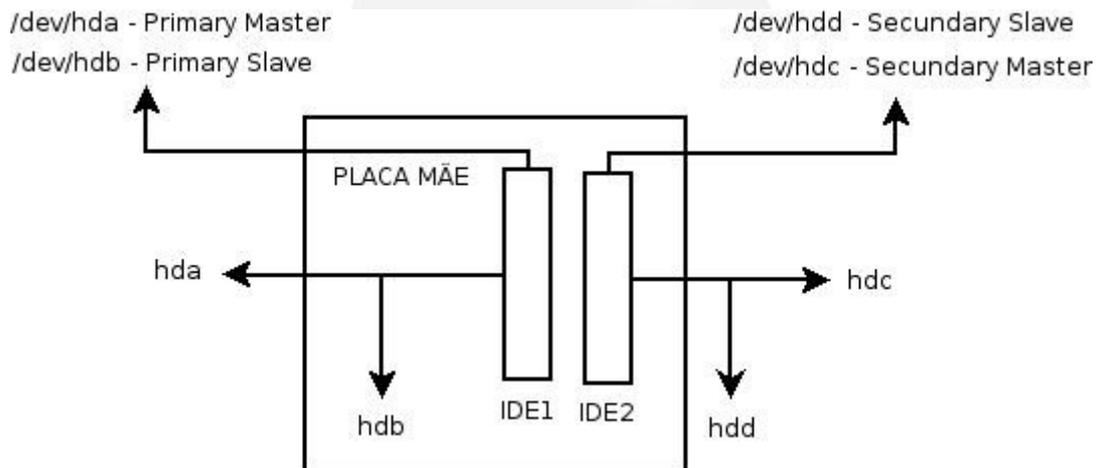
No *GNU/Linux*, os dispositivos existentes em seu computador (como discos rígidos, disquetes, portas de impressora, modem, etc) são identificados por um arquivo referente a este dispositivo no diretório */dev*.

2.5 - Dispositivos

Os nomes de dispositivos no sistema *GNU/Linux* são acessados através do diretório onde esses dispositivos físicos são tratados como arquivos. Estes arquivos são um tipo especial no sistema de arquivos *Linux*. Esses dispositivos são: impressoras, *CD-ROMs*, modems, entradas *USB*, mouse, Hds, etc.

Dispositivos Linux	Dispositivos Windows
<i>/dev/hda1</i>	C: (Partição 1 da IDE Primária Master)
<i>/dev/hda2</i>	D: (Partição 2 da IDE Primária Master)
<i>/dev/ttyS0</i>	COM1 (Porta serial 1)
<i>/dev/ttyS1</i>	COM2 (Porta serial 2)
<i>/dev/ttyS2</i>	COM3 (Porta serial 3)
<i>/dev/ttyS3</i>	COM4 (Porta serial 4)
<i>/dev/fd0</i>	disquete(drive A:)
<i>/dev/lp0</i>	LPT1(porta paralela - impressora)
<i>/dev/dsp</i>	Acesso a sua placa de som pelo canal digital
<i>/dev/mixer</i>	Acesso ao mixer da sua placa de som

A identificação de discos rígidos no GNU/Linux é feita da seguinte forma:



*Representação gráfica da placa mãe e barramentos IDE

/dev/hda1

- | | | | |
- | | | | |_Número que identifica o número da partição no disco rígido.
- | | | | |_Letra que identifica o disco rígido (a=primeiro, b=segundo, etc...).
- | | | | |_Sigla que identifica o tipo do disco rígido (hd=ide, sd=SCSI).
- | | | | |_Diretório onde são armazenados os dispositivos existentes no sistema.

Parte 3 – Introdução ao Linux

Inicialmente o Linux vinha sendo utilizado apenas como servidores por conta de sua estabilidade, performance e segurança, porém há alguns anos veio a pergunta: Porque apenas servidores? E se eu quiser um sistema operacional Livre, Poderoso e Seguro para usar em minha casa para minhas atividades do dia a dia?

Foi essa pergunta que motivou desenvolvedores da comunidade Linux espalhados pelo mundo inteiro para começarem a desenvolver aplicações em modo gráfico de forma a tornar o Linux o mais usual possível para Desktops.

Hoje em dia a quantidade de aplicativos para Desktop Linux já é incontável contando com softwares de altíssima qualidade e confiabilidade como por exemplo o navegador Mozilla Firefox, o OpenOffice...

O nosso foco neste curso não é o de manipular a interface gráfica, mas sim manipular o que acontece por trás das coisas, sendo assim estaremos utilizando o nosso querido Terminal ou Console para o restante do curso, afinal de contas é isso que diferencia um administrador de sistemas Linux do administrador de outros sistemas operacionais. Não fiquemos presos apenas à uma janelinha bonita e bem desenhada.

3.1 – Entendendo uma Shell

Ao abrir um terminal, o mesmo lhe trará de início alguma shell, como o exemplo a seguir:

```
kalib@tuxcaverna:~$
```

Explicando esta shell podemos perceber alguns aspectos:

- kalib = Nome do Usuário
- tuxcaverna = Nome da Máquina
- ~ = Diretório Home do Usuário
- \$ = Significa que você está logado como um usuário comum

Outro exemplo de shell seria a seguinte:

```
root@tuxcaverna:/home/kalib/imgs#
```

Explicando esta shell podemos concluir:

- root = Superusuário do sistema
- tuxcaverna = Nome da Máquina
- /home/kalib/imgs = Árvore de Diretórios onde você se encontra no momento
- # = Significa que você está logado como root

Mas também pode acontecer de você pegar uma shell que ainda não esteja logada. Veremos agora como é simples fazer um login seja como usuário normal ou como superusuário root.

3.2 – Efetuando Login e Logout

```
tuxcaverna login:
```

Ele nos apresenta o nome da máquina e o pedido de login. Basta inserir o nome de seu usuário, no meu caso é kalib, e em seguida ele lhe pedirá a senha. Repare que ao digitar a sua senha de usuário, a mesma não aparece na tela obviamente por questões básicas de segurança. Digitada a senha, aperte enter e você estará na shell inicial de seu usuário apontando para sua pasta home.

```
kalib@tuxcaverna:~$
```

Para mudar de usuário utilize o comando `su + nome do usuário`. Caso deseje mudar para o usuário `root`, o simples `su` será o suficiente. Ao utilizar este comando, ele lhe pedirá a senha do usuário com o qual você deseja logar. Por exemplo:

```
Logando como usuário teste:  
kalib@tuxcaverna:~$ su teste
```

```
Logando como super usuário root:  
kalib@tuxcaverna:~$ su
```

Para encerrar o login ou deslogar pode-se usar as teclas `Ctrl + D` ou simplesmente digitar o comando `logout`, ou mesmo o comando `exit` que serve para encerrar uma sessão de shell. Exemplo:

```
kalib@tuxcaverna:~$ logout
```

3.3 – Desligando e Reiniciando a máquina

Alguns comandos como os de desligar e reiniciar a máquina, obviamente, requerem alguns poderes do usuário, sendo assim só podem ser utilizados pelo superusuário `root`.

Para desligar a máquina temos duas possibilidades de comandos:

```
root@tuxcaverna:~# shutdown -h now
```

ou

```
root@tuxcaverna:~# halt
```

Para reiniciar a máquina também possuímos duas opções:

```
root@tuxcaverna:~# shutdown -r now
```

ou

```
root@tuxcaverna:~# reboot
```

3.4 – Comandos de Ajuda

No Linux é muito comum nos depararmos com situações em que não saberemos o que fazer. Para nos ajudar a solucionar esses problemas temos os amigos, o São Google e até mesmo o próprio Linux com seus comandos de ajuda. Comandos que trazem informações, descrições, exemplos de uso e etc.

MAN

O comando `man` serve para consultar os manuais do sistema.

O comando a seguir nos apresenta o manual de informações sobre o comando `ls` assim como as suas possibilidades de uso.

```
kalib@tuxcaverna:~$ man ls
```

OBS: Para sair do man, basta apertar a tecla Q.

INFO

Outro comando útil e que nos apresentará informações sobre nosso sistema é o `info`.

```
kalib@tuxcaverna:~$ info vim
```

WHATIS

Tem a função de lhe dar uma breve e resumida explicação sobre o que é o comando desejado ou aplicação.

```
kalib@tuxcaverna:~$ whatis konqueror
```

HELP

Não podia faltar o comando `help` que é como um arquivo de ajuda.

```
kalib@tuxcaverna:~$ vim --help
```

3.5 – Comandos de Navegação

Os comandos de navegação são os responsáveis por nos permitir uma navegação entre nossos arquivos, pastas, partições e etc.

CD

Muda o diretório de trabalho para um qualquer que desejarmos:

```
kalib@tuxcaverna:~$ cd /home/kalib/mp3
```

Retorna ao diretório raiz:

```
kalib@tuxcaverna:~$ cd /
```

Retorna ao diretório home do usuário:

```
kalib@tuxcaverna:~$ cd
```

ou

```
kalib@tuxcaverna:~$ cd ~
```

Retorna um nível na árvore de diretórios:

```
kalib@tuxcaverna:~$ cd ..
```

PWD

Exibe a árvore de diretório corrente:

```
kalib@tuxcaverna:~$ pwd
```

LS

Exibe o conteúdo dos diretórios:

```
kalib@tuxcaverna:~$ ls
```

Você também pode utilizar alguns parâmetros do ls, sendo eles:

opção explicação

- a exibe arquivos ocultos
- l exibe detalhes
- h exibe detalhes de diretório
- d exibe o tamanho do arquivo junto com a opção
 -l.
- F exibe as características de cada arquivo, sendo
 eles:

/ Diretórios

* arquivo
 executável

@ link simbólico

~ arquivo de backup

= sockets

| Pipes.

Exemplos:

```
kalib@tuxcaverna:~$ ls -a
```

ou

```
kalib@tuxcaverna:~$ ls -la -F
```

ou

```
kalib@tuxcaverna:~$ ls -F /home/kalib/imgs
```

3.6 – Comandos para Manipulação de Arquivos

Comandos gerais para manipulação de arquivos e/ou pastas, criando, editando, deletando, copiando, movendo...

MKDIR

Comando para criar diretórios.

Modo de utilização: *mkdir [opções] <nome_diretório>*

[Opção]	[Descrição]
-p	Cria a árvore de diretórios completa se necessário
-v	Modo verbose. Mostra o nome de cada diretório criado

Exemplos:

```
kalib@tuxcaverna:~$ mkdir
```

ou

```
kalib@tuxcaverna:~$ mkdir -p /home/kalib/teste/teste2/teste3
```

ou

```
kalib@tuxcaverna:~$ mkdir -pv /home/kalib/teste/aqui/agora
```

RMDIR

Comando para remover diretórios vazios.

Modo de utilização: *rmdir [opções] <nome_diretório>*

[Opção]	[Descrição]
-p	Remove árvore de diretórios completa se necessário
-v	Modo verbose. Mostra o nome de cada diretório removido

Exemplos:

```
kalib@tuxcaverna:~$ rmdir teste2 teste3 teste4
```

ou

```
kalib@tuxcaverna:~$ rmdir -pv /home/kalib/teste/teste2/teste3
```

MV

Serve para mover ou renomear arquivos e diretórios

Modo de utilização: *mv [opções] <fonte> <destino>*

[Opção]	[Descrição]
-i	Pede confirmação antes de mover um arquivo que irá remover outro
-v	Exibe o nome do arquivo antes de movê-lo

Exemplos:

```
kalib@tuxcaverna:~$ mv teste.doc teste1.doc
```

ou

```
kalib@tuxcaverna:~$ mv -i teste.doc /home/kalib/imgs/
```

ou

```
kalib@tuxcaverna:~$ mv -v teste* /home/kalib/testes
```

RM

Tem a função de remover arquivos.

Modo de utilização: `rm [opções] <arquivos>`

[Opção]	[Descrição]
-i	Pede confirmação antes de eliminar o arquivo
-f	Força a eliminação sem pedir confirmação
-r	Apaga recursivamente o conteúdo de um diretório
-v	Exibe o nome de cada arquivo antes de eliminá-lo

Exemplos:

```
kalib@tuxcaverna:~$ rm teste.doc teste2.doc teste3.doc
```

ou

```
kalib@tuxcaverna:~$ rm -f teste.jpg teste.gif teste.txt
```

ou

```
kalib@tuxcaverna:~$ mv -r teste2/
```

ou

```
kalib@tuxcaverna:~$ mv -v teste*
```

CP

Utilizado para copiar um ou mais arquivos.

Modo de utilização: `cp [opções] <arquivos_fonte> <arquivos_destino>`

[Opção]	[Descrição]
-a	Mantém os atributos do arquivo
-r	Copia recursivamente todos os arquivos incluindo subdiretórios
-u	Copia apenas quando os arquivos_fonte são mais novos que arquivos_destino
-v	Exibe o nome de cada arquivo antes de copiá-lo

Exemplos:

```
kalib@tuxcaverna:~$ cp texto.doc /home/kalib/arquivo.doc
```

ou

```
kalib@tuxcaverna:~$ cp -a velho.doc novo.doc
```

ou

```
kalib@tuxcaverna:~$ cp -ru /home/kalib/teste/* /home/kalib/textos/
```

SPLIT

Utilizado para repartir arquivos.

O que acontece quando você tem um arquivo grande e não consegue colocá-lo no seu cd virgem, ou no disquete ou onde quer que seja? Bom, existe a possibilidade de partirmos ele em pedaços menores para posteriormente uní-los novamente.

Modo de utilização: `split -b <tamanho> <nome_arquivo_original> <nome_fatias>`

Exemplo:

```
kalib@tuxcaverna:~$ split -b 50k angra-rebirth.mp3 rebirth
```

Este comando irá partir o meu arquivo angra-rebirth.mp3 em vários pedaços de 50k, e utilizará a palavra rebirth para nomear as fatias do mesmo.

Para listar-mos o resultado da operação podemos usar um simples ls:

```
kalib@tuxcaverna:~$ ls
```

Encontraremos o resultado da seguinte forma:

```
--> rebirthaa rebirthab rebirthac rebirthad rebirthae rebirthaf rebirthag angra-rebirth.mp3
```

Depois, para agruparmos novamente os arquivos em um só:

```
kalib@tuxcaverna:~$ cat rebirth* > angra-rebirth.mp3
```

3.7 – Comandos de terminal

Normalmente são os comandos para se limpar uma tela de terminal, escrever algo ou enviar mensagens para outros usuários ou terminais.

CLEAR

Limpa a tela do terminal virtual corrente. Exemplo:

```
kalib@tuxcaverna:~$ clear
```

ECHO

Envia uma mensagem para o terminal.

Exemplo:

```
kalib@tuxcaverna:~$ echo Teste de Mensagem
```

WRITE

Assim como o echo, também serve para enviar uma mensagem para o terminal. O Write possui algumas possibilidades a mais como escolher a qual usuário você deseja enviar a mensagem ou mesmo em que terminal quer que ela seja enviada escolhendo o tty.

O write também pode ser usado para enviar mensagens à outras máquinas que estejam na rede utilizando a ip da máquina juntamente com o usuário.

Modo de utilização: `write<usuário> [tty]`

Digitado este comando, ele lhe permitirá digitar sua mensagem, ao encerrar a mensagem pressione as teclas Ctrl + D para encerrar a tarefa.

Exemplos:

```
kalib@tuxcaverna:~$ write lucia
```

ou

```
kalib@tuxcaverna:~$ write lucia@192.168.0.5 tty2
```

3.8 – Comandos de localização de arquivos

Como o nome já diz são utilizados para se localizar arquivos do sistema de forma rápida e eficiente.

FIND

Procura arquivos no disco rígido.

Modo de utilização: *find [caminho] [opções] <nome_arquivo>*

[Opção]	[Descrição]
-name <padrao>	Localiza arquivos com nome igual a padrao
-type <x>	Localiza especificando o tipo do arquivo de acordo com x:
f	Arquivos
d	Diretórios

Exemplos:

```
kalib@tuxcaverna:~$ find /home/kalib -name teste
```

ou

```
kalib@tuxcaverna:~$ find /home/kalib/imgs -type f
```

ou

```
kalib@tuxcaverna:~$ find /home/kalib -type d
```

WHEREIS

Este comando localiza o arquivo binário, o código-fonte e a página do manual para um comando.

Modo de utilização: *whereis [opções] <comando>*

[Opção]	[Descrição]
-b	Localiza apenas o arquivo binário
-s	Localiza apenas o código-fonte
-m	Localiza apenas a página do manual

OBS: Caso nenhuma das três opções seja utilizada, o comando lhe trará como resposta as três informações.

Exemplos:

```
kalib@tuxcaverna:~$ whereis konqueror
```

ou

```
kalib@tuxcaverna:~$ whereis -b konqueror
```

ou

```
kalib@tuxcaverna:~$ whereis -m konqueror
```

WHICH

Mostra o caminho de um programa.

Exemplo:

```
kalib@tuxcaverna:~$ which firefox
```

3.9 – Comandos de informações do sistema

São comandos que permitem ver/alterar data e hora do sistema assim como nos exibem informações do sistema em si.

CAL

Este comando nos informa o calendário do mês atual com ênfase no dia de hoje. Porém utilizando-se suas opções, pode-se ver o mês e o ano que se desejar.

Modo de utilização: *cal [mês [ano]]*

[Opção]	[Descrição]
Mês	Especifica de qual mês deve ser mostrado o calendário, devendo ser apresentado com dois dígitos: 01 - 12
Ano	Especifica o ano que deve ser mostrado, devendo ser apresentado com quatro dígitos.
-3	Apresenta o mês anterior, o presente e o próximo.
-y	Apresenta o calendário do ano inteiro.

Exemplos:

```
kalib@tuxcaverna:~$ cal 07 1986
```

ou

```
kalib@tuxcaverna:~$ cal -3
```

DATE

Indica a data e hora atual assim como permite definir nova data e hora.

Modo de utilização: *date [MMDDHHmmAAAA]*

[Opção]	[Descrição]
MM	Mês – 2 caracteres
DD	Dia – 2 caracteres
HH	Hora – 2 caracteres
mm	Minutos – 2 caracteres
AAAA	Ano – 4 caracteres

Exemplos:

```
kalib@tuxcaverna:~$ date
```

ou

```
kalib@tuxcaverna:~# date 071706301986 (06:30 - 17/07/1986)
```

UNAME

Exibe informações sobre o sistema operacional e o hardware da máquina caso usado com as devidas opções. Se utilizado sem nenhuma opção, ele apenas nos retorna o nome do sistema operacional.

Modo de utilização: *uname [opções]*

[Opção]	[Descrição]
-a	Exibe todas as opções
-m	Exibe o nome da plataforma
-n	Exibe o nome da máquina
-s	Exibe o nome do Sistema Operacional
-v	Exibe a data de lançamento do Sistema Operacional
-r	Exibe a versão do Sistema Operacional

Exemplos:

```
kalib@tuxcaverna:~$ uname
```

ou

```
kalib@tuxcaverna:~$ uname -a
```

ou

```
kalib@tuxcaverna:~$ uname -n -s
```

ou

```
kalib@tuxcaverna:~$ uname -ns -mv
```

HOSTNAME

Exibe o nome da máquina.

Exemplo:

```
kalib@tuxcaverna:~$ hostname
```

UPTIME

Informa a hora atual e à quanto tempo o sistema está ativo sem ser desligado ou reiniciado.

Exemplo:

```
kalib@tuxcaverna:~$ uptime
```

FREE

Exibe as informações de memória total, em uso e livre.

Modo de utilização: *free [opção]*

[Opção]	[Descrição]
-t	Exibe também o valor total

Exemplos:

```
kalib@tuxcaverna:~$ free
```

ou

```
kalib@tuxcaverna:~$ free -t
```

3.10 – Comandos de exibição e edição de arquivos

MORE

Exibe arquivos texto.

Modo de utilização: *more [opção]*

[Opção]	[Descrição]
+n	Exibe o texto a partir da linha de número n
-s	Exibe o texto transformando múltiplas linhas em branco em apenas uma

Exemplos:

```
kalib@tuxcaverna:~$ more texto1.doc
```

ou

```
kalib@tuxcaverna:~$ more -s texto1.doc
```

ou

```
kalib@tuxcaverna:~$ more +4 -s texto1.doc
```

VIM

O vim é uma excelente ferramenta para edição de textos. Possui dois modos: Modo Comando e Modo Edição. Ao iniciar, o vim se encontra em modo comando. Para iniciar o modo edição pressione a tecla "i" enquanto que para iniciar o modo comando pressione a tecla "esc".

Modo de utilização: *vim* ou *vim <arquivo>*

No modo comando o vim possui algumas teclas de atalho:

[Comando]	[Descrição]
<:> + <w>	Salva o arquivo
<:> + <q>	Sai do arquivo sem salvar as alterações
<:> + <q> + <!>	Força a saída do arquivo sem salvar as alterações
</> <pesquisa>	Busca pela palavra pesquisa no arquivo

Parte 4 – Utilizando o APT

4.1 – O que é o APT

Como explicado anteriormente, por ser um software livre, qualquer pessoa pode pegar o código fonte do Linux em si como Kernel ou de uma distribuição Linux em específico e alterá-lo da maneira que desejar. Por conta disso ao longo dos anos foram surgindo inúmeras distribuições Linux, logo formas diferentes de instalação de programas para adequar-se às particularidades de cada uma delas.

O método que trataremos na apostila é o APT que é o mais focado para nosso objetivo aqui.

O APT (Advanced Packaging Tool) é um conjunto de ferramentas desenvolvido pela Debian para gerenciar os pacotes de uma distribuição Linux da forma mais automatizada possível, de forma que ao instalar este pacote, também sejam atualizados/instalados todos os pacotes necessários para o funcionamento deste.

As distribuições que utilizam o APT como gerenciador de arquivos possuem servidores onde estão localizadas as suas aplicações disponíveis para download e instalação. O que o APT faz é justamente ir até estes servidores para baixar e instalar o pacote e suas dependências. Estes são os chamados Repositórios.

Para saber onde exatamente o nosso sistema deve procurar por estes repositórios, o nosso sistema possui um arquivo que aponta quais os repositórios utilizados por nossa distribuição. O arquivo é: `/etc/apt/sources.list`

Segue um exemplo de arquivos `sources.list`:

```
deb http://us.archive.ubuntu.com/ubuntu/ edgy main restricted
deb-src http://us.archive.ubuntu.com/ubuntu/ edgy main restricted
```

Traduzindo este por exemplo:

```
deb-src http://us.archive.ubuntu.com/ubuntu/ edgy main restricted
```

`deb` = significa que é um repositório do tipo debian

`src` = significa que é um repositório de source code (código fonte)

`http...` = host ou endereço onde se encontra o repositório

Os parâmetros que aparecem após a url não são obrigatórios em todas as distribuições, pois indicam que tipo de repositório é aquele. No nosso caso, Kubuntu, seus repositórios estão divididos em quatro **componentes**:

[Componente]	[Descrição]
Main	Softwares oficialmente suportados
Restricted	Softwares suportados que não estão disponíveis completamente sob uma licença livre
Universe	Softwares mantidos pela comunidade, ou seja, softwares não suportados oficialmente
Multiverse	Softwares que não são "livres"

Agora que sabemos o que é o APT, vejamos algumas de suas funcionalidades:

4.2 – Atualizando o banco de dados com o UPDATE

É recomendado que se dê um UPDATE logo ao instalar uma distribuição Linux assim como antes de instalar qualquer pacote. O update serve para atualizar a sua sourcelist assim como o banco de dados de seus pacotes.

Modo de utilização: `#apt-get update`

4.3 – Procurando por pacotes

Veremos duas formas de procurar pacotes através do APT.

A primeira forma é através de uma busca por descrição. Nos repositórios os pacotes sempre aparecem com uma descrição sobre o mesmo. Para fazer uma busca utilizando-se da descrição, usaremos o comando APT-CACHE SEARCH.

Exemplo:

```
root@tuxcaverna:/home/kalib# apt-cache search game
```

Este comando nos exibirá todos os pacotes que possuem a palavra game na descrição.

A segunda é através do nome do pacote. Com o comando APTITUDE SEARCH você busca por um nome em específico.

Exemplo:

```
root@tuxcaverna:/home/kalib# aptitude search broffice
```

Este comando nos retornará a lista de pacotes que possuem a palavra broffice no nome.

4.4 – Instalando pacotes

Depois de feita a devida busca, é hora de instalar aquilo que queremos. Supondo que ao buscar por broffice, obtive como resposta um pacote chamado broffice.org, vamos instalar o mesmo seguindo o exemplo:

```
root@tuxcaverna:/home/kalib# apt-get install broffice.org
```

4.5 – Removendo pacotes

Se por algum motivo você não deseja mais ter o broffice.org em sua máquina, você pode facilmente removê-lo com o comando APT-GET REMOVE, como no exemplo a seguir:

```
root@tuxcaverna:/home/kalib# apt-get remove broffice.org
```