

Linux



**Bacharelados em
Computação**

Centro Universitário de Anápolis - UniEVANGÉLICA

Título: **Linux** / Ano – 2018

Preparo de Originais

Aline Dayany de Lemos Gleison de Jesus Silva Gustavo Lupercino de Freitas	Lucas Nunes Santana Mateus da Silva Cordeiro
---	---

Equipe editorial:

Revisores:

Prof. Esp. Aline Dayany de Lemos

Prof. M.e. Marcelo de Castro Cardoso

Prof. M.e. William Pereira dos Santos Junior

Coordenadora dos Projetos Interdisciplinares: Prof. M.e Adrielle Beze Peixoto

Diretora dos cursos de Engenharia de Computação e Engenharia de Software: Prof. M.e Viviane

Carla Batista Pocivi

Centro Universitário de Anápolis

Chanceler – Ernei de Oliveira Pina

Reitor – Carlos Hassel Mendes da Silva

Pró-Reitora Acadêmica – Cristiane Martins Rodrigues Bernardes

Pró-Reitor de Pós-Graduação, Pesquisa, Extensão e Ação Comunitária – Sandro Dutra e Silva

Coordenador da Pesquisa e Inovação – Bruno Junior Neves

Coordenador de Extensão e Ação Comunitária – Fábio Fernandes Rodrigues

Cursos de Engenharia de Computação e Engenharia de Software:

Diretora - Viviane Carla Batista Pocivi

Núcleo Docente Estruturante - Adrielle Beze Peixoto

Núcleo Docente Estruturante - Natasha Sophie Pereira

Núcleo Docente Estruturante - Renata Dutra Braga

Núcleo Docente Estruturante - Walquíria Fernandes Marins

Sumário

Linux	1
Indicação de Ícones	5
Palavra da Coordenação	7
Apresentação	9
Sobre o Conteúdo	9
Introdução ao LINUX	9
Algumas Características do Linux	10
Distribuições Linux	13
Processo de Instalação	17
Comandos no Linux	28
Executando programas em seqüência	28
Controle de execução de processos	29
Interrompendo a execução de um processo	29
Parando momentaneamente a execução de um processo	29
Fechando um programa quando não se sabe como sair	31
Comandos para manipulação de diretório	32
Comandos para manipulação de Arquivos	34
Comandos Diversos	35
Comandos de rede	39
Comandos para manipulação de contas	39
Instalando / atualizando e removendo pacotes	41
Entendendo o modo gráfico no Linux	44
Um pouco de história	45
Como usar a Central de programas do Ubuntu	45
Como remover um programa	49
Introdução ao Shell Script	53
Criação Shell Script	53

Permissão de arquivo	54
Editar E Executar O Arquivo	54
Inserir Comentários.....	54
Uso Das Variáveis	54
Entrada De Dados Do Usuário	55
Comando De Seleção.....	55
Funções	55
Inserção De Argumentos	56
Curiosidades	56
Referências Bibliográficas	59

Indicação de Ícones



Saiba mais: oferece novas informações que enriquecem o assunto ou "curiosidades" e notícias recentes relacionadas ao tema estudado.



Glossário: indica a definição de um termo, palavra ou expressão utilizada no texto.



Atenção: indica pontos de maior relevância no texto.



Mídias integradas: sempre que se desejar que os estudantes desenvolvam atividades empregando diferentes mídias: vídeos, filmes, jornais e outras.

Palavra da Coordenação

A formação integral é uma preocupação constante na construção didático-pedagógica dos Bacharelados em Computação do Centro Universitário de Anápolis – UniEvangélica. Sua efetividade é garantida por meio da inserção de projetos interdisciplinares, como disciplinas integradoras na matriz curricular do curso, e objetiva que as habilidades e competências exigidas no perfil do egresso sejam fortalecidas por meio de projetos que contribuam para a integração dos conhecimentos teóricos, técnicos, práticos e interpessoais.

As disciplinas de Projeto Interdisciplinar (PI) promovem a associação entre os diferentes conteúdos, habilidades e cenários em projetos que favoreçam a construção do conhecimento científico, aliado à autoaprendizagem, proatividade, resolução conjunta de problemas, trabalho em equipe, reflexividade, entre outros. Para tanto, estas disciplinas têm início no primeiro período do curso e evolui em uma constante de maturidade pessoal, interpessoal e científica.

Esta revista apresenta o resultado dos trabalhos desenvolvidos nestas disciplinas e apresentados durante o SITES – Seminário Interdisciplinar de Tecnologia e Sociedade. Entendemos que este é um passo importante para o processo que vem sendo construído e intentamos que assim como o desenvolvimento deste material foi enriquecedor aos discentes do curso, o acesso aos mesmos, traga resultados positivos a aqueles que tiverem acesso.

Apresentação

Este material foi produzido por alunos e com a supervisão de professores do curso de Engenharia de Computação da UniEvangélica de Anápolis – Goiás. Este documento tem como intenção atender às competências e habilidades elencadas no Plano de Pedagógico do Curso e reiterado no Plano de Ensino da disciplina de Projeto Interdisciplinar IV.

A construção de um perfil profissional, interpessoal e técnico é um desafio que construímos juntos e ao longo do semestre. Despertar no aluno a capacidade de organizar e estruturar conhecimentos gerais e desafia-lo na tomada de ações além da capacidade de absolvição e reprodução de conteúdo. Proporcionar a construção de conteúdos técnicos para auxiliar colegas e a comunidade em geral, possibilitando a identificação de potencialidades e fraquezas sobre a sua comunicação verbal e escritas.

Em contraponto delíamos ao aluno do minicurso a responsabilidade quanto à sua responsabilidade de aprendizado. Em ambientes EAD, ou a responsabilidade do aluno é significativamente maior em relação à educação presencial. Esta modalidade pode ser um pouco nova para você. Estamos "conectados" em um mesmo local, porém não o estamos fisicamente. O aluno, a partir desta perspectiva, tem total responsabilidade sobre o seu aprendizado, então leia, releia e leia novamente se necessário, mas não deixe que o tempo passe e você não aprenda o que se propôs.

Sobre o Conteúdo

Neste documento vamos aprender um pouco o sistema operacional Linux em uma Distribuição do Ubuntu. O Linux é um sistema operacional livre e aberto, desta forma qualquer pessoa pode utilizá-lo e ainda o modificar. É um sistema operacional muito utilizado em servidores por características de segurança. Agora vamos ler e entender um pouquinho.

Introdução ao LINUX

O Linux é um sistema operacional criado em 1991 por Linus Torvalds na universidade de Helsinki na Finlândia. É um sistema Operacional de código aberto distribuído gratuitamente pela Internet. Seu código fonte é liberado como Free Software (software livre), sob licença GPL, o aviso de copyright do kernel feito por Linus descreve detalhadamente isto e mesmo ele não pode fechar o sistema para que seja usado apenas comercialmente. Isto quer dizer que você não precisa pagar nada para usar o Linux, e não é crime fazer cópias para instalar em outros computadores, nós inclusive incentivamos você a fazer isto. Ser um sistema de código aberto pode explicar a performance, estabilidade e velocidade em que novos recursos são adicionados ao sistema.

O requisito mínimo para rodar o Linux depende do kernel que será usado:

- 2.2.x - Computador 386 SX com 2 MB de memória
- 2.4.x - Computador 386 SX com 4MB de memória
- 2.6.x - Computador 486 DX com no mínimo 8MB

Para espaço em disco é requerido 500MB para uma instalação básica usando modo texto com suporte a rede. Claro que não é considerada a execução de ambiente gráfico ou serviços de rede em produção, que neste caso é exigido mais memória RAM e espaço em disco para armazenamento de dados de programas e usuários.

O sistema segue o padrão POSIX que é o mesmo usado por sistemas UNIX e suas variantes. Assim, aprendendo o Linux você não encontrará muita dificuldade em operar um sistema do tipo UNIX, FreeBSD, HPUX, SunOS, etc., bastando apenas aprender alguns detalhes encontrados em cada sistema.

O código fonte aberto permite que qualquer pessoa veja como o sistema funciona (útil para aprendizado), corrigir algum problema ou fazer alguma sugestão sobre sua melhoria, esse é um dos motivos de seu rápido crescimento, do aumento da compatibilidade de periféricos (como novas placas sendo suportadas logo após seu lançamento) e de sua estabilidade.

Outro ponto em que ele se destaca é o suporte que oferece a placas, CD/DVD-RWs, BluRay e outros tipos de dispositivos de última geração e mais antigos (a maioria deles já ultrapassados e sendo completamente suportados pelo sistema operacional). Este é um ponto forte para empresas que desejam manter seus micros em funcionamento e pretendem investir em avanços tecnológicos com as máquinas que possui.

O Linux é desenvolvido por milhares de pessoas espalhadas pelo mundo, cada uma fazendo sua contribuição ou mantendo alguma parte do kernel gratuitamente. Linus Torvalds ainda trabalha em seu desenvolvimento e na coordenação dos grupos de trabalho do kernel.

O suporte ao sistema também se destaca como sendo o mais eficiente e rápido do que qualquer programa comercial disponível no mercado. Existem milhares de consultores e empresas especializadas no suporte e treinamento espalhados ao redor do mundo. Outra opção de suporte é através da comunidade Linux; você pode se inscrever em uma lista de discussão e relatar sua dúvida ou alguma falha, e sua mensagem será vista por centenas de usuários na Internet e algum irá te ajudar ou avisará as pessoas responsáveis sobre a falha encontrada para devida correção.

Algumas Características do Linux

- É livre e desenvolvido voluntariamente por programadores experientes, hackers, e contribuidores espalhados ao redor do mundo que tem como objetivo a contribuição para a melhoria e crescimento deste sistema operacional. Muitos deles estavam cansados do excesso de propaganda (Marketing) e baixa qualidade de sistemas comerciais existentes.

- Também recebe apoio de grandes empresas como IBM, Sun, HP, etc. para seu desenvolvimento.
- Convivem sem nenhum tipo de conflito com outros sistemas operacionais (com o DOS Windows, OS/2)

no mesmo computador.

- Multitarefa real.
- Multiusuário.
- Suporte a nomes extensos de arquivos e diretórios (255 caracteres).
- Conectividade com outros tipos de plataformas como Apple, Sun, Macintosh, Sparc, Alpha PowerPc, ARM, Unix, Windows, DOS, etc.

- Utiliza permissões de acesso a arquivos, diretórios e programas em execução na memória RAM.
- Proteção entre processos executados na memória RAM

- Suporte a mais de 63 terminais virtuais (consoles)
- Modularização - O Linux somente carrega para a memória o que é usado durante o processamento, liberando totalmente a memória assim que o programa/dispositivo é finalizado.
 - Devido a modularização, os drivers dos periféricos e recursos do sistema podem ser carregados e removidos completamente da memória RAM a qualquer momento. Os drivers (módulos) ocupam pouco espaço quando carregados na memória RAM (cerca de 6Kb para a Placa de rede NE 2000, por exemplo)
 - Suporte nativo a rede e tecnologias avançadas como: balanceamento de carga, ips alias, failover, vlans, bridge, trunking, OSPF, BGP.
 - Não há a necessidade de se reiniciar o sistema após a modificar a configuração de qualquer periférico ou parâmetros de rede. Somente é necessário reiniciar o sistema no caso de uma instalação interna de um novo periférico, falha em algum hardware (queima do processador, placa mãe, etc.).
 - Não precisa de um processador potente para funcionar. O sistema roda bem em computadores 386Sx 25 com 4MB de memória RAM (sem rodar o sistema gráfico X, que é recomendado 32MB de RAM). Já pensou no seu desempenho em um Pentium, Xeon, ou Athlon?).
- Suporte nativo a múltiplas CPUs, assim processadores como Dual Core, Core Duo, Athlon Duo, Quad Core tem seu poder de processamento integralmente aproveitado, tanto em 32 ou 64 bits.
- Suporte nativo a dispositivos SATA, PATA, Fiber Channel
- Suporte nativo a virtualização, onde o Linux se destaca como plataforma preferida para execução de múltiplos sistemas operacionais com performance e segurança.
 - O crescimento e novas versões do sistema não provocam lentidão, pelo contrário, a cada nova versão os desenvolvedores procuram buscar maior compatibilidade, acrescentar recursos úteis e melhor desempenho do sistema (como o que aconteceu na passagem do kernel 2.0.x para 2.2.x, da 2.2.x para a 2.4.x e da 2.4.x para a 2.6.x).
 - O GNU/Linux é distribuído livremente e licenciado de acordo com os termos da GPL.
 - Acessa corretamente discos formatados pelo DOS, Windows, Novell, OS/2, NTFS, SunOS, Amiga, Atari, Mac, etc.
 - O LINUX NÃO É VULNERÁVEL A VÍRUS! Devido a separação de privilégios entre processos e respeitadas as recomendações padrão de política de segurança e uso de contas privilegiadas (como a de root, como veremos adiante), programas como vírus tornam-se inúteis pois tem sua ação limitada pelas restrições de acesso do sistema de arquivos e execução. Qualquer programa (nocivo ou não) poderá alterar partes do sistema que possui permissões (será abordado como alterar permissões e tornar seu sistema mais restrito no decorrer do guia). Frequentemente são criados exploits que tentam se aproveitar de falhas existentes em sistemas desatualizados e usa-las para causar danos. Erroneamente este tipo de ataque é classificado como vírus por pessoas mal informadas e são resolvidas com sistemas bem mantidos. Em geral, usando uma boa distribuição que tenha um eficiente sistema de atualização e bem configurado, você terá 99.9% de sua tranquilidade.
 - Rede TCP/IP mais rápida que no Windows e tem sua pilha constantemente melhorada. O GNU/Linux tem suporte nativo a redes TCP/IP e não depende de uma camada intermediária como o WinSock. Em acessos via modem a Internet, a velocidade de transmissão é 10% maior.
 - Executa outros sistemas operacionais como Windows, MacOS, DOS ou outro sistema Linux através de consagrados sistemas de virtualização como Xen, vmware, ou emulação como o DOSEMU, QEMU, WINE.
 - Suporte completo e nativo a diversos dispositivos de comunicação via infravermelho,

Bluetooth, Firewire, USB. Basta conectar e o seu dispositivo é automaticamente reconhecido. Raramente são necessários drivers externos, exceto no caso de dispositivos muito novos que não tenham o suporte ainda adicionado no sistema.

- Suporte a rede via rádio amador.
- Suporte a dispositivos Plug-and-Play.
- Suporte nativo a pendrivers, dispositivos de armazenamento e cartões de memória.
- Suporte nativo a dispositivos I2C
- Integração com gerenciamento de energia ACPI e APM
- Dispositivos de rede Wireless. Tanto com criptografia WEB e WPA PSK
- Vários tipos de firewalls avançados de alta qualidade na detecção de tráfego indesejável dando ao administrador uma excelente ferramenta de proteção e controle de sua rede.

- Roteamento estático e dinâmico de pacotes.
- Ponte entre Redes, proxy arp.
- Proxy Tradicional e Transparente.
- Possui recursos para atender a mais de um endereço IP na mesma placa de rede, sendo muito útil para situações de manutenção em servidores de redes ou para a emulação de "múltiplos computadores". O servidor WEB e FTP podem estar localizados no mesmo computador, mas o usuário que se conecta tem a impressão que a rede possui servidores diferentes.

- Os sistemas de arquivos usados pelo GNU/Linux (Ext2, Ext3, reiserfs, xfs, jfs) organiza os arquivos de forma inteligente evitando a fragmentação e fazendo-o um poderoso sistema para aplicações multi-usuárias exigentes e gravações intensivas.

- Permite a montagem de um servidor de publicação Web, E-mail, News, etc. com um baixo custo e alta performance. O melhor servidor Web do mercado, o Apache, é distribuído gratuitamente junto com a maioria das distribuições Linux. O mesmo acontece com o Sendmail.

- Por ser um sistema operacional de código aberto, você pode ver o que o código fonte (instruções digitadas pelo programador) faz e adapta-lo as suas necessidades ou de sua empresa. Esta característica é uma segurança a mais para empresas sérias e outros que não querem ter seus dados roubados (você não sabe o que um sistema sem código fonte faz na realidade enquanto está processando o programa).

- Suporte a diversos dispositivos e periféricos disponíveis no mercado, tanto os novos como obsoletos.
- Pode ser executado em 16 arquiteturas diferentes (Intel, Macintosh, Alpha, Arm, etc.) e diversas outras sub-arquiteturas.

- Empresas especializadas e consultores especializados no suporte ao sistema espalhados por todo o mundo.

- Entre muitas outras características que você descobrirá durante o uso do sistema (além de poder criar outras, caso seja um administrador avançado ou desenvolvedor).

TODOS OS ITENS DESCRITOS ACIMA SÃO VERDADEIROS E TESTADOS PARA QUE TIVESSE PLENA CERTEZA DE SEU FUNCIONAMENTO.

Distribuições Linux

Agora veremos as principais distribuições e suas funcionalidades, a escolha de qual utilizar depende somente do usuário, que escolherá a partir da necessidade e familiaridade com o sistema.

Só o kernel GNU/Linux não é suficiente para se ter um sistema funcional, mas é o principal. Existem grupos de pessoas, empresas e organizações que decidem “distribuir” o Linux junto com outros aplicativos (como por exemplo editores gráficos, planilhas, bancos de dados, ambientes de programação, formatação de documentos, firewalls, etc).

Este é o significado essencial de distribuição. Cada distribuição tem sua característica própria, 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 de cada um.

Algumas distribuições bastante conhecidas são: Ubuntu, Debian, Slackware, Red Hat, Gentoo, Suse todas usando o SO Linux como kernel principal (a Debian é uma distribuição independente de kernel e pode ser executada sob outros kernels, como o GNU hurd ou o kernel BSD).

Segue abaixo as características de algumas distribuições seguidas do site principal e endereço para download:



Figura 1 - Distribuição Linux - Debian

Distribuição desenvolvida e atualizada através do esforço de voluntários espalhados ao redor do mundo, seguindo o estilo de desenvolvimento GNU/Linux. Por este motivo, foi adotada como a distribuição oficial do projeto GNU. Possui suporte a língua Portuguesa, é a única que tem suporte a 14 arquiteturas diferentes (i386, IA64, AMD64, Alpha, Sparc, PowerPC, Macintosh, Arm, etc.) e aproximadamente 15 sub-arquiteturas. A instalação da distribuição pode ser feita tanto através de Disquetes, CD-ROM, Tftp, Ftp, NFS ou através da combinação de vários destes em cada etapa de instalação. Acompanha mais de 25000 programas distribuídos em forma de pacotes cada um destes programas é mantido e testado pela pessoa ou grupo responsável por seu empacotamento. Possui tanto ferramentas para administração de redes e servidores quanto para desktops, estações multimídia, jogos, desenvolvimento, web, etc.

É a única distribuição não comercial onde todos podem contribuir usando seu conhecimento para o desenvolvimento. Para gerenciar os voluntários, conta com centenas de listas de discussão envolvendo determinados desenvolvedores das mais diversas partes do mundo.

O suporte ao usuário e desenvolvimento da distribuição são feitos através de listas de discussões e canais IRC. Existem uma lista de consultores habilitados a dar suporte e assistência a sistemas Debian ao redor do mundo na área consultores do site principal da distribuição.

Prós:

- É mantido por seus próprios usuários;
- Suporte não-paralelo;
- O melhor sistema de empacotamento do mundo;
- Fácil instalação;

Contras:

- Falta de software comercial popular;
- O Debian é difícil de configurar;
- Nem todo hardware é suportado;

- Endereço para download: [//ftp.debian.org/](http://ftp.debian.org/)



Figura 2 - Distribuição Linux - Ubuntu

Ubuntu é um sistema operacional de código aberto, construído a partir do núcleo Linux, baseado no Debian. Variante da distribuição Debian voltada a interação mais amigável com o usuário final e facilidade de instalação. Atualmente é a melhor para usuários que tem o primeiro contato com o Linux. Conta tanto com a instalação do sistema em HD e execução através de Live CD.

Prós:

- Sistema operacional totalmente gratuito;

- Segurança que o sistema oferece;
- Imune a vírus e outros malwares;
- Utilização do hardware de forma inteligente;

Contras:

- É necessário um nível de conhecimento maior para fazer muitas tarefas consideradas comuns;
- Interface pouco intuitiva;
- A configuração do sistema também exige bastante do usuário;

Endereço para download: <http://www.ubuntu.com/getubuntu/download/>



Figura 3 - Distribuição Linux - Suse

Distribuição comercial Alemã com a coordenação sendo feita através dos processos administrativos dos desenvolvedores e de seu braço norte americano. O foco da Suse é o usuário com conhecimento técnico no Linux (programador, administrador de rede, etc.) e não o usuário iniciante no Linux. Preferencialmente a administração deve ser feita usando o Yast, mas também pode ser feita manualmente através de alteração dos arquivos de configuração.

Uma média de 2000 programas acompanham a versão 10 distribuídos em 6 CD-ROMs. O sistema de gerenciamento de pacotes é o RPM padronizado. A seleção de pacotes durante a instalação pode ser feita através da seleção do perfil de máquina (developer, estação kde, gráficos, estação gnome, servidor de rede, etc.) ou através da seleção individual de pacotes.

A atualização da distribuição pode ser feita através do CD-ROM de uma nova versão ou baixando pacotes de <ftp://ftp.suse.com/>. Usuários registrados ganham direito a suporte de instalação via e-mail. A base de dados de suporte também é excelente e está disponível na web para qualquer usuário, independente de registro.

Prós:

- É um sistema prático;
- Uma das distribuições com mais recursos inovadores;

Contras:

- É um sistema lento, pesado;
- É um sistema pouco estável;
- Necessita de conhecimentos avançados para utilização;

Endereço para download: <ftp://ftp.suse.com/>



Figura 4 - Distribuição Linux - Redhat

Distribuição comercial suportada pela Red Hat e voltada a servidores de grandes e médias empresas. Também conta com uma certificação chamada RHCE específica desta distribuição.

Ela não está disponível para download, apenas vendida a custos a partir de 179 dólares (a versão workstation) até 1499 dólares (advanced server).

Prós:

- Possui versões estáveis;
- Atualizações constantes;
- Líder no mercado enterprise;

Contras:

- Licenciamento não é gratuito;
- Requer mais do hardware;

Não possui endereço para download devido ao licenciamento.



Figura 5 - Distribuição Linux - Fedora

O Fedora Linux é a distribuição de desenvolvimento aberto patrocinada pela RedHat e pela comunidade, originada em 2002 e baseada em versão da antiga linha de produtos RedHat Linux. Esta distribuição não é suportada pela Red Hat como distribuição oficial (ela suporta apenas a linha Red Hat Enterprise Linux), devendo obter suporte através da comunidade ou outros meios.

A distribuição Fedora dá prioridade ao uso do computador como estação de trabalho.

Além de contar com uma ampla gama de ferramentas de escritório possui funções de servidor e aplicativos para produtividade e desenvolvimento de softwares. Considerado um dos sistemas mais fáceis de instalar e utilizar, inclui tradução para português do Brasil e suporte às plataformas Intel e 64 bits.

Por basear-se no RedHat, o Fedora conta com um o up2date, um software para manter o sistema atualizado e utiliza pacotes de programas no formato RPM, um dos mais comuns.

Prós:

- Serve perfeitamente os usuários iniciantes e intermediários;
- É um sistema operacional prático;

Contras:

- Não é de fácil instalação;
- Possui pacotes de atualizações extensos;

Endereço

para

download:

<http://download.fedora.redhat.com/pub/fedora/linux/core/2/i386/iso/>

Processo de Instalação

Ubuntu é a distribuição concorrente do famoso Windows, sendo assim nesse curso iremos fazer uma introdução de utilização dessa distribuição. Segue abaixo o passo a passo da instalação do ubuntu 14.10. Para baixar o arquivo do ubuntu 14.10 acesse o o seguinte link <http://www.tecmundo.com.br/ubuntu/64938-versao-final->

ubuntu-14-10-baixada.htm. O arquivo se encontra em ISO, você poderá montar uma imagem em um DVD utilizando o programa DAEMON Tools Lite, ou se preferir você pode montar a imagem em um pendrive utilizando o programa Universal-USB-Installer disponível para download no seguinte endereço: <http://www.pendrivelinux.com/universal-usb-installer-easy-as-1-2-3>, particularmente prefiro montar a imagem no pendrive.

Passo 1 – Coloque o CD/DVD ou pendrive de instalação, e configure o boot pelo setup de seu computador. Depois de serem carregados alguns componentes, é apresentada a primeira janela de configuração. Neste momento, o usuário decide qual o idioma que pretende e se:

- quer apenas experimentar o Ubuntu (Experimente o Ubuntu), nesta opção você pode acessar os dados do seu computador sem precisar instalar o Ubuntu.
- ou instalar o Ubuntu (Instalar Ubuntu)

Para este tópico vamos proceder à instalação completa do Ubuntu.

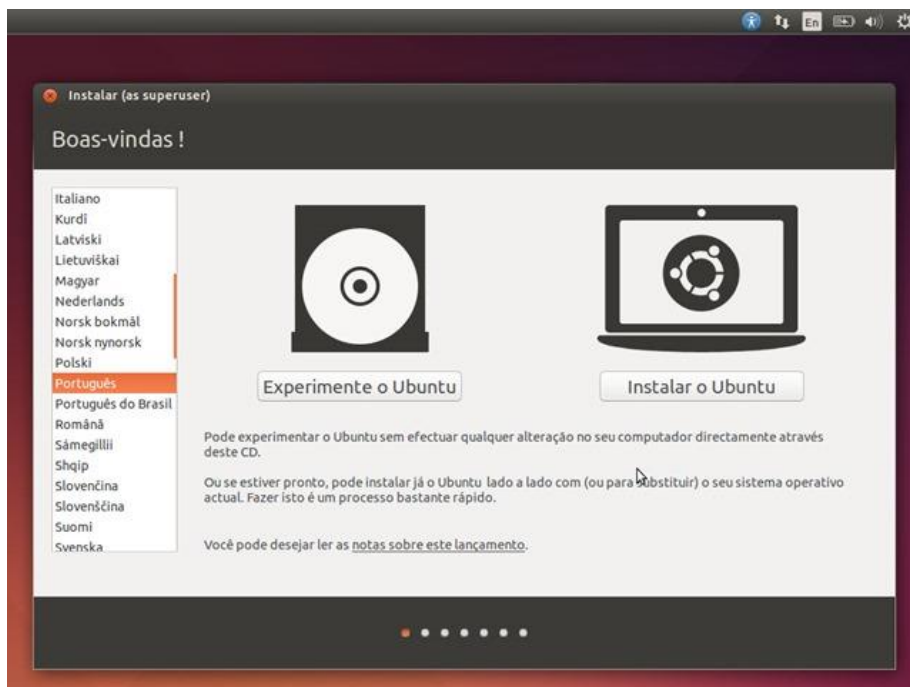


Figura 6 - Instalação Linux

Passo 2 – Nesta fase são fornecidas algumas indicações a nível de requisitos para instalação do Ubuntu, e é dada a possibilidade ao utilizador de indicar se pretende instalar algum software adicional para suporte a alguns formatos populares (ex. MP3, flash, MPEG, etc).

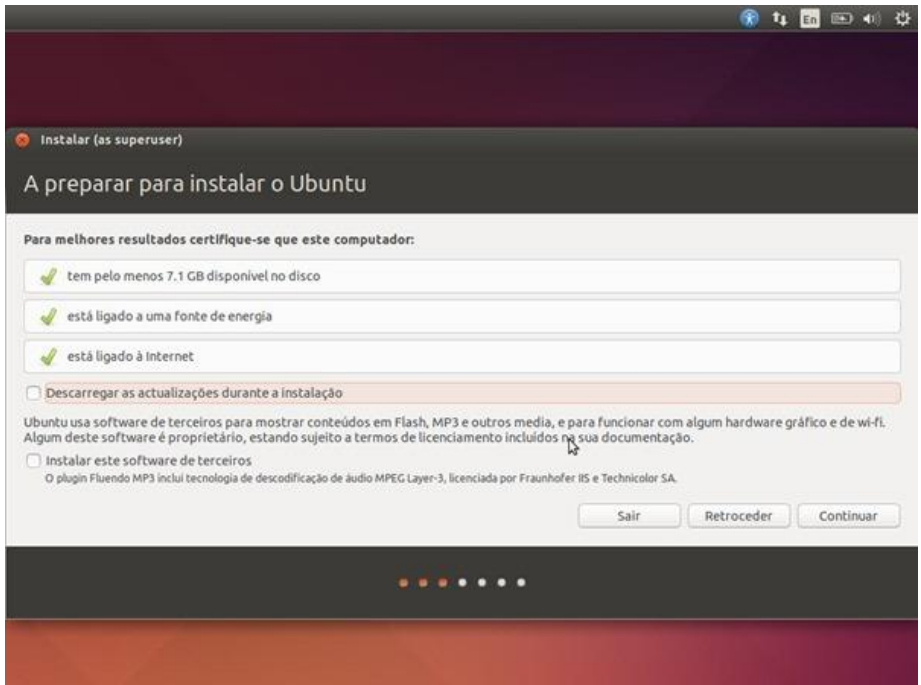


Figura 7 - Instalação Linux

Passo 3 – Particionamento do Disco. Para quem pretender definir partições, poderá ver aqui como fazer. Para este exemplo, definimos a opção Apagar disco e instalar Ubuntu que irá proceder a formatação de todo o disco e instalação do Ubuntu.

Atenção: Ao escolher esta opção, toda a informação do disco será apagada. Caso pretenda usar uma determinada partição, deverá escolher a opção Mais uma coisa.



Figura 8 - Instalação Linux

Passo 4 – Vamos agora definir o nosso fuso horário

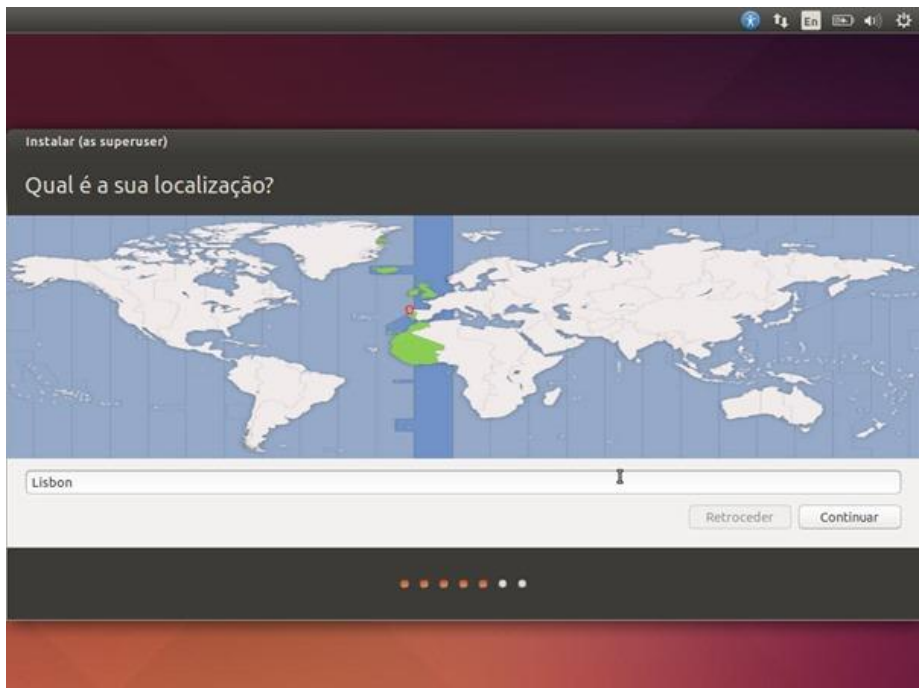


Figura 9 - Instalação Linux

Passo 5 – E agora indicamos qual o layout do nosso teclado

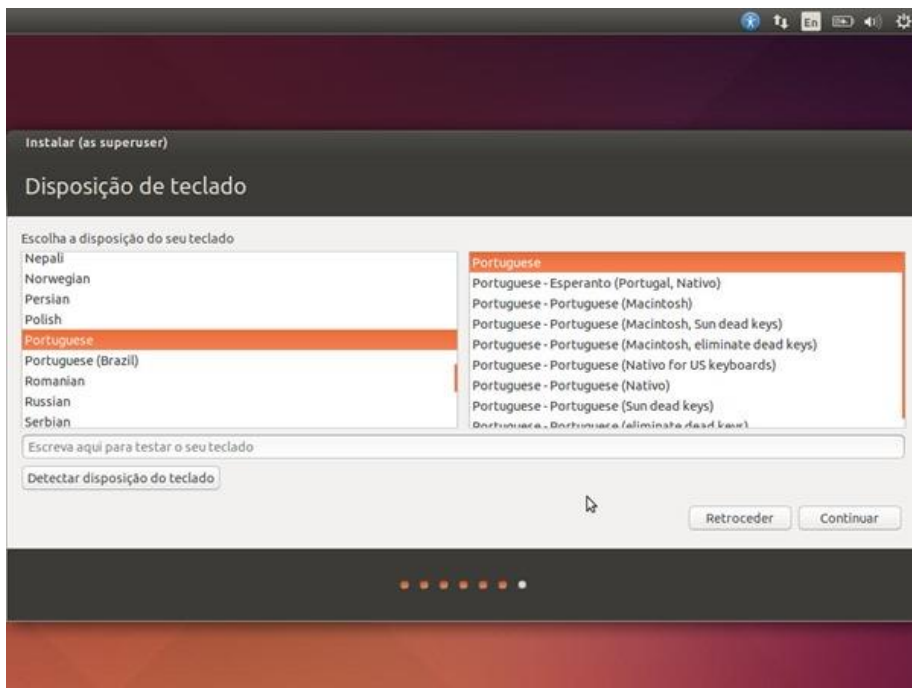


Figura 10 - Instalação Linux

Passo 6 – Neste passo criamos o primeiro usuário assim com a respectiva senha. Definimos também qual o nome da máquina.

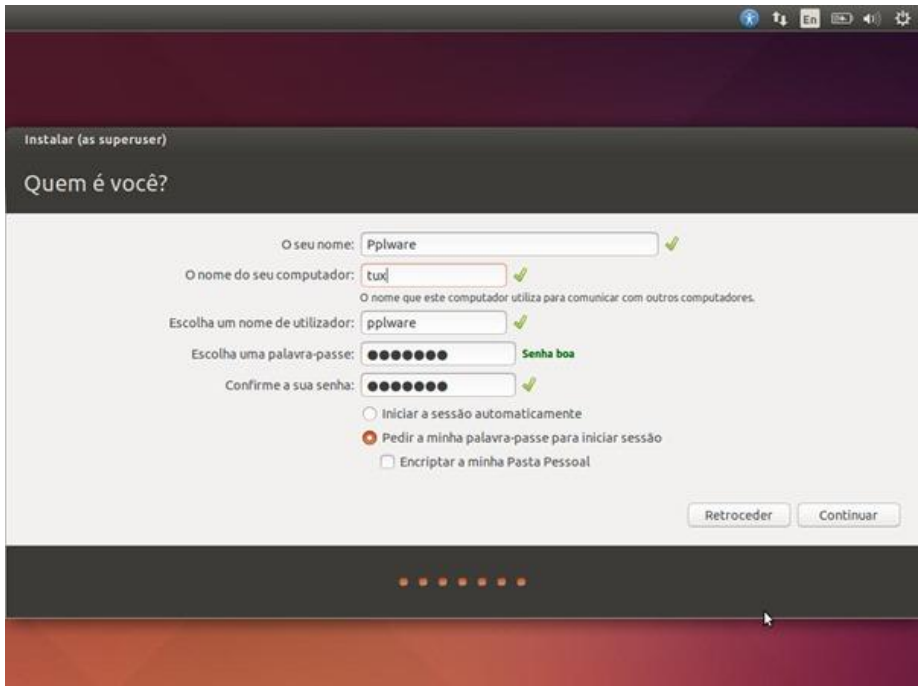


Figura 11 - Instalação Linux

...início do processo de copia de ficheiros (no meu caso demorou cerca de 20 minutos)



Figura 12 - Instalação Linux

Depois de instalado é só reiniciar.

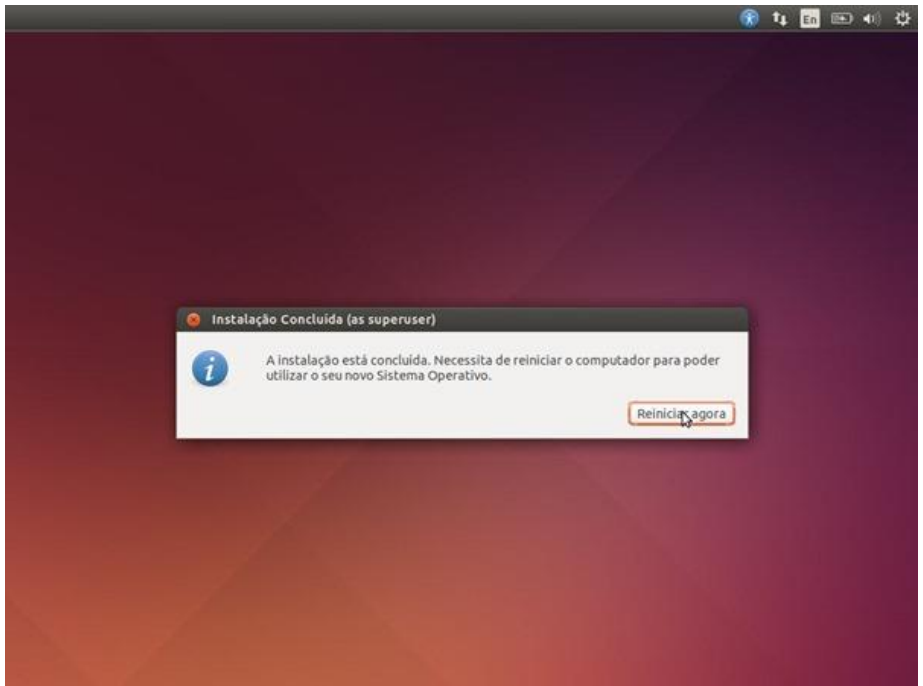


Figura 13 - Instalação Linux

Depois de reiniciar, basta introduzir as credenciais para autenticação definidas no passo 6.

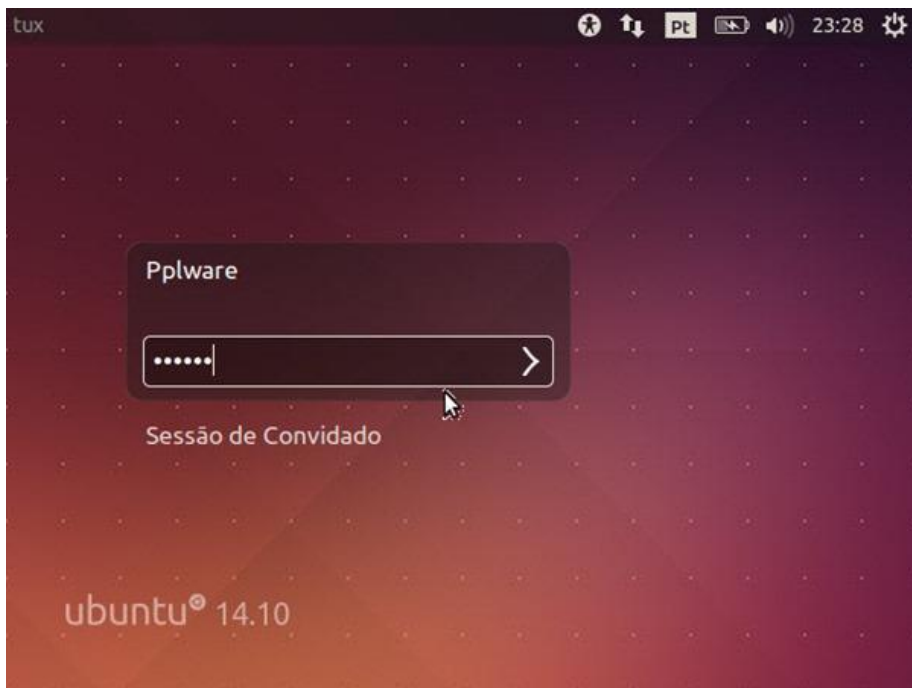


Figura 14 - Instalação Linux

Agora é só usar e explorar!

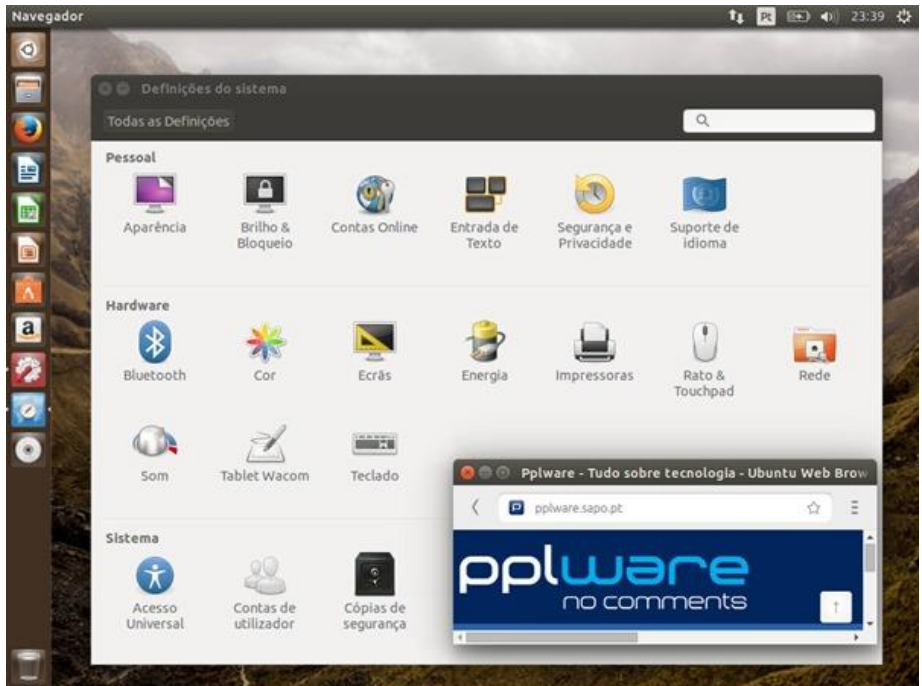


Figura 15 - Instalação Linux

Como as versões anteriores, o Ubuntu 14.10 tem uma boa performance, o que o torna agradável, garantindo uma boa experiência ao utilizador.

Neste tópico ainda temos um vídeo do canal Diolinux, que ensina de forma simples como instalar o Ubuntu 14.04 em uma máquina virtual. Para você que não conhece o termo máquina virtual dê uma olhada no seguinte link https://pt.wikipedia.org/wiki/M%C3%A1quina_virtual.

Se você quiser usar uma máquina virtual faça o download no seguinte link <https://www.virtualbox.org/wiki/Downloads>.

Comandos no Linux

Tipos de Execução de comandos/programas

Um programa pode ser executado de duas formas:

1 Primeiro Plano - Também chamado de foreground. Quando você deve esperar o término da execução de um programa para executar um novo comando. Somente é mostrado o aviso de comando após o término de execução do comando/programa.

2 Segundo Plano - Também chamado de background. Quando você não precisa esperar o término da execução de um programa para executar um novo comando. Após iniciar um programa em background, é mostrado um número PID (identificação do Processo) e o aviso de comando é novamente mostrado, permitindo o uso normal do sistema. O programa executado em background continua sendo executado internamente. Após ser concluído, o sistema retorna uma mensagem de pronto acompanhado do número PID do processo que terminou.

Para iniciar um programa em primeiro plano, basta digitar seu nome normalmente. Para iniciar um programa em segundo plano, acrescente o caracter "&" após o final do comando.

OBS: Mesmo que um usuário execute um programa em segundo plano e saia do sistema, o programa continuará sendo executado até que seja concluído ou finalizado pelo usuário que iniciou a execução (ou pelo usuário root).

Exemplo: `find / -name boot.b &`

O comando será executado em segundo plano e deixará o sistema livre para outras tarefas.

Após o comando find terminar, será mostrada uma mensagem.

Executando programas em seqüência

Os comandos podem ser executados em seqüência (um após o término do outro) se os separarmos com ";". Por exemplo: `echo primeiro;echo segundo;echo terceiro`

Capítulo 5. Execução de programas 53

ps

Algumas vezes é útil ver quais processos estão sendo executados no computador. O comando `ps` faz isto, e também nos mostra qual usuário executou o programa, hora que o processo foi iniciado, etc.

`ps [opções]` Onde:

Exemplos: `ps`, `ps ax|grep inetd`, `ps auxf`, `ps auxw`.

Top

Mostra os programas em execução ativos, parados, tempo usado na CPU, detalhes sobre o uso da memória RAM, Swap, disponibilidade para execução de programas no sistema, etc.

Top é um programa que continua em execução mostrando continuamente os processos que estão rodando em seu computador e os recursos utilizados por eles. Para sair do top, pressione a tecla q.

-d [tempo] Atualiza a tela após o [tempo] (em segundos).

-s Diz ao top para ser executado em modo seguro.

-i Inicia o top ignorando o tempo de processos zumbis.

-c Mostra a linha de comando ao invés do nome do programa.

A ajuda sobre o top pode ser obtida dentro do programa pressionando a tecla h ou pela página de manual (man top).

Abaixo algumas teclas úteis:

- espaço - Atualiza imediatamente a tela.
- CTRL+L - Apaga e atualiza a tela.
- h - Mostra a tela de ajuda do programa. É mostrado todas as teclas que podem ser usadas com o top.
- i - Ignora o tempo ocioso de processos zumbis.
- q - Sai do programa.
- k - Finaliza um processo - semelhante ao comando kill. Você será perguntado pelo número de identificação do processo (PID). Este comando não estará disponível caso esteja usando o top com a opção -s.
- n - Muda o número de linhas mostradas na tela. Se 0 for especificado, será usada toda a tela para listagem de processos.

Controle de execução de processos

Abaixo algumas comandos e métodos úteis para o controle da execução de processos no GNU/Linux.

Interrompendo a execução de um processo

Para cancelar a execução de algum processo rodando em primeiro plano, basta pressionar as teclas CTRL+C. A execução do programa será cancelada e será mostrado o aviso de comando. Você também pode usar o comando 'kill' on the next page para interromper um processo sendo executado.

Parando momentaneamente a execução de um processo

Para parar a execução de um processo rodando em primeiro plano, basta pressionar as teclas CTRL+Z. O programa em execução será pausado e será mostrado o número de seu job e o aviso de comando.

Para retornar a execução de um comando pausado, use 'fg' on the facing page ou 'bg' on the next page.

O programa permanece na memória no ponto de processamento em que parou quando ele é interrompido. Você pode usar outros comandos ou rodar outros programas enquanto o programa atual está interrompido.

Jobs

O comando `jobs` mostra os processos que estão parados ou rodando em segundo plano. Processos em segundo plano são iniciados usando o símbolo "&" no final da linha de comando ou através do comando `bg`. `jobs`. O número de identificação de cada processo parado ou em segundo plano (`job`), é usado com os comandos 'fg' on the current page e 'bg' on this page. Um processo interrompido pode ser finalizado usando-se o comando `kill %[num]`, onde `[num]` é o número do processo obtido pelo comando `jobs`.

Kill

Permite enviar um sinal a um comando/programa. Caso seja usado sem parâmetros, o `kill` enviará um sinal de término ao processo sendo executado.

```
kill [opções] [sinal] [número]
```

Onde: número É o número de identificação do processo obtido com o comando 'ps'. Também pode ser o número após o sinal de % obtido pelo comando `jobs` para matar uma tarefa interrompida. Veja 'jobs' on the preceding page.

sinal - Sinal que será enviado ao processo. Se omitido usa -15 como padrão.

opções

-9 Envia um sinal de destruição ao processo ou programa. Ele é terminado imediatamente sem chances de salvar os dados ou apagar os arquivos temporários criados por ele.

Você precisa ser o dono do processo ou o usuário `root` para terminá-lo ou destruí-lo. Você pode verificar se o processo foi finalizado através do comando `ps`. Os tipos de sinais aceitos pelo GNU/Linux são explicados em detalhes em 'Sinais do Sistema' on the current page. Exemplo: `kill 500`, `kill -9 500`, `kill %1`.

Killall

Permite finalizar processos através do nome.

```
killall [opções] [sinal] [processo]
```

Onde: processo Nome do processo que deseja finalizar

sinal - Sinal que será enviado ao processo (pode ser obtido usando a opção -i).

opções

-i Pede confirmação sobre a finalização do processo.

-l Lista o nome de todos os sinais conhecidos.

-q Ignora a existência do processo.

-v Retorna se o sinal foi enviado com sucesso ao processo.

-w Finaliza a execução do killall somente após finalizar todos os processos.

Os tipos de sinais aceitos pelo GNU/Linux são explicados em detalhes na 'Sinais do Sistema'.

Exemplo: killall -HUP inetd

killall5

Envia um sinal de finalização para todos os processos sendo executados.

killall5 [sinal]

Fechando um programa quando não se sabe como sair

Muitas vezes quando se está iniciando no GNU/Linux você pode executar um programa e talvez não saber como fecha-lo. Este capítulo do guia pretende ajuda-lo a resolver este tipo de problema.

Isto pode também ocorrer com programadores que estão construindo seus programas e por algum motivo não implementam uma opção de saída, ou ela não funciona!

Em nosso exemplo vou supor que executamos um programa em desenvolvimento com o nome contagem que conta o tempo em segundos a partir do momento que é executado, mas que o programador esqueceu de colocar uma opção de saída. Siga estas dicas para finaliza-lo:

1-Normalmente todos os programas UNIX (o GNU/Linux também é um Sistema Operacional baseado no UNIX) podem ser interrompidos com o pressionamento das teclas <CTRL> e <C>. Tente isto primeiro para finalizar um programa. Isto provavelmente não vai funcionar se estiver usando um Editor de Texto (ele vai entender como um comando de menu). Isto normalmente funciona para comandos que são executados e terminados sem a intervenção do usuário.

Caso isto não der certo, vamos partir para a força! ;-)

2-Mude para um novo console (pressionando <ALT> e <F2>), e faça o login como usuário root.

3-Localize o PID (número de identificação do processo) usando o comando: ps ax, aparecerão várias linhas cada uma com o número do processo na primeira coluna, e a linha de comando do programa na última coluna. Caso aparecerem vários processos você pode usar ps ax|grep contagem, neste caso o grep fará uma filtragem da saída do comando ps ax mostrando somente as linhas que tem a palavra "contagem".

4-Feché o processo usando o comando kill PID, lembre-se de substituir PID pelo número encontrado pelo comando ps ax acima.

O comando acima envia um sinal de término de execução para o processo (neste caso o programa contagem). O sinal de término mantém a chance do programa salvar seus dados ou apagar os arquivos temporários que criou e então ser finalizado, isto depende do programa.

5-Altere para o console onde estava executando o programa contagem e verifique se ele ainda está em execução. Se ele estiver parado mas o aviso de comando não está disponível, pressione a tecla <ENTER>. Frequentemente acontece isto com o comando kill, você finaliza um programa mas o aviso de comando não é mostrado até que se pressione <ENTER>.

6-Caso o programa ainda não foi finalizado, repita o comando kill usando a opção -9:

kill -9 PID. Este comando envia um sinal de DESTRUIÇÃO do processo, fazendo ele terminar "na marra"!

Uma última dica: todos os programas estáveis (todos que acompanham as boas distribuições GNU/Linux) tem sua opção de saída. Lembre-se que quando finaliza um processo todos os dados do programa em execução podem ser perdidos principalmente se estiver em um editor de textos), mesmo usando o kill sem o parâmetro -9.

Procure a opção de saída de um programa consultando o help on line, as páginas de manual, a documentação que acompanha o programa.

Comandos para manipulação de diretório

Abaixo comandos úteis para a manipulação de diretórios.

ls

Lista os arquivos de um diretório.

ls [opções] [caminho/arquivo] [caminho1/arquivo1] ...

onde: caminho/arquivo Diretório/arquivo que será listado.

caminho1/arquivo1 Outro Diretório/arquivo que será listado. Podem ser feitas várias listagens de uma só

vez.

Exemplos do uso do comando ls:

- ls - Lista os arquivos do diretório atual.
- ls /bin /sbin - Lista os arquivos do diretório /bin e /sbin
- ls -la /bin - Listagem completa (vertical) dos arquivos do diretório /bin inclusive os ocultos.

cd

Entra em um diretório. Você precisa ter a permissão de execução para entrar no diretório.

cd [diretório]

onde: diretório - diretório que deseja entrar.

Exemplos:

- Usando cd sem parâmetros ou cd ~, você retornará ao seu diretório de usuário (diretório home).
- cd /, retornará ao diretório raiz.
- cd -, retornará ao diretório anteriormente acessado.
- cd ..., sobe um diretório.

• `cd ../[diretório]`, sobe um diretório e entra imediatamente no próximo (por exemplo, quando você está em `/usr/sbin`, você digita `cd ../bin`, o comando `cd` retorna um diretório (`/usr`) e entra imediatamente no diretório `bin` (`/usr/bin`).

pwd

Mostra o nome e caminho do diretório atual.

Você pode usar o comando `pwd` para verificar em qual diretório se encontra (caso seu aviso de comandos não mostre isso).

mkdir

Cria um diretório no sistema. Um diretório é usado para armazenar arquivos de um determinado tipo. O diretório pode ser entendido como uma pasta onde você guarda seus papéis (arquivos). Como uma pessoa organizada, você utilizará uma pasta para guardar cada tipo de documento, da mesma forma você pode criar um diretório vendas para guardar seus arquivos relacionados com vendas naquele local.

`mkdir [opções] [caminho/diretório] [caminho1/diretório1]`

onde: caminho Caminho onde o diretório será criado.

diretório Nome do diretório que será criado.

opções:

-p Caso os diretórios dos níveis acima não existam, eles também serão criados.

-verbose Mostra uma mensagem para cada diretório criado. As mensagens de erro serão mostradas mesmo que esta opção não seja usada.

Para criar um novo diretório, você deve ter permissão de gravação. Por exemplo, para criar um diretório em `/tmp` com o nome de teste que será usado para gravar arquivos de teste, você deve usar o comando "`mkdir/tmp/teste`".

Podem ser criados mais de um diretório com um único comando (`mkdir /tmp/teste /tmp/teste1 /tmp/teste2`).

rmdir

Remove um diretório do sistema. Este comando faz exatamente o contrário do `mkdir`. O diretório a ser removido deve estar vazio e você deve ter permissão de gravação para removê-lo.

`rmdir [caminho/diretório] [caminho1/diretório1]`

onde: caminho Caminho do diretório que será removido.

diretório

Nome do diretório que será removido.

É necessário que esteja um nível acima do diretório(s) que será(ão) removido(s). Para remover diretórios que contenham arquivos, use o comando `rm` com a opção `-r`.

Por exemplo, para remover o diretório `/tmp/teste` você deve estar no diretório `tmp` e executar o comando `rmdir teste`.

Comandos para manipulação de Arquivos

Abaixo, alguns comandos utilizados para manipulação de arquivos.

rm

Apaga arquivos. Também pode ser usado para apagar diretórios e sub-diretórios vazios ou que contenham arquivos.

`rm [opções][caminho][arquivo/diretório] [caminho1][arquivo1/diretório1]`

onde:

`caminho` Localização do arquivo que deseja apagar. Se omitido, assume que o arquivo esteja no diretório atual.

`arquivo/diretório` Arquivo que será apagado.

Exemplos:

- `rm teste.txt` - Apaga o arquivo teste.txt no diretório atual.
- `rm *.txt` - Apaga todos os arquivos do diretório atual que terminam com .txt.
- `rm *.txt teste.novo` - Apaga todos os arquivos do diretório atual que terminam com .txt e também o arquivo teste.novo.
- `rm -rf /tmp/teste/*` - Apaga todos os arquivos e sub-diretórios do diretório /tmp /teste mas mantém o sub-diretório /tmp/teste.
- `rm -rf /tmp/teste` - Apaga todos os arquivos e sub-diretórios do diretório /tmp /teste, inclusive /tmp/teste.
- `rm -f --arquivo--` - Remove o arquivo de nome -arquivo-.

cp

Copia arquivos.

`cp [opções] [origem] [destino]`

onde: `origem` Arquivo que será copiado. Podem ser especificados mais de um arquivo para ser copiado usando "Curingas".

`destino` O caminho ou nome de arquivo onde será copiado. Se o destino for um diretório, os arquivos de origem serão copiados para dentro do diretório.

Exemplos:

- `cp teste.txt teste1.txt` Copia o arquivo teste.txt para teste1.txt.
- `cp teste.txt /tmp` Copia o arquivo teste.txt para dentro do diretório /tmp.
- `cp * /tmp` Copia todos os arquivos do diretório atual para /tmp.
- `cp /bin/*` Copia todos os arquivos do diretório /bin para o diretório em que nos encontramos no momento.
- `cp -R /bin /tmp` Copia o diretório /bin e todos os arquivos/sub-diretórios existentes para o diretório /tmp.
- `cp -R /bin/* /tmp` Copia todos os arquivos do diretório /bin (exceto o diretório /bin)

todos os arquivos/sub-diretórios existentes dentro dele para /tmp.
 cp -R /bin /tmp Copia todos os arquivos e o diretório /bin para /tmp.

mv

Move ou renomeia arquivos e diretórios. O processo é semelhante ao do comando cp mas o arquivo de origem é apagado após o término da cópia.

mv [opções] [origem] [destino]

Onde:

origem Arquivo/diretório de origem.

destino Local onde será movido ou novo nome do arquivo/diretório.

opções

-f, --force Substitui o arquivo de destino sem perguntar.

-i, --interactive Pergunta antes de substituir. É o padrão.

-v, --verbose Mostra os arquivos que estão sendo movidos.

O comando mv copia um arquivo da ORIGEM para o DESTINO (semelhante ao cp), mas após a cópia, o arquivo de ORIGEM é apagado.

Exemplos:

mv teste.txt teste1.txt Muda o nome do arquivo teste.txt para teste1.txt.

mv teste.txt /tmp Move o arquivo teste.txt para /tmp. Lembre-se que o arquivo de origem é apagado após ser movido.

mv teste.txt teste.new (supondo que teste.new já exista) Copia o arquivo teste.txt por cima de teste.new e apaga teste.txt após terminar a cópia.

Comandos Diversos

Comandos de uso diversos no sistema.

Clear

Limpa a tela e posiciona o cursor no canto superior esquerdo do vídeo.

date

Permite ver/modificar a Data e Hora do Sistema. Você precisa estar como usuário root para modificar a data e hora. .

date MesDiaHoraMinuto[AnoSegundos]

Onde: MesDiaHoraMinuto[AnoSegundos] São respectivamente os números do mês, dia, hora e minutos sem espaços. Opcionalmente você pode especificar o Ano (com 2 ou 4 dígitos) e os Segundos.

Find

Procura por arquivos/diretórios no disco. find pode procurar arquivos através de sua data de modificação, tamanho, etc através do uso de opções. find, ao contrário de outros programas, usa opções longas através de um "-".

`find [diretório] [opções/expressão]`

Onde: diretório Inicia a procura neste diretório, percorrendo seu sub-diretórios.

opções/expressão

-name [expressão] Procura pelo nome [expressão] nos nomes de arquivos e diretórios processados.

-depth Processa os sub-diretórios primeiro antes de processar os arquivos do diretório principal.

-maxdepth [num] Faz a procura até [num] sub-diretórios dentro do diretório que está sendo pesquisado.

-mindepth [num] Não faz nenhuma procura em diretórios menores que [num] níveis.

-mount, -xdev Não faz a pesquisa em sistemas de arquivos diferentes daquele de onde o comando `find`

foi executado.

-size [num] Procura por arquivos que tiverem o tamanho [num]. [num] pode ser antecedido de "+" ou "-" para especificar um arquivo maior ou menor que [num]. A opção `-size` pode ser seguida de:

- b - Especifica o tamanho em blocos de 512 bytes. É o padrão caso [num] não seja acompanhado de nenhuma letra.
- c - Especifica o tamanho em bytes.
- k - Especifica o tamanho em Kbytes.

-type [tipo] Procura por arquivos do [tipo] especificado. Os seguintes tipos são aceitos:

- b - bloco
- c - caracter
- d - diretório
- p - pipe
- f - arquivo regular
- l - link simbólico
- s - sockete

A maior parte dos argumentos numéricos podem ser precedidos por "+" ou "-". Para detalhes sobre outras opções e argumentos, consulte a página de manual.

Exemplo:

- `find / -name grep` - Procura no diretório raiz e sub-diretórios um arquivo/diretório chamado `grep`.
- `find / -name grep -maxdepth 3` - Procura no diretório raiz e sub-diretórios até o 3o. nível, um arquivo/diretório chamado `grep`.
- `find . -size +1000k` - Procura no diretório atual e sub-diretórios um arquivo com tamanho maior que 1000 kbytes (1Mbyte).

free

Mostra detalhes sobre a utilização da memória RAM do sistema.

`free [opções]`

Onde:

opções

- b Mostra o resultado em bytes.
 - k Mostra o resultado em Kbytes.
 - m Mostra o resultado em Mbytes.
 - o Oculta a linha de buffers.
 - t Mostra uma linha contendo o total.
 - s [num] Mostra a utilização da memória a cada [num] segundos.
- O free é uma interface ao arquivo /proc/meminfo.

head

Mostra as linhas iniciais de um arquivo texto.

head [opções]

Onde: Capítulo 8. Comandos Diversos 74

-c [numero] Mostra o [numero] de bytes do início do arquivo.

-n [numero] Mostra o [numero] de linhas do início do arquivo. Caso não for especificado, o head mostra as 10 primeiras linhas.

Exemplos: head teste.txt, head -n 20 teste.txt.

uptime

Mostra o tempo de execução do sistema desde que o computador foi ligado.

Uptime

dmesg

Mostra as mensagens de inicialização do kernel. São mostradas as mensagens da última inicialização do sistema.

dmesg | less

mesg

Permite ou não o recebimento de requisições de talk de outros usuários.

mesg [y/n]

Onde: y permite que você receba "talks" de outros usuários.

Digite mesg para saber se você pode ou não receber "talks" de outros usuários. Caso a resposta seja "n" você poderá enviar um talk para alguém mas o seu sistema se recusará em receber talks de outras pessoas.

É interessante colocar o comando mesg y em seu arquivo de inicialização .bash_profile para permitir o recebimento de "talks" toda vez que entrar no sistema.

echo

Mostra mensagens. Este comando é útil na construção de scripts para mostrar mensagens na tela para o usuário acompanhar sua execução.

`echo` [mensagem]

A opção `-n` pode ser usada para que não ocorra o salto de linha após a mensagem ser mostrada.

su

Permite o usuário mudar sua identidade para outro usuário sem fazer o logout. Útil para executar um programa ou comando como root sem ter que abandonar a seção atual.

`su` [usuário] [-c comando]

Onde: usuário é o nome do usuário que deseja usar para acessar o sistema. Se não digitado, é assumido o usuário root. Caso seja especificado `-c` comando, executa o comando sob o usuário especificado.

Será pedida a senha do superusuário para autenticação. Digite `exit` quando desejar retornar a identificação de usuário anterior.

uname

Retorna o nome e versão do kernel atual.

`uname`

reboot

Reinicia o computador.

shutdown

Desliga/reinicia o computador imediatamente ou após determinado tempo (programável) de forma segura. Todos os usuários do sistema são avisados que o computador será desligado. Este comando somente pode ser executado pelo usuário root ou quando é usada a opção `-a` pelos usuários cadastrados no arquivo `/etc/shutdown.allow` que estejam logados no console virtual do sistema.

`shutdown` [opções] [hora] [mensagem]

hora

Momento que o computador será desligado. Você pode usar HH:MM para definir a hora e minuto, MM para definir minutos, +SS para definir após quantos segundos, ou `now` para imediatamente (equivalente a +0). O `shutdown` criará o arquivo `/etc/nologin` para não permitir que novos usuários façam login no sistema (com exceção do root). Este arquivo é removido caso a execução do `shutdown` seja cancelada (opção `-c`) ou após o sistema ser reiniciado.

mensagem

Mensagem que será mostrada a todos os usuários alertando sobre o reinício/desligamento do sistema.

opções

`-h` Inicia o processo para desligamento do computador.

`-r` Reinicia o sistema

-c Cancela a execução do shutdown. Você pode acrescentar uma mensagem avisando aos usuários sobre o fato.

Comandos de rede

Este capítulo traz alguns comandos úteis para uso em rede e ambientes multiusuário.

who

Mostra quem está atualmente conectado no computador. Este comando lista os nomes de usuários que estão conectados em seu computador, o terminal e data da conexão.

`who [opções]`

onde: opções

-H, -heading Mostra o cabeçalho das colunas.

-b, -boot Mostra o horário do último boot do sistema.

-d, -dead Mostra processos mortos no sistema.

`telnet`

Permite acesso a um computador remoto. É mostrada uma tela de acesso correspondente ao computador local onde deve ser feita a autenticação do usuário para entrar no sistema. Muito útil, mas deve ser tomado cuidados ao disponibilizar este serviço para evitar riscos de segurança e usado o ssh sempre que possível por ser um protocolo criptografado e com recursos avançados de segurança.

`telnet [opções] [ip/dns] [porta]`

onde: ip/dns Endereço IP do computador de destino ou nome DNS.

porta Porta onde será feita a conexão. Por padrão, a conexão é feita na porta 23.

opções -8 Requisita uma operação binária de 8 bits. Isto força a operação em modo binário para envio e recebimento. Por padrão, telnet não usa 8 bits.

-a Tenta um login automático, enviando o nome do usuário lido da variável de ambiente USER.

-d Ativa o modo de debug.

-r Ativa a emulação de rlogin.

-l [usuário] Faz a conexão usando [usuário] como nome de usuário.

Exemplo: `telnet 192.168.1.1`, `telnet 192.168.1.1 23`.

Comandos para manipulação de contas

Este capítulo traz comandos usados para manipulação de conta de usuários e grupos em sistemas GNU/Linux. Entre os assuntos descritos aqui estão adicionar usuários ao sistema, adicionar grupos, incluir usuários em grupos existentes, etc.

adduser

Adiciona um usuário ou grupo no sistema. Por padrão, quando um novo usuário é adicionado, é criado um grupo com o mesmo nome do usuário. Opcionalmente o `adduser` também pode ser usado para adicionar um usuário a um. Será criado um diretório `home` com o nome do usuário (a não ser que o novo usuário criado seja um usuário do sistema) e este receberá uma identificação. A identificação do usuário (UID) escolhida será a primeira disponível no sistema especificada de acordo com a faixa de UIDS de usuários permitidas no arquivo de configuração `/etc/adduser.conf`.

Este é o arquivo que contém os padrões para a criação de novos usuários no sistema.

`adduser [opções] [usuário/grupo]`

Onde: usuário/grupo Nome do novo usuário que será adicionado ao sistema.

Opções

-`disable-passwd` Não executa o programa `passwd` para escolher a senha e somente permite o uso da conta após o usuário escolher uma senha.

-`force-badname` Desativa a checagem de senhas ruins durante a adição do novo usuário. Por padrão o `adduser` checa se a senha pode ser facilmente adivinhada.

-`group` Cria um novo grupo ao invés de um novo usuário. A criação de grupos também pode ser feita pelo comando `addgroup`.

-`system` Cria um usuário de sistema ao invés de um usuário normal.

Os dados do usuário são colocados no arquivo `/etc/passwd` após sua criação e os dados do grupo são colocados no arquivo `/etc/group`.

OBSERVAÇÃO: Caso esteja usando senhas ocultas (`shadow passwords`), as senhas dos usuários serão colocadas no arquivo `/etc/shadow` e as senhas dos grupos no arquivo `/etc`

`/gshadow`. Isto aumenta mais a segurança do sistema porque somente o usuário `root` pode ter acesso a estes arquivos, ao contrário do arquivo `/etc/passwd` que possui os dados de usuários e devem ser lidos por todos.

addgroup

Adiciona um novo grupo de usuários no sistema. As opções usadas são as mesmas do `'adduser'` on the preceding page.

`addgroup [usuário/grupo] [opções]`

passwd

Modifica a parâmetros e senha de usuário. Um usuário somente pode alterar a senha de sua conta, mas o superusuário (`root`) pode alterar a senha de qualquer conta de usuário, inclusive a data de validade da conta, etc. Os donos de grupos também podem alterar a senha do grupo com este comando.

Os dados da conta do usuário como nome, endereço, telefone, também podem ser alterados com este comando.

`passwd [usuário] [opções]`

Onde:

usuário Nome do usuário que terá sua senha alterada.

opções

-e Força a expiração de senha para a conta especificada.

`gpasswd`

Modifica parametros e senha de grupo. Um usuário somente pode alterar a senha de seu grupo, mas o superusuário (root) pode alterar a senha de qualquer grupo de usuário, inclusive definir o administrador do grupo.

`gpasswd [opções] [usuario] [grupo]`

Onde:

usuário Nome do usuário/grupo que terá sua senha alterada.

opções

-r usuario grupo Remove a senha de grupo.

-R usuario grupo Desativa o acesso do grupo usando o comando `newgrp`.

Quando o grupo não possui senha, somente quem faz parte do grupo pode utilizar o comando `new-grp`.

Você deve ser o dono da conta para poder modificar a senhas. O usuário root pode modificar/ apagar a senha de qualquer usuário.

Exemplo: `gpasswd grupo`, `gpasswd -a gleison grupo`.

`Newgrp` Altera a identificação de grupo do usuário. Para retornar a identificação anterior, digite `exit` e tecla `Enter`. Para executar um comando com outra identificação de grupo de usuário, use o comando `'sg' on the next page`.

`newgrp - [grupo]`

Onde: - Se usado, inicia um novo ambiente após o uso do comando `newgrp` (semelhante a um novo login no sistema), caso contrário, o ambiente atual do usuário é mantido.

grupo Nome do grupo ou número do grupo que será incluído.

Quando este comando é usado, é pedida a senha do grupo que deseja acessar. Caso a senha do grupo esteja incorreta ou não exista senha definida, a execução do comando é negada.

Instalando / atualizando e removendo pacotes

APT GET

O `apt-get` é um recurso desenvolvido originalmente para a distribuição Debian que permite a instalação e a atualização de pacotes (programas, bibliotecas de funções, etc) no Linux de maneira fácil e precisa. Vale frisar que o `apt-get` deve ser utilizado através de um usuário com privilégios de administrador.

Em qualquer sistema operacional há a necessidade de se instalar programas para que o uso do computador faça sentido. O problema é que, muitas vezes, o processo de instalação é trabalhoso, forçando o usuário a procurar arquivos indispensáveis ao software (dependências) ou a editar scripts, por exemplo.

Para poder lidar com esse e com problemas oriundos, desenvolvedores da distribuição Debian Linux criaram o APT (Advanced Packaging Tool), uma ferramenta para instalar e atualizar pacotes (e suas dependências) de maneira rápida e prática. Não demorou para que o APT fosse utilizado em outras distribuições, sendo padrão no Debian e noUbuntu (já que este é baseado no Debian).

Como utilizar o `apt-get`

Atualizando a lista de pacotes

Quando o `apt-get` é utilizado no Linux, o comando consulta um arquivo denominado `sources.list`, geralmente disponível no diretório `/etc/apt/`. Esse arquivo informa onde estão os locais (repositórios) nos quais ele obterá os pacotes requisitados pelo usuário.

Pelo arquivo `/etc/apt/sources.list` o APT sabe onde procurar os pacotes solicitados. O problema é que ele precisa conhecer o que está instalado no sistema operacional para saber exatamente o que procurar. Se não tiver essas informações, o APT não saberá que um determinado pacote essencial a um programa não está instalado e não providenciará sua instalação, por exemplo. Para lidar com essa situação, o APT utiliza um banco de dados próprio e o compara aos repositórios registrados nos `sources.list` para saber quando um pacote está atualizado ou quando há uma nova versão de um determinado aplicativo. Para fazer o APT executar essa tarefa, basta digitar em um terminal o seguinte comando:

```
apt-get update
```

Antes de instalar um programa, é necessário executar o comando mostrado no tópico anterior para o APT saber o que está e o que não está instalado no computador. É por isso que esse comando foi explicado primeiro.

Depois de ter executado o `apt-get update`, basta digitar o seguinte comando para instalar os pacotes desejados:

```
apt-get install nomedoprograma
```

Por exemplo, suponha que você queira instalar o cliente de e-mail Mozilla Thunderbird. Para isso, basta dar o comando:

```
apt-get install mozilla-thunderbird
```

Quando esse comando é executado, o APT procura nos repositórios os pacotes correspondentes e os instala imediatamente após baixá-los.

Caso o programa necessite de um outro pacote, o APT o procurará e perguntará se você deseja instalá-lo, no que você deverá responder com Y (sim) ou N (não), como mostra a situação abaixo:

```
0 packages upgraded, 6 newly installed, 0 to remove and 2 not upgraded. Need to get 6329kB of archives.
After unpacking 14.1MB will be used.
```

```
Do you want to continue? [Y/n]
```

Removendo pacotes com o `apt-get`

Se você não quer mais utilizar um determinado programa, pode desinstalá-lo através do comando `apt-get`. Para isso, basta digitar:

```
apt-get remove nomedoprograma
```

Por exemplo, suponha que você queira desinstalar o Mozilla Thunderbird. O comando ficará então da seguinte forma:

```
apt-get remove mozilla-thunderbird
```

É importante destacar que o comando acima remove o programa, mas mantém os arquivos de configuração, caso estes existam. Para uma desinstalação total do programa, incluindo arquivos extras, deve-se usar o comando:

```
apt-get --purge remove nomedoprograma
```

Atualizando pacotes com o `apt-get`

Versões novas de programas são lançadas periodicamente por conterem correções de segurança, disponibilizarem novos recursos ou por executarem de forma melhor. Com o comando `apt-get` é possível obter e instalar versões mais novas de pacotes. Para isso basta digitar o comando:

```
apt-get -u upgrade
```

O parâmetro `-u` serve para o `apt-get` mostrar os pacotes que estão sendo atualizados, portanto, seu uso é opcional.

Um detalhe importante: na atualização de pacotes, também é importante executar antes o comando `apt-get update`.

CHMOD

altera permissões

Lembre que somente quem tem permissão no arquivo ou diretório poderá alterar as permissões, quando ocorrer erros, verifique seus direitos sobre o arquivo ou diretório com o comando `ls -l` ou faça com poderes de root.

Vamos aprender de modo simples a usar as permissões que muitas vezes é bastante complicado.

Tendo em mente que o conceito de binário significa 0 = desligado e 1 = ligado, vejamos como fica na tabela abaixo:

```
rwX
0 - 000
1 - 001
2 - 010
3 - 011
4 - 100
5 - 101
6 - 110
7 - 111
```

Onde "rwX" são as permissões de um arquivo, ou seja:

r=read (leitura)

w=write (gravação, alteração, deleção)

x=execute (execução)

Feito isso, sabemos que um arquivo ou diretório possui 3 modos de permissão. Uma permissão para o DONO do arquivo ou seja, quem o criou, uma outra permissão para o GRUPO do usuário dono do arquivo, e outra permissão para QUALQUER outro usuário ou grupo.

Sabendo disso, temos a tabela final abaixo:

```
dono grupo outros
rwx rwx rwx
```

```
0 - 000 000 000
1 - 001 001 001
2 - 010 010 010
3 - 011 011 011
4 - 100 100 100
5 - 101 101 101
6 - 110 110 110
7 - 111 111 111
```

Exemplo: Temos um diretório qualquer chamado mp3 que eu quero permissão total para o dono [eu], permissão de apenas leitura para o meu grupo e nenhuma permissão para o resto do mundo. O comando para executar essa ação é o CHMOD (em minúsculo mesmo).

```
$ chmod 740 /mp3
```

(acompanhem na tabela acima, que o 7=rwx, 4=r e 0=nada)

Entendendo o modo gráfico no Linux

X Window System, X, GUI, Window manager, gdm, kdm ?

Para decidir entre uma interface de prompt de comando e uma interface gráfica com o usuário conhecida como GUI, você precisa primeiro considerar DESEMPENHO.

Se você planeja apenas compilar programas ou realizar tarefas de intenso processamento que tendam tornar a máquina com um desempenho abaixo do esperado, rodar um GUI irá apenas aumentar o problema da falta de desempenho, principalmente, se você não tiver um computador rápido. Outro fator importante na escolha entre essas duas interfaces é a legibilidade.

O sistema Xfree86 (também chamado de X ou X Window System) foi projetado para dar ao Linux e aos sistemas Unix uma interface gráfica amigável.

Em geral, o X é um conjunto de aplicativos que funcionam juntos para fornecer uma interface gráfica ao usuário. E isso é importante desde que se leve em consideração as pessoas que migram de sistemas operacionais como Windows e MacOS.

Diferente de outros sistemas operacionais como Windows e MacOS, O X Window System não é parte integrada ao Linux. O X Window System é uma aplicação que roda no topo da base do sistema operacional. Isso permite você executar a GUI quando quiser, mas para isso você terá que ter uma quantidade razoável de memória como dito anteriormente.

Várias pessoas confundem entre cliente e servidor quando o assunto é X Window System.

A maioria das pessoas considera suas estações de trabalho como cliente, mas no caso do X, suas estações são na verdade servidores. Os programas são clientes e o ambiente gráfico, servidor. Logo, um cliente faz solicitações ao servidor, que se encarrega de respondê-las.

Como qualquer arquitetura cliente-servidor, o X Window System permite que o servidor esteja em uma máquina e o cliente, em outra.

Os dois maiores concorrentes GUI são o KDE e Gnome que são gerenciadores de janelas (window manager).

Afinal, o que é um gerenciador de janelas???

Gerenciador de janelas consiste em um programa que roda sob o X Window System e é responsável pelas funções de ajuste de tamanho de janela, minimização, maximização, barras de título, bordas e outras decorações para as janelas das aplicações, ícones.

É o gerenciador de janelas que determinará a aparência do seu sistema X, portanto a escolha de um vai do gosto do freguês. A utilização de um ambiente gráfico aumenta a produtividade do usuário, no qual o mouse é indispensável hoje em dia em ambientes desktop.

Como é a relação do X com as outras camadas no sistema:

Hardware -> Módulos -> Kernel -> Shell -> X (servidor gráfico) -> Gerenciador de janelas (KDE ou Gnome etc)

Hoje em dia, as distribuições Linux estão migrando de Xfree86 para Xorg. O Xfree86 é um projeto antigo e foi o grupo que portou o X para o Linux. O Xorg surgiu através reclamações na demora de incluir correções e atualizações de drivers no XFree.

O Xorg utilizou como base o Xfree, a partir daí novos drivers e atualizações foram incluídas.

Além de tudo isso, existem também os gerenciadores de login. Antigamente foi muito usado o famoso "startx" para abrir o modo gráfico, em algumas vezes ainda é usado. Hoje em dia é comum usar um gerenciador de login como o gdm (Gnome) ou o kdm (Kde).

A função do gerenciador de login é carregar o X, mostrar uma tela para fazer autenticação de usuários e carregar um gerenciador de janelas escolhido.

Um pouco de história

O X foi desenvolvido originalmente no MIT (Massachusetts Institute of Technology) e foi lançado em 1984 e é um derivado de um projeto chamado Athena, no qual um grupo formado por MIT, IBM e Digital Equipment projetaram um ambiente gráfico educacional.

Em 1987, o MIT lançou a primeira versão do X que nós conhecemos hoje como X11. A versão do X que é usado no Linux é Xfree86, esse nome surgiu pois seu primeiro pacote foi testado em um x86 (processador Intel), mas atualmente roda em várias plataformas e sistemas.

Como usar a Central de programas do Ubuntu

Neste tópico aprendemos a utilizar a central de programas do Ubuntu, vejam a facilidade de instalar ou remover programas pela central no passo a passo abaixo.

A Central de programas do Ubuntu é a loja oficial de aplicativos do sistema, nela você pode descobrir, comprar e instalar aplicativos para seu Ubuntu. Acompanhe esse tutorial do TechTudo e aprenda como usá-la.

Como instalar um programa

Passo 1. Na barra de atalho do Ubuntu, clique na opção "Central de programas do Ubuntu";



Figura 16 - Central de Programas Ubuntu

Passo 2. Dentro do programa, digite o nome do aplicativo dentro da caixa de pesquisa que fica no canto superior direito da tela;

Passo 3. Ao encontrar o programa, clique nele e aparecerá dois botões: "Mais informações" e "Instalar" (ou "Comprar"). Clique em "Mais informações" para ver mais detalhes sobre o programa;

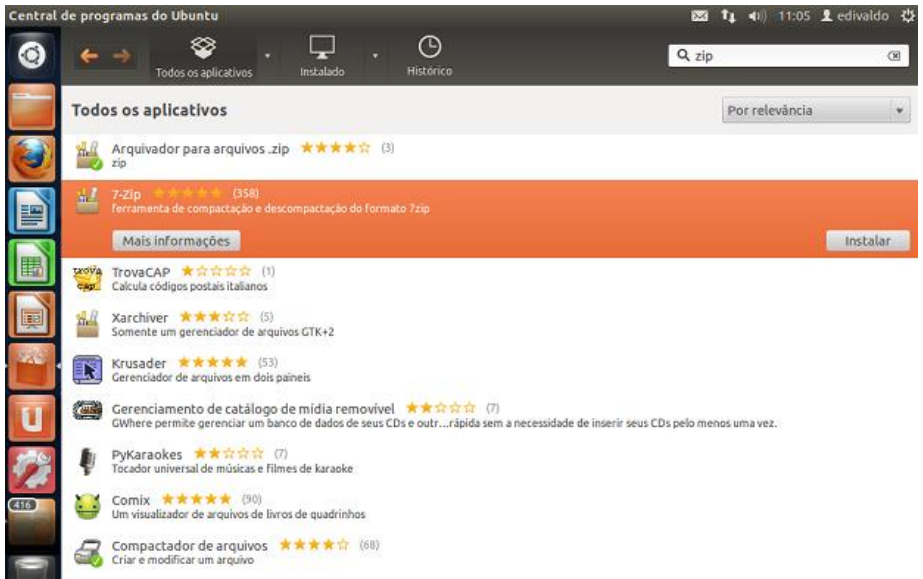


Figura 17 - Centra de Programas Ubuntu

Passo 4. Na próxima janela, veja os detalhes do programa como descrição, preço e dependências. Depois clique no botão "Instalar" (ou comprar, se for um programa pago);

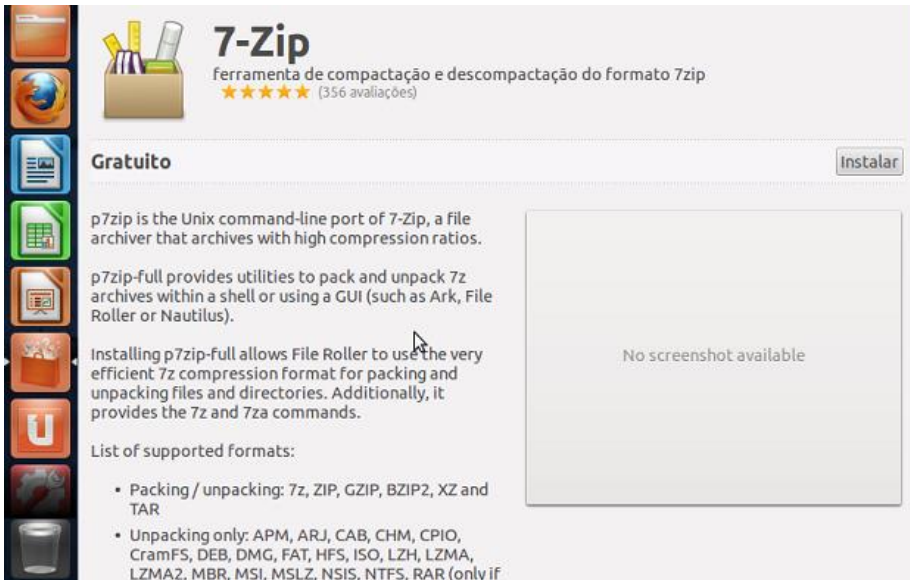


Figura 18 - Central de Programas Ubuntu

Passo 5. Se for um programa pago, serão pedidas as informações para o pagamento através do serviço "Ubuntu Single sign on", senão será solicitada a senha de administrador para continuar a instalação, digite-a e confirme clicando em "Autenticar". Terminado o processo o processo, o programa já estará disponível para uso. Feche a Central de programas do Ubuntu.



Figura 19 - Central de Programas Ubuntu

Como remover um programa

Passo 1. Na barra de atalho do Ubuntu, clique na opção "Central de programas do Ubuntu";

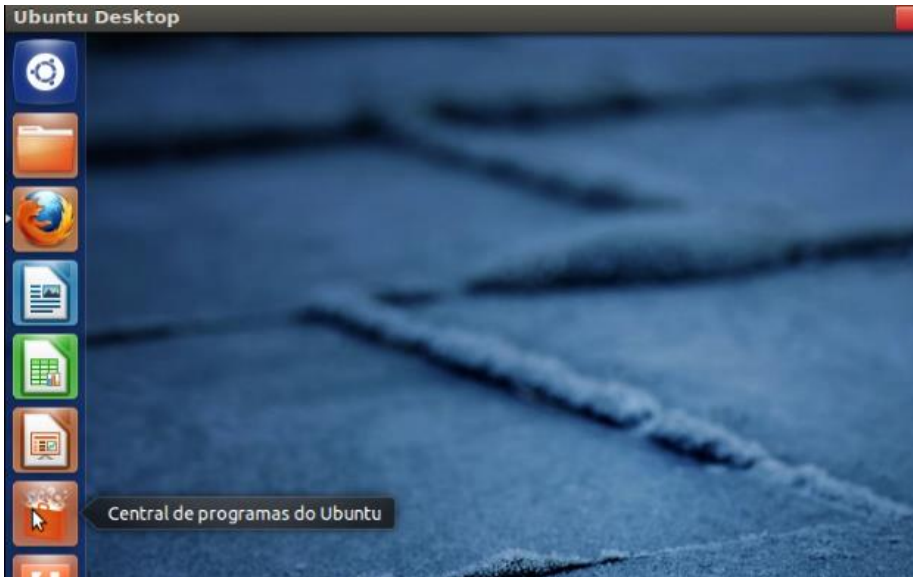


Figura 20 - Central de Programas Ubuntu

Passo 2. Dentro do programa, digite o nome do aplicativo dentro da caixa de pesquisa que fica no canto superior direito da tela;

Passo 3. Ao encontrar o programa, clique nele e aparecerá dois botões: "Mais informações" e "Remover". Clique em "Mais informações" para ver mais detalhes sobre o programa;

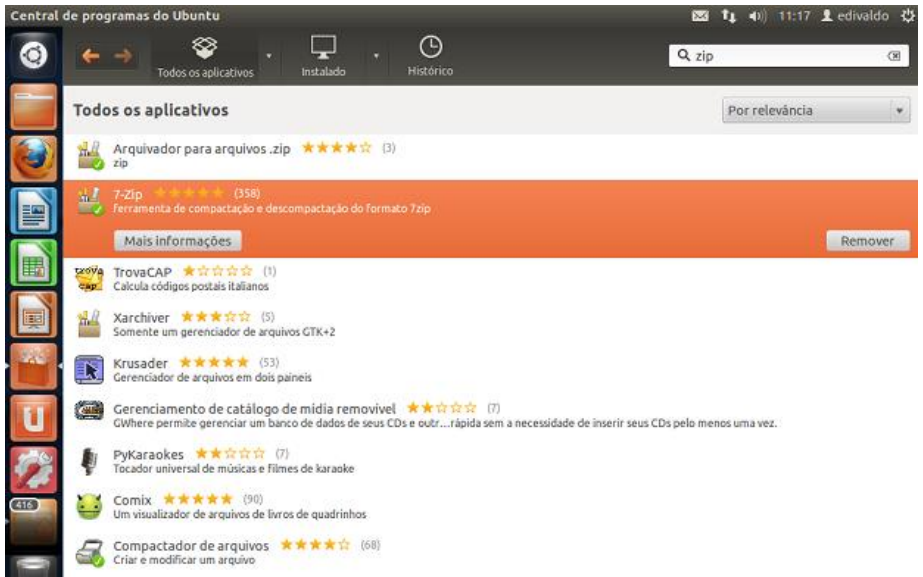


Figura 21 - Centra de Programas Ubuntu

Passo 4. Na próxima janela, veja os detalhes do programa como, descrição, preço e dependências. Se tiver certeza que realmente deseja desinstalá-lo, clique no botão "Remover";

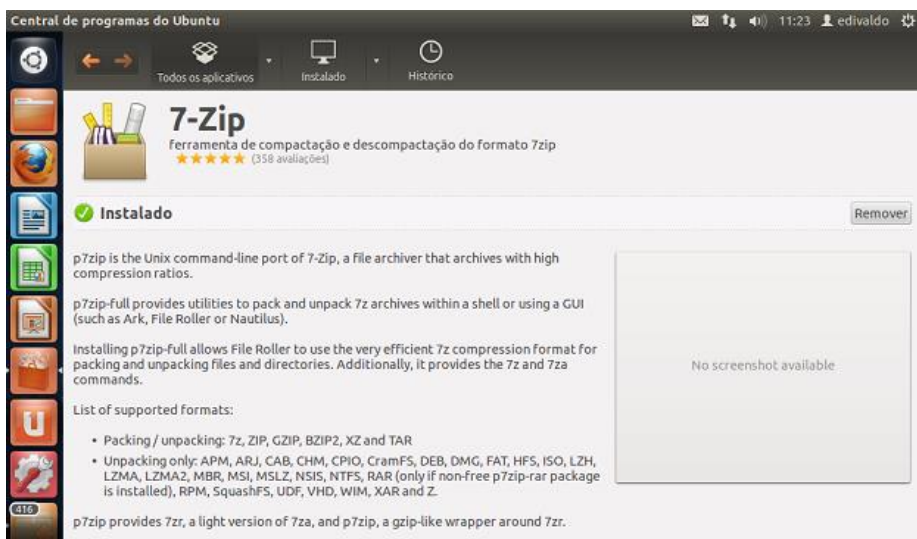


Figura 22 - Centra de Programas Ubuntu

Passo 5. Será solicitada a senha de administrador para continuar a instalação, digite-a e clique no botão "Autenticar". Aguarde terminar o processo;



Figura 23 - Central de Programas Ubuntu

Com é possível ver, com a Central de programas do Ubuntu, a instalação e remoção de programas no Ubuntu fica muito fácil.

Introdução ao Shell Script

Criação Shell Script

Utilizando o interpretador de comandos bash ou shell, podemos criar códigos ou scripts no sistema Linux. Esses códigos criados facilitam muito a vida do programador ao economizar tempo, pois permitem automatizar tarefas de rotina e também tornam mais simples ações como: abrir programas e esvaziar diretórios temporários e lixeira, entre outras tarefas.

Em primeiro lugar precisaremos de um arquivo para escrever o nosso script. Podemos fazer isso via terminal ou via modo gráfico, sendo que, no último caso, basta apenas clicar com o botão direito do mouse em um diretório desejado e escolher 'criar novo arquivo de texto' ou 'criar novo documento'.

Para criar um arquivo via terminal, basta abrir o mesmo e digitar:

```
vi exemplo1.sh
```

Ou também:

```
touch exemplo1.sh
```

O comando `vi` cria e abre um arquivo para leitura/escrita no terminal, enquanto o comando `touch` cria um arquivo, mas não o abre. Posteriormente é possível abri-lo com um editor de sua preferência.

Permissão de arquivo

Para começar a editar o arquivo, é preciso conceder a permissão de escrita a ele. Se você digitou o comando `vi`, pressione `ESC` para que apareça o caractere `“:”`. Então digite:

```
!chmod 777 %
```

Caso tenha optado pelo comando `touch`, escreva:

```
chmod 777 exemplo1.sh
```

Vamos entender o que foi feito: o comando `chmod` é usado para conceder permissões em diretórios e arquivos, enquanto o valor `777` permite que o usuário total liberdade para editar o arquivo. O caractere `(!)` obriga o comando `vi` a executar o `chmod`, enquanto o símbolo `(%)` refere-se ao arquivo atual.

Editar E Executar O Arquivo

Depois da permissão do arquivo, iniciaremos a parte de edição do arquivo e usaremos o comando `vi` para essa tarefa. Abra o arquivo de digite:

```
vi exemplo1.sh
```

É necessário digitar `“i”` para colocar o `vi` no modo inserção.

Nesse ponto, você precisa definir qual Shell será usado — vamos escolher o `bash`. Porém, você pode escolher outro interpretador de comandos, como `sh`, `ksh` ou `csh`. Para usar o `bash`, digite:

```
#!/bin/bash
```

Inserir Comentários

Esse item é fundamental na programação do sistema, porque, ao inserir comentários que expliquem a função do código e que ações estão sendo feitas no programa, o programador não perde a linha de execuções.

Inicie com o caractere `“#”` para inserir o comentário, como mostra o exemplo abaixo:

```
#!/bin/bash
```

```
#comentários
```

Uso Das Variáveis

Deixar de declarar as variáveis pode prejudicar o funcionamento dos comandos, visto que são elas que armazenam informações importantes que serão utilizadas pelos códigos. Por isso essa ação é tão importante na linguagem da programação.

Para declarar a variável, siga o exemplo:

```
nome_da_variavel=valor
```

Vale lembrar que o valor pode ser strings ou números. Para utilizar o valor, devemos colocar o símbolo (\$) na frente do nome que será inserido.

Entrada De Dados Do Usuário

Se o script pedir ao usuário o fornecimento de dados para a entrada do processamento, o programador precisará digitar um comando para que o ele leia os dados escritos. Nesse caso, digite:

```
read nome_da_variavel_para_armazenar_o_valor_a_ser_lido
```

Validação da execução dos comandos

Um script precisa validar suas tarefas para cumpri-las com eficiência. Para isso, acompanhe sempre os logs, para que possa perceber erros ou melhorias. Existem recursos de monitoramento que podem ser utilizados caso você deseje.

Comando De Seleção

Um comando simples que permite a execução de uma tarefa baseada na ação de um usuário que esteja usando o seu sistema é o condicional. Faça o seguinte:

```
if [ CONDICAO ] ;
```

```
then
```

AÇÕES

```
fi
```

É importante lembrar que para cada "if" aberto, deve haver um "fi" que feche a sequência — a mesma regra serve para os colchetes.

Leia também ... 5 Maneiras arrasadoras de blogar utilizando Linux

Funções

Para organizar, separar e estruturar um algoritmo, é necessário o uso de funções — não importa a linguagem de programação utilizada. Veja o que você pode fazer:

```
nome-funcao()
```

```
{
```

AÇÕES

```
}
```

Inserção De Argumentos

O shell script recebe dados fornecidos por outro programa ou por um usuário com a finalidade de produzir saídas — esses dados são os argumentos. Veja alguns dos nomes que qualificam os argumentos:

`$#` — total de argumentos que foram passados

`$*` — retorno aos argumentos

`$0` — nome do script executado

Referências:

<https://www.devmedia.com.br/introducao-ao-shell-script-no-linux/25778>

<http://e-tinet.com/linux/programacao-shell-script/>

Curiosidades

É livre e desenvolvido voluntariamente por programadores experientes, hackers, e contribuidores espalhados ao redor do mundo que tem como objetivo a contribuição para a melhoria e crescimento deste sistema operacional. Muitos deles estavam cansados do excesso de propaganda (Marketing) e baixa qualidade de sistemas comerciais existentes;

- Também recebe apoio de grandes empresas como IBM, Sun, HP, etc. para seu desenvolvimento;
- Convivem sem nenhum tipo de conflito com outros sistemas operacionais (com o DOS Windows, OS/2)

no mesmo computador;

- Multitarefa real;
- Multiusuário;
- Suporte a nomes extensos de arquivos e diretórios (255 caracteres);
- Conectividade com outros tipos de plataformas como Apple, Sun, Macintosh, Sparc, Alpha PowerPc,

ARM, Unix, Windows, DOS;

- Utiliza permissões de acesso a arquivos, diretórios e programas em execução na memória RAM;
- Proteção entre processos executados na memória RAM;
- Suporte a mais de 63 terminais virtuais (consoles);
- Modularização - O Linux somente carrega para a memória o que é usado durante o processamento, liberando totalmente a memória assim que o programa/dispositivo é finalizado.

• Devido a modularização, os drivers dos periféricos e recursos do sistema podem ser carregados e removidos completamente da memória RAM a qualquer momento. Os drivers (módulos) ocupam pouco espaço quando carregados na memória RAM (cerca de 6Kb para a Placa de rede NE 2000, por exemplo)

• Suporte nativo a rede e tecnologias avançadas como: balanceamento de carga, ips alias, failover, vlans, bridge, trunking, OSPF, BGP.

• Não há a necessidade de se reiniciar o sistema após a modificar a configuração de qualquer periférico ou parâmetros de rede. Somente é necessário reiniciar o sistema no caso de uma instalação interna de um novo periférico, falha em algum hardware (queima do processador, placa mãe, etc.).

- Não precisa de um processador potente para funcionar. O sistema roda bem em computadores 386Sx 25 com 4MB de memória RAM (sem rodar o sistema gráfico X, que é recomendado 32MB de RAM). Já pensou no seu desempenho em um Pentium, Xeon, ou Athlon?).
- Suporte nativo a múltiplas CPUs, assim processadores como Dual Core, Core Duo, Athlon Duo, Quad Core tem seu poder de processamento integralmente aproveitado, tanto em 32 ou 64 bits.
- Suporte nativo a dispositivos SATA, PATA, Fiber Channel
- Suporte nativo a virtualização, onde o Linux se destaca como plataforma preferida para execução de múltiplos sistemas operacionais com performance e segurança.
 - O crescimento e novas versões do sistema não provocam lentidão, pelo contrário, a cada nova versão os desenvolvedores procuram buscar maior compatibilidade, acrescentar recursos úteis e melhor desempenho do sistema (como o que aconteceu na passagem do kernel 2.0.x para 2.2.x, da 2.2.x para a 2.4.x e da 2.4.x para a 2.6.x).
 - O GNU/Linux é distribuído livremente e licenciado de acordo com os termos da GPL.
 - Acessa corretamente discos formatados pelo DOS, Windows, Novell, OS/2, NTFS, SunOS, Amiga, Atari, Mac, etc.
 - O LINUX NÃO É VULNERÁVEL A VÍRUS! Devido a separação de privilégios entre processos e respeitadas as recomendações padrão de política de segurança e uso de contas privilegiadas (como a de root, como veremos adiante), programas como vírus tornam-se inúteis pois tem sua ação limitada pelas restrições de acesso do sistema de arquivos e execução. Qualquer programa (nocivo ou não) poderá alterar partes do sistema que possui permissões (será abordado como alterar permissões e tornar seu sistema mais restrito no decorrer do guia). Frequentemente são criados exploits que tentam se aproveitar de falhas existentes em sistemas desatualizados e usa-las para causar danos. Erroneamente este tipo de ataque é classificado como vírus por pessoas mal informadas e são resolvidas com sistemas bem mantidos. Em geral, usando uma boa distribuição que tenha um eficiente sistema de atualização e bem configurado, você terá 99.9% de sua tranquilidade.
 - Rede TCP/IP mais rápida que no Windows e tem sua pilha constantemente melhorada. O GNU/Linux tem suporte nativo a redes TCP/IP e não depende de uma camada intermediária como o WinSock. Em acessos via modem a Internet, a velocidade de transmissão é 10% maior.
 - Executa outros sistemas operacionais como Windows, MacOS, DOS ou outro sistema Linux através de consagrados sistemas de virtualização como Xen, vmware, ou emulação como o DOSEMU, QEMU, WINE.
 - Suporte completo e nativo a diversos dispositivos de comunicação via infravermelho, Bluetooth, Firewire, USB. Basta conectar e o seu dispositivo é automaticamente reconhecido. Raramente são necessários drivers externos, exceto no caso de dispositivos muito novos que não tenham o suporte ainda adicionado no sistema.
 - Suporte a rede via rádio amador.
 - Suporte a dispositivos Plug-and-Play.
 - Suporte nativo a pendrivers, dispositivos de armazenamento e cartões de memória.
 - Suporte nativo a dispositivos I2C
 - Integração com gerenciamento de energia ACPI e APM
 - Dispositivos de rede Wireless. Tanto com criptografia WEB e WPA PSK
 - Vários tipos de firewalls avançados de alta qualidade na detecção de tráfego indesejável dando ao administrador uma excelente ferramenta de proteção e controle de sua rede.
 - Roteamento estático e dinâmico de pacotes.
 - Ponte entre Redes, proxy arp.

- Proxy Tradicional e Transparente.
- Possui recursos para atender a mais de um endereço IP na mesma placa de rede, sendo muito útil para situações de manutenção em servidores de redes ou para a emulação de “múltiplos computadores”. O servidor WEB e FTP podem estar localizados no mesmo computador, mas o usuário que se conecta tem a impressão que a rede possui servidores diferentes.
- Os sistemas de arquivos usados pelo GNU/Linux (Ext2, Ext3, reiserfs, xfs, jfs) organiza os arquivos de forma inteligente evitando a fragmentação e fazendo-o um poderoso sistema para aplicações multi-usuárias exigentes e gravações intensivas.
- Permite a montagem de um servidor de publicação Web, E-mail, News, etc. com um baixo custo e alta performance. O melhor servidor Web do mercado, o Apache, é distribuído gratuitamente junto com a maioria das distribuições Linux. O mesmo acontece com o Sendmail.
- Por ser um sistema operacional de código aberto, você pode ver o que o código fonte (instruções digitadas pelo programador) faz e adapta-lo as suas necessidades ou de sua empresa. Esta característica é uma segurança a mais para empresas sérias e outros que não querem ter seus dados roubados (você não sabe o que um sistema sem código fonte faz na realidade enquanto está processando o programa).
- Suporte a diversos dispositivos e periféricos disponíveis no mercado, tanto os novos como obsoletos.
- Pode ser executado em 16 arquiteturas diferentes (Intel, Macintosh, Alpha, Arm, etc.) e diversas outras sub-arquiteturas.
- Empresas especializadas e consultores especializados no suporte ao sistema espalhados por todo o mundo.
- Entre muitas outras características que você descobrirá durante o uso do sistema (além de poder criar outras, caso seja um administrador avançado ou desenvolvedor).

TODOS OS ITENS DESCRITOS ACIMA SÃO VERDADEIROS E TESTADOS PARA QUE TIVESSE PLENA CERTEZA DE SEU FUNCIONAMENTO.

Referências Bibliográficas

Delfino, P. Programação Shell Script: O básico para iniciantes e criar scripts no Bash Linux. Disponível em <<http://e-tinet.com/linux/programacao-shell-script/>> , acessado em 10 de Out. de 2018.

Lousada, F. Introdução ao Shell Script no Linux. Disponível em <<https://www.devmedia.com.br/introducao-ao-shell-script-no-linux/25778>>, acessado em 07 de Set. de 2018

Bernardi, R. Linux: pós e contras pra que usa o sistema. Disponível em <<http://hardware.rbtech.info/linux-pros-contras-sistema/>>, acessado em 09 de Out. de 2018

https://getfedora.org/pt_BR/, acessado em 11 de Out. de 2018

Higa, P. Os pós e contras de 6 distribuições de Linux. Disponível em <<https://www.guiadopc.com.br/artigos/344/os-pros-e-contras-de-6-distribuicoes-linux.html>>. Acessado em 09 de Out. de 2018.

Castro, S. de. Guia de pós-instalação do Debian 9 Stretch. Disponível em <<https://www.blogopcaolinux.com.br/2017/06/Guia-de-pos-instalacao-do-Debian-9-Stretch.html>>