

Pequeno Manual do Linux

Sumário

Sobre o autor.....	2
Introdução.....	3
História do Linux.....	4
1 – Desktop.....	15
1.1 – Comandos.....	15
1.2 - Compactação e Descompactação.....	16
1.3 - gerenciamento de pacotes com apt.....	19
1.3.1 – Copiar arquivos preservando permissões.....	21
1.3.2 – Editor de texto para o terminal.....	22
1.3.3 – mc - Gerenciador de arquivos para o terminal.....	23
1.3.4 – Navegador para o modo texto.....	24
1.4 - Permissões de arquivos.....	25
1.5 – Cron.....	26
1.6 - Calendário (ncal) e Calculadora (bc).....	33
1.7 – Aplicativos para o modo gráfico.....	34
1.8 – Diferenças entre Linux e Windows.....	41
1.8 - Recuperando um linux do windows.....	43
3 - Shell Scripts.....	48
3.1 - aliases.....	48
3.2 – Exemplos de scripts.....	50
3.2.1 – Ajustando o .env.....	50
3.2.2 – Backup incremental.....	51
3.2.7 – Maiores arquivos do diretório atual.....	53
3.2.8 – Instalação do servidor web.....	53

Sobre o autor

Rubamar FS

<https://github.com/ribafs2>

<https://github.com/ribafs>

<https://ribafs2.github.io/portal/>

<https://ribafs2.github.io/blog/>

<https://ribamar.net.br> (sem conteúdo atualmente)

Introdução

Linux é o sistema utilizado em 96,3% dos principais servidores web e em mais de 460 dos 500 supercomputadores mais poderosos do mundo.

Linux Statistics (Editor's Choice)

- 47% of professional developers use Linux-based operating systems. (Statista)
- Linux powers 39.2% of websites whose operating system is known. (W3Techs)
- Linux powers 85% of smartphones. (Hayden James)
- Linux, the third most popular desktop OS, has a market share of 2.09%. (Statista)
- The Linux market size worldwide will reach \$15.64 billion by 2027. (Fortune Business Insights)
- The world's top 500 fastest supercomputers all run on Linux. (Blackdown)
- 96.3% of the top one million web servers are running Linux. (ZDNet)
- Today, there are over 600 active Linux distros. (Tecmint)

<https://truelist.co/blog/linux-statistics/>

Ubuntu	24.5%
Debian	13.2%
CentOS	5.2%
Red Hat	0.4%
AlmaLinux	0.3%
Gentoo	0.3%
Fedora	0.1%

<https://w3techs.com/technologies/details/os-linux>

Linux - Entenda como tudo começou! Um projeto pessoal que se tornou um grande sistema
<https://www.youtube.com/watch?v=PSFDpRHol5I>

História do Linux

A História do Linux começou em 1991 com o início de um projeto pessoal de um estudante finlandês chamado Linus Torvalds com o intuito de criar um novo núcleo de sistema operacional.

Desde então, o núcleo Linux resultante foi marcado por um crescimento constante através de sua história. A partir do lançamento inicial de seu código-fonte em 1991, cresceu de um pequeno grupo de arquivos em C sob uma proibitiva licença de distribuição comercial, em 1991, a possuir mais de 370 megabytes de fonte sob a licença GPL.[1]

Eventos que levaram à criação

O sistema operativo Unix foi concebido e implementado por Ken Thompson e Dennis Ritchie (ambos dos AT&T Bell Laboratories) em 1969 e primeiramente lançado em 1970. Sua disponibilidade e portabilidade fizeram com que fosse amplamente adotado, copiado e modificado por instituições acadêmicas e negócios. Seu design influenciou autores de outros sistemas.[carece de fontes]

Em 1983, Richard Stallman começou o Projeto GNU com o objetivo de criar um Sistema operacional tipo Unix gratuito e livre.[2] Como parte desse trabalho, ele escreveu a GNU General Public License (GPL). No começo dos anos 1990, havia software quase suficiente para se criar um sistema operacional completo. Entretanto, o núcleo GNU, chamado de Hurd, não conseguiu atrair atenção suficiente dos desenvolvedores, deixando o GNU incompleto.

Outro projeto de sistema operacional livre, inicialmente lançado em 1977, foi o Berkeley Software Distribution (BSD). Foi desenvolvido pela Universidade da Califórnia em Berkeley a partir da versão 6 do Unix da AT&T. Uma vez que o BSD continha código do Unix do qual a AT&T era proprietária, a AT&T entrou com um processo (USL v. BSDi) no começo dos anos 1990 contra a Universidade da Califórnia. Isso limitou fortemente o desenvolvimento e adoção do BSD.[3][4]

Em 1985, a Intel lançou o 80386, o primeiro microprocessador x86 com conjunto de instruções de 32-bit e MMU com paginação.[5]

Em 1986, Maurice J. Bach, of AT&T Bell Labs, publicou The Design of the UNIX Operating System.[6] Essa descrição definitiva cobria principalmente o núcleo System V versão 2, com algumas novas características da versão 3 e do BSD.

O MINIX, um sistema operacional tipo Unix pensada para uso acadêmico, foi lançado por Andrew S. Tanenbaum em 1987. Se bem o código-fonte do sistema estava disponível, modificações e redistribuições não era permitidas. Ademais, o design do MINIX de 16-bit não se adaptou muito bem às características da cada vez mais barata e popular arquitetura de 32-bit do Intel 386 para computadores pessoais.

Esses fatores e a falta de uma adoção ampla de um kernel livre deram o impulso para que Torvalds iniciasse seu projeto. Ele declarou que se o núcleo GNU ou o núcleo 386BSD estivessem disponíveis naquela época, ele possivelmente não teria escrito o seu próprio.[7][8]

A criação do Linux

Linus Torvalds em 2002

Em 1991, em Helsinque, Linus Torvalds começou o projeto que mais tarde se tornaria o núcleo Linux. Era inicialmente um emulador de terminal, o qual Torvalds usava para acessar os grandes servidores UNIX da universidade. Ele escreveu um programa especificamente para o hardware que estava usando e independente de um sistema operacional porque queria usar as funções de seu novo computador com um processador 80386. O desenvolvimento foi feito no MINIX usando o GNU C compiler, o qual é ainda hoje a escolha principal para compilar o Linux (embora o código possa ser construído com outros compiladores, como o Intel C Compiler).[carece de fontes]

Como Torvalds escreveu em seu livro *Just for Fun*,[9] ele eventualmente percebeu que havia escrito o núcleo de um sistema operacional. No dia 25 de agosto de 1991, ele anunciou esse sistema em um post no newsgroup "comp.os.minix." da Usenet:[10]

Olá a todos que estão usando minix -

Eu estou fazendo um sistema operacional livre (é apenas um hobby, não será grande e profissional como o gnu) para os clones AT 386(486). Está sendo desenvolvido desde abril e está quase pronto. Gostaria de receber qualquer feedback sobre o que as pessoas gostam/não gostam no minix, uma vez que o meu SO se parece um pouco com ele (mesmo layout físico de sistema de arquivos (devido a razões práticas) entre outras coisas.

No momento eu o portei para bash(1.08) e gcc(1.40), e as coisas parecem funcionar. Isso implica que irei conseguir algo prático dentro de poucos meses e gostaria de saber quais características a maioria das pessoas gostaria que tivesse. Quaisquer sugestões são bem-vindas, mas não prometo que eu vá implementá-las :-)

Linus (torvalds@kruuna.helsinki.fi)

PS. Sim — ele não tem nenhum código minix, e possui um fs multitarefa. Ele NÃO é portátil (usa troca de contexto 386, etc), e provavelmente nunca será compatível com nada além de discos rígidos AT, uma vez que isso é tudo o que eu tenho :-).

— Linus Torvalds [11]

O nome

Disquetes contendo uma versão bem inicial do Linux.

Linus Torvalds queria chamar seu invento de Freax, um portmanteau de "freak", "free", and "x" (como uma alusão ao Unix). Durante o começo de seu trabalho no sistema, ele guardou os arquivos sob o nome de "Freax" por cerca de um ano. Torvalds já havia considerado o nome "Linux," mas inicialmente o descartou por ser demasiadamente egocêntrico.[9]

Com o intuito de facilitar o desenvolvimento, foi feito o upload dos arquivos para o FTP server (ftp.funet.fi) da FUNET em setembro de 1991. Ari Lemmke, que trabalhava junto com Torvalds na Universidade de Helsinque e era um dos administradores voluntários do servidor FTP naquela época, não achava que "Freax" fosse um bom nome. Então, deu ao projeto o nome de "Linux" no servidor sem consultar Torvalds.[9] Mais tarde, contudo, Torvalds consentiu o nome "Linux".

Para demonstrar como a palavra "Linux" deveria ser pronunciada, Torvalds incluiu um guia de áudio (reproduzirⁱ) com o código-fonte do núcleo.[12]

Linux sob a licença GNU GPL

Torvalds primeiramente publicou o núcleo Linux sob sua própria licença, que tinha restrições no que diz respeito à atividade comercial.

O software a ser usado junto com o núcleo era o desenvolvido como parte do Projeto GNU, licenciado sob os termos da GNU General Public License, uma licença de software livre. O primeiro lançamento do núcleo Linux, o Linux 0.01, incluía um binário do Bash shell do GNU.[13]

Nas "Notas para o lançamento do linux 0.01", Torvalds listou que o software GNU era necessário para o funcionamento do Linux:[13]

Infelizmente, um núcleo por si só não leva a lugar nenhum. Para conseguir um sistema que funcione, são necessários um shell, compiladores, uma biblioteca etc. Essas são partes separadas e podem estar sob um copyright mais restrito (ou mesmo menos restrito). A maioria das ferramentas usadas com o linux são software GNU e estão sob o copyleft GNU. Tais ferramentas não estão na distribuição - pergunte-me (ou ao GNU) para mais informações.[13]

Em 1992, ele sugeriu o lançamento do núcleo sob a GNU General Public License. Ele anunciou sua decisão primeiramente nas notas de lançamento da versão 0.12.[14] Em meados de dezembro de 1992, ele publicou a versão 0.99 usando a GNU GPL.[15]

Desenvolvedores do Linux e do GNU trabalharam para integrar os componentes do GNU com o Linux para fazer um sistema operacional totalmente funcional e totalmente livre.[16]

Torvalds declarou que “tornar o Linux compatível com a GPL foi definitivamente a melhor coisa que eu já fiz.”[17]

Controvérsia quanto à nomenclatura GNU/Linux

Ver artigo principal: Controvérsia quanto à nomenclatura GNU/Linux

A designação "Linux" foi usada inicialmente por Torvalds apenas para o núcleo Linux. O núcleo foi, contudo, usado frequentemente em conjunto com outros softwares, especialmente os do projeto GNU, tornando-se rapidamente a mais popular adoção do software GNU. Em junho de 1994, no boletim do GNU, o Linux era referido como um "clone livre do UNIX", e o projeto Debian começou a chamar seu produto de Debian GNU/Linux. Em maio de 1996, Richard Stallman publicou o editor Emacs 19.31, no qual o tipo do sistema foi renomeado de Linux para Lignux. Essa grafia tinha a intenção de referir-se especificamente à combinação de GNU e Linux, mas foi rapidamente abandonada e substituída por "GNU/Linux".[18]

Esse nome provocou reações variadas. Os projetos GNU e Debian usam esse nome, embora a maioria das pessoas simplesmente usam o termo "Linux" para referir-se à combinação.[19]

Mascote oficial

Ver artigo principal: Tux

Tux

Torvalds anunciou em 1996 que haveria um mascote para o Linux, um pinguim. Isso deve-se ao fato de que quando se estava por escolher um mascote, Torvalds mencionou que ele havia sido bicado por um pequeno pinguim em uma visita ao Zoológico & Aquário Nacional de Camberra, Austrália. Larry Ewing foi o responsável pelo esboço original do hoje bem conhecido mascote, baseado em sua descrição. O nome "Tux" foi sugerido por James Hughes como um derivativo de Torvalds' UniX.[9]

Novo desenvolvimento

Núcleo/Kernel

Existem muitos outros conhecidos mantenedores do núcleo Linux além de Torvalds, como Alan Cox e Marcelo Tosatti. Cox foi o mantenedor da versão 2.2 do núcleo até esse ser descontinuado no final de 2003. Da mesma maneira, Tossatti foi o mantenedor da versão 2.4 do núcleo até meados de 2006. Andrew Morton é o responsável pelo desenvolvimento e administração do núcleo 2.6, o qual teve sua primeira versão estável lançada em 18 de dezembro de 2003. Também as branches mais antigas ainda são constantemente aprimoradas.

Comunidade

LinuxTag em Karlsruhe, 2004.

A maior parte do trabalho no Linux é feita pela comunidade: os milhares de programadores em todo o mundo que usam Linux e enviam suas sugestões de aprimoramento aos mantenedores. Várias companhias também ajudaram não apenas no desenvolvimento dos núcleos, mas também com a escrita do corpo do software auxiliar, que é distribuído junto com o Linux.

É lançado tanto por projetos organizados como o Debian, quanto por projetos conectados diretamente a companhias como o Fedora e o openSUSE. Os membros dos respectivos projetos encontram-se em várias conferências e eventos, com o objetivo de intercambiar ideias. Um dos maiores desses eventos é o LinuxTag na Alemanha (atualmente em Berlim), onde cerca de 10.000 pessoas se reúnem anualmente, com o intuito de discutir sobre o Linux e sobre projetos associados a ele.

O Open Source Development Labs e a Linux Foundation

O Open Source Development Labs (OSDL) foi criado no ano 2000, e é uma organização independente sem fins lucrativos que persegue o objetivo de otimizar o Linux para uso em data centers, entre outros. Funcionou como patrocinadora de Linus Torvalds e também de Andrew Morton (até meados de 2006 quando Morton transferiu-se para a Google). Torvalds trabalha em tempo integral para o OSDL, desenvolvendo núcleos do Linux.

Em 22 de janeiro de 2007, o OSDL e o Free Standards Group uniram-se para formar a Linux Foundation, estreitando seus respectivos focos para a promoção da competição do GNU/Linux com o Microsoft Windows.[20]

Empresas

Apesar do fato ter o código-fonte aberto, algumas poucas empresas obtém lucro do Linux. Essas companhias, a maiorias das quais também são membros do Open Source Development Lab, investem recursos substanciais no avanço e desenvolvimento do Linux, com o objetivo de torná-lo adequado para aplicação em várias áreas. Isso inclui doação de hardware para os desenvolvedores de drivers, doações em dinheiro para as pessoas que desenvolvem software Linux, e o emprego de programadores do Linux na companhia. Alguns exemplos são a IBM e a HP, que usam o Linux em seus próprios servidores, e a Red Hat, que mantém sua própria distribuição. Da mesma forma, a Nokia apoia o Linux pelo desenvolvimento e licenciamento do Qt em LGPL, o qual torna possível o desenvolvimento do KDE, e pelo emprego de alguns desenvolvedores do X e do KDE.

Controvérsia sobre o Linux

O Linux sempre tem sido cercado por controvérsias desde a sua concepção.

"O Linux é obsoleto"

Ver artigo principal: Debate entre Tanenbaum e Torvalds

Em 1992, Andrew S. Tanenbaum, reconhecido cientista da computação e autor do sistema de micronúcleo Minix, escreveu um artigo no newsgroup comp.os.minix da Usenet com o título "O Linux é obsoleto",[21] o qual marcou o começo de um famoso debate sobre a estrutura do então recente núcleo Linux. Entre as críticas mais significativas figuravam as seguintes:

O núcleo era monolítico e portanto ultrapassado.

A falta de portabilidade, devido ao uso de características exclusivas dos processadores Intel 386. "Escrever um sistema operacional que está fortemente atado a um tipo particular de hardware, especialmente um esquisito como são os da linha Intel, é basicamente errado."[22]

Não há controle estrito sobre o código-fonte por nenhuma pessoa em particular.[23]

O Linux empregava uma série de características que eram inúteis (Tanenbaum acreditava que um sistema de arquivos que realizasse multitarefas fosse simplesmente um "hack para melhorar o desempenho").[24]

A previsão de Tanenbaum de que o Linux se tornaria ultrapassado dentro de poucos anos e seria substituído pelo GNU Hurd (que ele considerava ser mais moderno) mostrou-se incorreta. O Linux foi portado para todas as principais plataformas e seu modelo de desenvolvimento aberto levou a um exemplar caminho de desenvolvimento. Em contraste, o GNU Hurd ainda não alcançou o nível de estabilidade que permitiria que fosse usado em locais de produção.[25] Sua rejeição à linha de processadores 386 da Intel por serem "esquitos" também provou-se nada visionária, uma vez que a série de processadores x86 e a Intel Corporation tornaria-se mais tarde quase ubíqua nos computadores pessoais.

Samizdat

Em seu livro não-publicado Samizdat, Kenneth Brown declarou que Torvalds copiou ilegalmente código do MINIX. Essas acusações foram refutadas por Tanenbaum:[26]

Ele [Kenneth Brown] queria entrar na questão da propriedade, mas ele também estava tentando evitar me dizer qual era seu verdadeiro propósito, então ele não formulava [a questão] muito bem. Finalmente ele me perguntou se eu pensava que Linus escreveu o Linux. Eu disse que até onde eu sabia, Linus escreveu todo o núcleo sozinho, mas que após seu lançamento, outras pessoas começaram a aperfeiçoar o núcleo, o qual era muito primitivo inicialmente, e acrescentarem novos softwares ao sistema — essencialmente o mesmo modelo de desenvolvimento do MINIX. Então ele começou a focar nisso, com questões como: "Ele não roubou partes do MINIX sem permissão?" Eu disse a ele que o MINIX tinha claramente uma enorme influência sobre o Linux em muitos aspectos, do layout do sistema de arquivos ao código-fonte, mas que eu não pensava que Linus tivesse usado meu código.

As declarações, a metodologia e as referências do livro foram seriamente questionadas e no final nunca foram lançadas, sendo descartadas pelo site da distribuidora.

Competição com a Microsoft

Embora Torvalds tenha dito que a sensação da Microsoft de sentir-se ameaçada pelo Linux no passado não tenha tido consequências sobre ele, os campos da Microsoft e do Linux tiveram várias interações antagonísticas entre 1997 e 2001. Isso tornou-se bastante claro pela primeira vez em 1998, quando o primeiro Halloween document foi trazido à luz por Eric S. Raymond. Era uma espécie de pequeno ensaio de um desenvolvedor da Microsoft que procurava expor as ameaças que o software livre representava para a Microsoft e identificava estratégias para contrapor-se às ameaças percebidas. Contudo, a Free Software Foundation publicou uma declaração dizendo que a produção de software proprietário pela Microsoft era ruim para os usuários de software pois negava

a eles a sua "devida liberdade." [27] A competição entrou em uma nova fase no começo de 2004, quando a Microsoft publicou os resultados de estudos de caso de consumidores avaliando o uso do Windows em comparação com o Linux, com o nome de "Get the Facts" (Obtenha os Fatos) em sua própria página na internet. Baseado em consultas, analistas, e alguns pesquisadores financiados pela Microsoft, os estudos de caso sustentavam que o uso empresarial do Linux em servidores era comparativamente inferior ao uso do Windows em termos de confiabilidade, segurança e custo total de propriedade. [28]

Em resposta, distribuidores comerciais do Linux produziram seus próprios estudos, enquetes e testemunhos para opor-se à campanha da Microsoft. No final de 2004, a campanha da Novell foi intitulada "Unbending the truth" (Desvelando a verdade) e procurou sublinhar as vantagens bem como dissipar as questões legais amplamente divulgadas na adoção do Linux (particularmente à luz do caso SCO v IBM). A Novell referiu-se particularmente aos estudos da Microsoft em muitos pontos. A IBM também publicou uma série de estudos sob o título de "The Linux at IBM competitive advantage" (O Linux e a vantagem competitiva da IBM) para, mais uma vez, fazer frente à campanha da Microsoft. A Red Hat lançou uma campanha chamada "Truth Happens" (A verdade acontece) focada em deixar a performance do produto falar por si só, ao invés de fazer propaganda dele através de estudos. [carece de fontes] A maioria dos membros da comunidade Linux lidou com a questão de maneira calma e com piadas como "O Linux e seus computadores nunca ficam azuis" ou "Mais cedo ou mais tarde migraremos". Além disso, a revista LinuxUser também publicou um review do Windows XP meio em tom de brincadeira com as mesmas críticas comumente feitas a uma típica distribuição do Linux.

No outono de 2006, a Novell e a Microsoft anunciaram um acordo de cooperação em interoperabilidade de software e proteção de patentes. [29] Isso incluía um acordo que tanto os clientes da Novell quanto os da Microsoft não poderiam ser processados por violação de patentes. Essa proteção de patentes também foi expandida para desenvolvedores de software não-livre. Essa última parte foi criticada porque incluía apenas desenvolvedores de software não-comercial.

Em julho de 2009, a Microsoft submeteu 22.000 linhas de código-fonte para o núcleo Linux sob a licença GPLv2, as quais foram subsequentemente aceitas. Embora isso tenha sido referido como "um gesto histórico" e como um possível indicativo de uma melhoria nas atitudes corporativas da Microsoft para com o Linux e o software de código-aberto, a decisão não foi totalmente altruísta, uma vez que prometia levar a significativas vantagens competitivas para a Microsoft e evitar ações legais contra ela. A Microsoft na verdade foi compelida a fazer essa contribuição para o código quando Stephen Hemminger, o principal engenheiro da Vyatta e colaborador do Linux, descobriu que a Microsoft havia incorporado um driver de rede Hyper-V, com componentes de licença de código-aberto GPL, estaticamente ligados a binários de código-fechado, em contravenção à licença GPL. A Microsoft contribuiu com os drivers para retificar a violação de licença, embora a companhia houvesse tentado passar a imagem de ter feito um ato caridoso, e não a de ter tentado evitar uma ação legal contra si. No passado a Microsoft qualificou o Linux de "câncer" e "comunista". [30][31][32][33][34]

Distribuições Linux

Algumas de suas distribuições principais em termos de estabilidade e/ou popularidade:

- Debian
- Slackware
- Arch
- OpenSUSE

Manjaro
Sabayon
SolusOS
Antergos
Mageia
Redhat
Kali Linux
Fedora
Ubuntu

Linux Educacional - Um projeto do Governo Federal que visa melhorar o aproveitamento dos laboratórios de informática das escolas, tendo diversas versões. Atualmente na versão 5.0, apresenta diferentes atividades educativas e planos de aula que auxiliam o ensino e a aprendizagem.

SCO

Ver artigo principal: Controvérsia entre SCO e Linux

Em março de 2003, o SCO Group acusou a IBM de violar seu copyright sobre o UNIX ao transferir código do UNIX para o Linux. O SCO reivindica a propriedade dos direitos autorais sobre o UNIX e um processo foi aberto contra a IBM. A Red Hat, então, entrou com um processo contra o SCO que, desde então, foi alvo de vários processos relacionados a esse tema. Ao mesmo tempo em que entrou com sua ação, o SCO começou a vender licenças Linux para usuários que não quisessem correr o risco de uma possível reclamação por parte da empresa. Porém, como a Novell também alega possuir direitos autorais sobre o UNIX, ela entrou com uma ação contra o SCO.

Posteriormente, o SCO entrou com um pedido de falência.[35]
Direitos sobre a marca

Entre 1994 e 1995, várias pessoas de diferentes países tentaram registrar o nome "Linux" como uma marca registrada comercial. Logo após, pedidos de pagamentos de royalties foram emitidos para várias empresas Linux, um passo com o qual muitos desenvolvedores e usuários do Linux não concordaram. Linus Torvalds reagiu contra essas empresas com a ajuda da Linux International e foi concedida a marca comercial para o nome, a qual ele transferiu para a Linux Internacional. Posteriormente, a proteção da marca foi administrada por um empresa sem fins lucrativos específica, a Linux Mark Institute. No ano 2000, Linus Torvalds especificou as regras básicas para a atribuição de licenças. Isso significa que qualquer um que ofereça um produto ou serviço com o nome "Linux" deve possuir uma licença para tanto, a qual pode ser obtida através de uma compra única.

Em junho de 2005, uma nova controvérsia surgiu no que tange ao uso dos royalties gerados pelo uso da marca Linux. O Linux Mark Institute, representante dos direitos de Linus Torvalds, anunciou um aumento de preço de 500 para 5.000 dólares para o uso do nome. Esse passo foi justificado como sendo necessário para cobrir os crescentes custos de proteção da marca.

Em resposta a esse aumento, que desagradou a comunidade, Linus Torvalds fez um anúncio no dia 21 de agosto de 2005, com o objetivo de esclarecer os mal-entendidos. Ele descreveu a situação e o cenário em um e-mail, além de tratar da questão de quem tinha que pagar os custos da licença:

[...] E vamos repetir: alguém que não quer `_proteger_` aquele nome nunca faria isso. Você pode chamar qualquer coisa de "MyLinux", mas o lado negativo é que pode haver alguém que sim

protegeu e que lhe envie uma carta exigindo que você desista do nome. Ou ainda, se o nome acabar aparecendo em uma pesquisa de marcas comerciais que o LMI precisa fazer periodicamente apenas para proteger a marca (outra exigência legal para marcas comerciais), o próprio LMI pode lhe enviar uma carta exigindo que você desista do nome ou adquira uma sublicença.

Em dado momento ou você escolhe outro nome ou adquire uma sublicença. Está vendo? Trata-se do fato de que o _você_ precisa ou não de proteção, e não de que o LMI queira dinheiro ou não.

[...] Finalmente, apenas para deixar claro: não apenas eu não ganho um centavo do dinheiro advindo da marca, como o próprio LMI (quem realmente administra a marca) historicamente tem até agora sempre perdido dinheiro com isso. Essa não é uma maneira de sustentar uma marca, então eles estão tentando no mínimo tornarem-se autossuficientes, mas até agora eu posso afirmar que as taxas dos advogados que dão a proteção que as empresas comerciais querem têm sido maiores que as taxas das licenças. Mesmo os advogados pro bono cobram pelo tempo de seus custos e paralegais etc.

— Linus Torvalds [36]

Desde então, o Linux Mark Institute começou a oferecer uma sublicença gratuita com cobertura mundial.[37]

Cronologia

1983: Richard Stallman lança o projeto GNU com o objetivo de criar um sistema operacional livre.

1989: Richard Stallman escreve a primeira versão da GNU General Public License.

1991: O núcleo Linux é anunciado publicamente no dia 25 de agosto por um estudante finlandês de 21 anos chamado Linus Benedict Torvalds.[10]

1992: O núcleo Linux é licenciado sob os termos da GNU GPL. As primeiras "distribuições do Linux" são criadas.

1993: Mais de 100 desenvolvedores trabalham no núcleo Linux. Com a sua ajuda, o núcleo é adaptado ao ambiente GNU, o que cria um grande espectro de tipos de aplicação para o Linux. A mais antiga distribuição do Linux ainda existente, o Slackware, é lançado pela primeira vez. Mais tarde, no mesmo ano, é estabelecido o projeto Debian. Hoje é a distribuição que possui a maior comunidade.

1994: Em março, Torvalds julga que todos os componentes do núcleo estão totalmente amadurecidos: ele lança a versão 1.0 do Linux. O projeto XFree86 contribui com uma interface gráfica de usuário (GUI). Nesse ano, as companhias Red Hat e SUSE publicam a versão 1.0 de suas distribuições do Linux.

1995: O Linux é portado para o DEC Alpha e para o Sun SPARC. Nos anos seguintes é portado para um número ainda maior de plataformas.

1996: A versão 2.0 do núcleo Linux é lançada. O núcleo agora pode servir a vários processadores ao mesmo tempo, tornando-se então uma alternativa séria para muitas companhias.

1998: Muitas grandes companhias como a IBM, a Compaq e a Oracle anunciam seu apoio ao Linux. Ademais, um grupo de programadores começa a desenvolver a interface gráfica de usuário KDE.

1999: Um grupo de desenvolvedores começa a trabalhar no ambiente gráfico GNOME, que deveria tornar-se um substituto livre para o KDE, o qual dependia do então software proprietário Qt toolkit. A IBM anuncia um grande projeto em apoio ao Linux.

2004: A equipe XFree86 separa-se e junta-se ao padrão existente X Window para formar a X.Org Foundation, o que resulta em um desenvolvimento substancialmente mais rápido do X Window Server para Linux.

2005: O projeto openSUSE começa a lançar uma distribuição livre da comunidade da Novell. Além disso, em outubro, o projeto OpenOffice.org introduz a versão 2.0, compatível com os padrões da OASIS OpenDocument.

2006: A Oracle lança sua própria distribuição da Red Hat. A Novell e a Microsoft anunciam uma cooperação para uma melhor interoperabilidade.

2007: A Dell começa a vender laptops com o Ubuntu pré-instalado.

2011: A versão 3.0 do núcleo Linux é lançada.

2015: A versão 4.0 do núcleo Linux é lançada.

2016-atual: A versão 4.4 do núcleo Linux é lançada.

Ver também

Portal do software livre

História do software livre

Refêrencias

«Linux Headquarters' kernel page». Consultado em 25 de dezembro de 2008

Anúncio inicial do Projeto GNU, 1983

«Berkeley UNIX and the Birth of Open-Source Software»

Marshall Kirk McKusick. «Twenty Years of Berkeley Unix From AT&T-Owned to Freely Redistributable»

Intel 80386

Bach, Maurice (1986), *The Design of the UNIX Operating System*, ISBN 0-13-201799-7, Prentice Hall

«Linus vs. Tanenbaum debate». Consultado em 28 de agosto de 2014. Arquivado do original em 3 de outubro de 2012

«The Choice of a GNU Generation - An Interview With Linus Torvalds»

Torvalds, Linus and David Diamond, *Just for Fun: The Story of an Accidental Revolutionary*, 2001, ISBN 0-06-662072-4

Torvalds, Linus Benedict (1991). «comp.os.minix». Consultado em 6 de setembro de 2009

Torvalds, Linus: What would you like to see most in minix? Usenet group comp.os.minix, August 25, 1991.

Torvalds, Linus (1994). «Index of /pub/linux/kernel/SillySounds». Consultado em 3 de agosto de 2009

Torvalds, Linus: Notes for linux release 0.01 kernel.org, 1991.

Torvalds, Linus (5 de janeiro de 1992). «RELEASE NOTES FOR LINUX v0.12». Linux Kernel Archives. Consultado em 23 de julho de 2007. "O copyright do Linux vai mudar: Eu recebi alguns pedidos para fazê-lo compatível com o copyleft GNU, removendo a condição "você não pode redistribuí-lo por dinheiro. Eu concordo. Eu propus a mudança no copyright que confirma a compatibilidade com o GNU - esperando a aprovação das pessoas que ajudaram a escrever o código. Eu acredito que isso não será problema para ninguém: Se você tem queixas ("Eu escrevi aquele código achando que o copyright permaneceria o mesmo") escreva-me. Do contrário, o copyleft GNU passa a ter efeito a partir de 1º de fevereiro. Se você não sabe o essencial do copyright GNU - leia-o."

z-archive of Linux version 0.99[ligação inativa], kernel.org, December 1992

[1]

Hiroo Yamagata: The Pragmatist of Free Software, Linus Torvalds Interview, 05.08.1997

Linux and GNU - GNU Project - Free Software Foundation (FSF)

Govind, Puru (2006). «The "GNU/Linux" and "Linux" Controversy». Consultado em 26 de outubro de 2008. Cópia arquivada em 25 de fevereiro de 2009

«New Linux Foundation Launches — Merger of Open Source Development Labs and Free Standards Group» (Nota de imprensa). The Linux Foundation. 22 de janeiro de 2007. Consultado em 21 de janeiro de 2007. "A computação está entrando em um mundo dominado por duas plataformas: Linux e Windows."

A. S. Tanenbaum (29 de janeiro de 1992). «LINUX is obsolete». Grupo de notícias: comp.os.minix. 12595@star.cs.vu.nl. Consultado em 16 de julho de 2008

A. S. Tanenbaum (30 de janeiro de 1992). «Re: LINUX is obsolete». Grupo de notícias: comp.os.minix. 12615@star.cs.vu.nl. Consultado em 16 de julho de 2008

Tanenbaum, Andy (1992). «Re: Unhappy campers». Consultado em 19 de outubro de 2008. Arquivado do original em 5 de junho de 2009

Andrew Tanenbaum, Linus Torvalds and others: Linux is obsolete Arquivado em 3 de maio de 2010, no Wayback Machine. Usenet post, 29.01.1992

The GNU Hurd Project

Tanenbaum, Andrew S. «Some Notes on the "Who wrote Linux" Kerfuffle, Release 1.5». Consultado em 22 de outubro de 2009

Free Software Foundation (19 de junho de 2007). «Is Microsoft the Great Satan?». Philosophy of the GNU Project. Consultado em 14 de agosto de 2007. "Vemos que a Microsoft está fazendo algo que é ruim para os usuários de software; fazendo software proprietário e portanto negando aos usuários sua devida liberdade. Mas a Microsoft não está sozinha nisso: quase todas as empresas de software fazem a mesma coisa com os usuários. Se outras empresas conseguem dominar menos usuários que a Microsoft, isso não é por falta de tentar."

«Get the Facts». Microsoft. 2004. Consultado em 24 de novembro de 2006

«Open Letter to Community from Novell». Novell. 2006. Consultado em 23 de março de 2007

John Fontana. «Microsoft stuns Linux world, submits source code to kernel». Consultado em 20 de julho de 2009. Arquivado do original em 2 de dezembro de 2009. "Em um gesto histórico, a Microsoft submeteu, na segunda-feira, código-fonte de driver para inclusão no núcleo Linux sob uma licença GPLv2 license. [...] Greg Kroah-Hartman, o Linux driver project lead and a Novell fellow, disse ter aceitado as 22.000 linhas do código da Microsoft às 09:00 hs da segunda-feira. Kroah-Hartman disse que o código da Microsoft estará disponível como parte do próximo lançamento do Linux nas próximas 24 horas. O código se tornará parte da lançamento da versão estável 2.6.30.1 [...] Então todo o mundo poderá ver o código, afirmou."

Paul, Ryan (2009). «Microsoft aims at VM market with Linux kernel code offering». Consultado em 22 de julho de 2009

Holwerda, Thom (2009). «Microsoft's Linux Kernel Code Drop Result of GPL Violation». Consultado em 23 de julho de 2009

Microsoft (2009). «Microsoft Contributes Linux Drivers to Linux Community». Consultado em 23 de julho de 2009

Richmond, Gary (2009). «Yes Linus, Microsoft hating is a disease. And it's a pandemic». Consultado em 1 de agosto de 2009. Arquivado do original em 29 de julho de 2009

«SCO Group files for bankruptcy protection». CNet News.com. 14 de setembro de 2007. Consultado em 14 de janeiro de 2008

Linus Torvalds: Linus trademarks Linux?! from the linux-Kernel mailing list, 21.08.2005

«Linux Mark Institute». Consultado em 24 de fevereiro de 2008. "O LMI reestruturou seu programa de sublicenciamento. Nosso novo acordo de sublicenciamento é: Grátis — detentores de sublicenças aprovadas não pagam taxas; Perpétuo — a sublicença termina apenas em razão de quebra de contrato ou quando a sua organização deixa de usar sua marca; Cobertura Mundial — uma sublicença cobre o uso da marca em qualquer lugar do mundo"

Fonte: Wikipedia

https://pt.wikipedia.org/wiki/Hist%C3%B3ria_do_Linux

1 – Desktop

Um bom computador desktop para o administrador de redes ou programador é muito importante.

Tanto o hardware precisa ser adequado quanto o sistema operacional, arquitetura, distribuição, versão e aplicativos.

Se for comprar um novo hardware é importante consultar a lista de compatibilidade com a distribuição que pretende usar e consultar um bom grupo sobre a experiência dos colegas com o hardware ou já comprar com linux instalado

Qual a melhor distribuição desktop?

É sempre relativa ao usuário que fará uso, mas existem pontos importantes

- Aquela que suporta perfeitamente seu hardware
- Que seja customizável, caso você pretenda customizar
- Que ofereça todos os principais recursos que você espera dela para uso com facilidade:
 - Multimídia: codecs de áudio, vídeo e suporte a diversos formatos: mp3, mp4, webm, mkv, etc
 - Documentos: pdf, odt, doc, docx, etc
 - Navegação pela internet
 - Planilhas e apresentações
 - Teclas de atalho customizadas
 - Gerenciador de arquivos com suporte (peça chave numa boa distribuição):
 - Suporte a painel duplo (dois painéis numa mesma tela, dividindo-a em duas)
 - Clique único para abrir pastas e arquivos
 - Conexão a servidores ftp, ssh, etc
 - Criação de atalhos para pastas na lateral esquerda
- Precisa estar em pleno desenvolvimento, ter frequentes atualizações e uma grande comunidade
- Ser robusta e leve são quesitos importantes, mas o mais importante é o suporte ao hardware e a usabilidade aliados aos recursos que a tornam de uso simples

1.1 – Comandos

O terminal do Linux é um verdadeiro canivete suíço, com uma enormidade de comandos disponíveis para uso com o sistema operacional. Atualmente o terminal tem ganhado ainda mais força e popularidade com ferramentas como composer, o artisa, tinker, etc.

Atualmente, se o programador tem um computador com bons recursos, já pode usar o Linux dentro do windows, através do WSL2. Podendo instalar as principais distribuições e então usar o terminal linux e os principais comandos do linux em um servidor;

1.2 - Compactação e Descompactação

Compactar e Descompactar Arquivos no Terminal do Linux

Instalar compactadores para Linux:

```
sudo apt-get install unace zip unzip p7zip-full p7zip-rar sharutils uuview mpack arj unrar rar lzma lha lzma-dev rar  
unrar-free ark ncompress
```

zip

Compactação

```
zip pacote.zip arquivoa.txt arquivos.txt arquivos.odt
```

```
zip pacote.zip *.txt
```

```
zip -r documentos.zip /usr/*.txt
```

Opções:

-r recursivo

-1 rápido

-9 maior compactação

-D compactar somente arquivos, nada de diretório

-x arquivos - excluir da compactação alguns arquivos

Descompactação

```
unzip nome.zip
```

Em diretório específico

```
unzip nome.zip -d /tmp
```

arj

Compactar

```
arj a pacote.arj arquivo.odt
```

Descompactar

```
arj x pacote.arj
```

tar

Compactar

```
tar -czpvf pacote.tar arquivo1.gif memorando.htm carta.doc
```

```
tar -czpvf pacote.tar pasta
```

Descompactar:

```
tar -zxpvf nomedoarq.tar
```


tar.gz

Compactar

```
tar -czpvf pacote.tar.gz arquivo1.gif memorando.htm carta.doc  
tar -czpvf pacote.tar.gz pasta
```

Descompactar no diretório atual

```
tar -zxpvf pacote.tar.gz -C /tmp
```

tar.bz2

```
tar -jxpvf pacote.tar.bz2
```

Descompactar apenas um arquivo de dentro do pacote

```
tar -xvf pacote.tar.gz foto1.png
```

Lista de parâmetros do tar:

- c – cria um novo arquivo tar;
- M – cria, lista ou extrai um arquivo multivolume;
- p – mantém as permissões originais do(s) arquivo(s);
- r – acrescenta arquivos a um arquivo tar;
- t – exhibe o conteúdo de um arquivo tar;
- v – exhibe detalhes da operação;
- w – pede confirmação antes de cada ação;
- x – extrai arquivos de um arquivo tar;
- z – comprime ou extrai arquivos tar resultante com o gzip;
- j – comprime ou extrai arquivos tar resultante com o bz2;
- f – especifica o arquivo tar a ser usado;
- C – especifica o diretório dos arquivos a serem armazenados.

gzip

Compactar

```
gzip documentos.odt
```

Descompactar

```
gunzip documentos.odt.gz
```

bzip2

Compactar

```
bzip2 pacote.gz
```

Descompactar

```
bunzip2 pacote.bz2
```

```
bunzip2 pacote.tar.bz2
```

rar

Compactar

rar a pacote.rar arquivoa arquivob

rar a pacote.rar /pasta

Descompactar

unrar x pacote.rar

7z

Compactar

7za a pacote.7z arquivoa arquivob

Descompactar

7za x pacote.7z

lha

Compactar

lha a pacote.lha arquivoa arquivob

Descompactar

lha x pacote.lha

zoo

Compactar

zoo a pacote.zoo arquivoa arquivob

Descompactar

zoo x pacote.zoo

Ajuda sobre um dos compactadores acima:

man nome_compactador

Ex:

man arj

Referências:

<http://blog.kolaborativa.com/2011/10/compactar-e-descompactar-arquivos-zip-rar-tar-gz-bz2-tar-tar-bz2-pelo-terminal/>

<https://linuxdicasuporte.blogspot.com.br/2015/03/compactacao-de-arquivos-para-debian.html>

1.3 - gerenciamento de pacotes com apt

O que é o Gerenciador de Pacotes Apt?

Quer você faça desenvolvimento de software ou administração de sistemas, o gerenciamento eficiente de pacotes é o coração que mantém tudo funcionando em conjunto. Permita-me apresentar o Apt Package Manager, uma ferramenta útil para sistemas operacionais baseados em Linux. Apt, abreviação de Advanced Package Tool, é um gerenciador de pacotes de linha de comando que simplifica o processo de instalação, atualização e remoção de pacotes de software em distribuições baseadas em Debian, incluindo Ubuntu.

Observação importante

Para usar qualquer comando com o apt precisa ser como root ou usar o sudo.

Apt serve como uma ponte entre o usuário e um vasto repositório de pacotes de software. Ele automatiza a complexa tarefa de resolver dependências, garantindo que quando você instala um pacote e todos os componentes necessários se encaixem perfeitamente.

Atualizando o repositório de pacotes

Antes de mergulhar nas instalações ou atualizações de pacotes, é essencial garantir que seu sistema tenha as informações mais recentes sobre os pacotes disponíveis. Para fazer isso, atualize o repositório de pacotes usando o seguinte comando:

```
apt update
```

Este comando recebe as informações mais recentes dos pacotes dos repositórios configurados em seu sistema para garantir que você tenha acesso aos pacotes de software mais atualizados quando instalar.

Instalar o editor de texto nano

```
apt install nano
```

O Apt resolverá automaticamente as dependências e solicitará confirmação antes de prosseguir com a instalação. Confirme digitando 'Y' e pressionando Enter.

Desinstalar o nano

```
apt remove nano
```

Atualizar a lista de pacotes e também todos os pacotes da distribuição atual

```
apt update
```

```
apt upgrade
```

Pesquisar por um pacote

```
apt search nano
```

ou

```
aptitude search nano
```

Adicionando novos repositórios para a lista atual

Adicionar repo da última versão do NodeJS no Ubuntu

```
add-apt-repository ppa:deadsnakes/ppa
```

```
apt update
```

Remover repositório adicionado

```
add-apt-repository --remove ppa:deadsnakes/ppa
```

Informações sobre um pacote

```
apt show nano
```

Corrigir dependências quebradas

```
apt -f install
```

```
apt autoremove
```

Após a execução do apt update, podemos ver a lista dos pacotes que podem ser instalados com

```
apt list --upgradable
```

Quando temos um pacote .deb que baixamos de algum site, devemos instalar com

```
dpkg -i pacote.deb
```

Corrigir pacotes quebrados

```
sudo apt -f install && sudo dpkg --configure -a
```

```
sudo fuser -cuk /var/lib/dpkg/lock
```

```
sudo rm -f /var/lib/dpkg/lock
```

```
sudo fuser -cuk /var/cache/apt/archives/lock
```

```
sudo rm -f /var/cache/apt/archives/lock
```

```
sudo rm /var/lib/apt/lists/* -vf
```

```
sudo apt update
```

<https://sempreupdate.com.br/corriger-pacotes-quebrados-no-ubuntu-debian-linux-mint/>

1.3.1 – Copiar arquivos preservando permissões

Para copiar arquivos ou diretórios preservando suas permissões, atributos, links, basta utilizar o comando "cp" com a opção "-p":

Exemplo de comando (Copiando Arquivos):

```
cp -p /home/User/teste.txt /etc
```

Exemplo de comando (Copiando Diretórios recursivamente):

```
cp -rp /home/User/ /etc
```

Obs: A opção "-r" copia recursivamente os diretórios, arquivos, links

Mais informações digite: "man cp"

Copiei todo o meu home para

```
mkdir /backup/ribafs
```

```
cp -rp /home/ribafs /backup/ribafs
```

E irei criar uma partição /home

Após a instalação irei copiar tudo de ribafs para /home/ribafs

Melhor, vou fazer isso logo agora.

Remover uma partição não usada e criar uma para /home

Depois de criar adicionar ao fstab

E copiar tudo para ela.

1.3.2 – Editor de texto para o terminal

Um pequeno e bem versátil editor de texto e bem amigável.

```

GNU nano 7.2                               Novo buffer
[ Bem-vindo ao nano. Para uma ajuda básica, digite Ctrl+G. ]
^G Ajuda      ^O Gravar     ^W Onde está? ^K Recortar   ^T Executar   ^C Local      M-U Desfazer   M-A Marcar
^X Sair       ^R Ler o arq ^Y Substituir ^U Colar     ^J Justificar ^_ Ir p/ linha M-E Refazer   M-6 Copiar
  
```

Veja que ele tem um menu abaixo com seus principais comandos. A seguir um resumo.

Editor de textos para o terminal/modo texto no Unix

Resumo

nano arquivo_texto

Ctrl+O ou F3 - salvar o arquivo

Ctrl+X - sair do nano

Ctrl+G - Help

Mover o cursor na horizontal para a esquerda e para a direita - Ctrl+F e Ctrl+B

Mover o cursor na vertical para cima e para baixo - Ctrl+P e Ctrl+N

Também podemos usar as setas do teclado.

Ctrl+W ou F6 - Procurar por parte do texto

Alt+\ - para a primeira linha do texto

Alt+/ - para a última linha

F7 - uma tela acima

F8 - uma tela abaixo

Ctrl+K - apaga a linha atual, melhor, recorta para a memória

Ctrl+U - cola o que está na memória. O conteúdo recortado por Ctrl+K

Ctrl+6 - Inicia a seleção de um texto. Arraste até o final.

No final tecele Ctrl+K para recortar

Mova o cursor para onde deseja colar e tecele Ctrl+U

Para selecionar todo o arquivo

Alt+\

Ctrl+6

Alt+/
 Alt+6 - copiar o texto selecionado

Ctrl+U - colar

Mostrar abaixo o número da linha num arquivo

nano -c arquivo

Ctrl+\ - Sobrescrever um trecho de texto/string

Ctrl+J - justifica um parágrafo

Ctrl+C - mostra abaixo a posição atual do cursor

Ctrl+/- - mudar o cursor para uma linha e coluna. Ex: 10.0

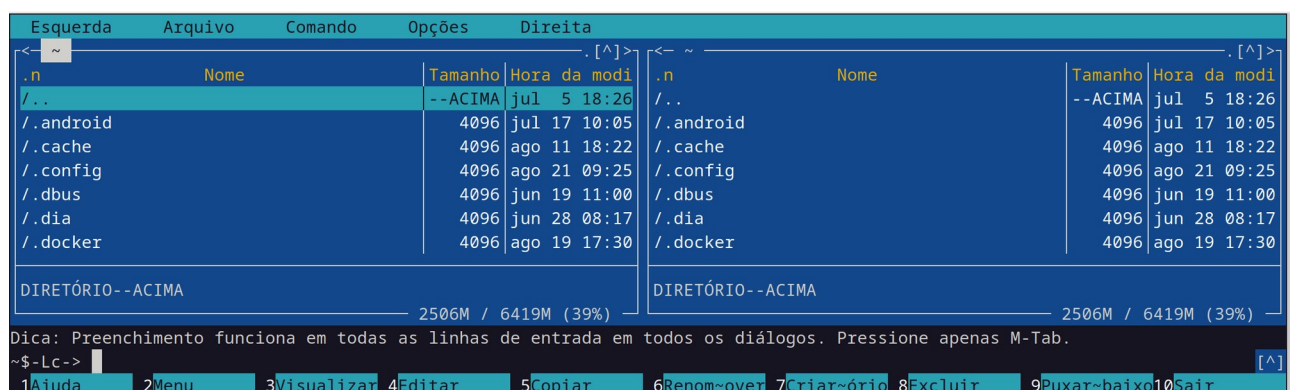
Ctrl+- - reduz o tamanho da fonte

Ctrl+0 - volta a fonte ao tamanho normal. São atalhos do teclado e não do nano

1.3.3 – mc - Gerenciador de arquivos para o terminal

Geralmente instalo este gerenciador nos servidores que administro, pois me ajuda bastante.

sudo apt install mc



```

Esquerda                               . [^]>                               . [^]>
Nome                                     Tamanho Hora da modi                       Nome                                     Tamanho Hora da modi
--ACIMA jul  5 18:26                      --ACIMA jul  5 18:26
/.android                                4096   jul 17 10:05                      /.android                                4096   jul 17 10:05
/.cache                                  4096   ago 11 18:22                          /.cache                                  4096   ago 11 18:22
/.config                                 4096   ago 21 09:25                          /.config                                 4096   ago 21 09:25
/.dbus                                   4096   jun 19 11:00                          /.dbus                                   4096   jun 19 11:00
/.dia                                    4096   jun 28 08:17                          /.dia                                    4096   jun 28 08:17
/.docker                                 4096   ago 19 17:30                          /.docker                                 4096   ago 19 17:30

DIRETÓRIO--ACIMA                          DIRETÓRIO--ACIMA
----- 2506M / 6419M (39%) -----          ----- 2506M / 6419M (39%) -----

Dica: Preenchimento funciona em todas as linhas de entrada em todos os diálogos. Pressione apenas M-Tab.
~$-Lc->
1Ajuda  2Menu  3Visualizar  4Editar  5Copiar  6Renom-over  7Criar-ório  8Excluir  9Puxar-baixo  10Sair

```

Veja que ele tem um menu acima e abaixo mostra seus principais comandos.

Um recurso muito importante é que ele vem com dois painéis, um na esquerda e outro na direita. Assim podemos copiar arquivos da pasta da esquerda para a direita e vice versa. Aceita também o mouse.

Já podemos abrir ele com duas pastas que desejamos

```
sudo mc usr/local/bin homeribafs
```

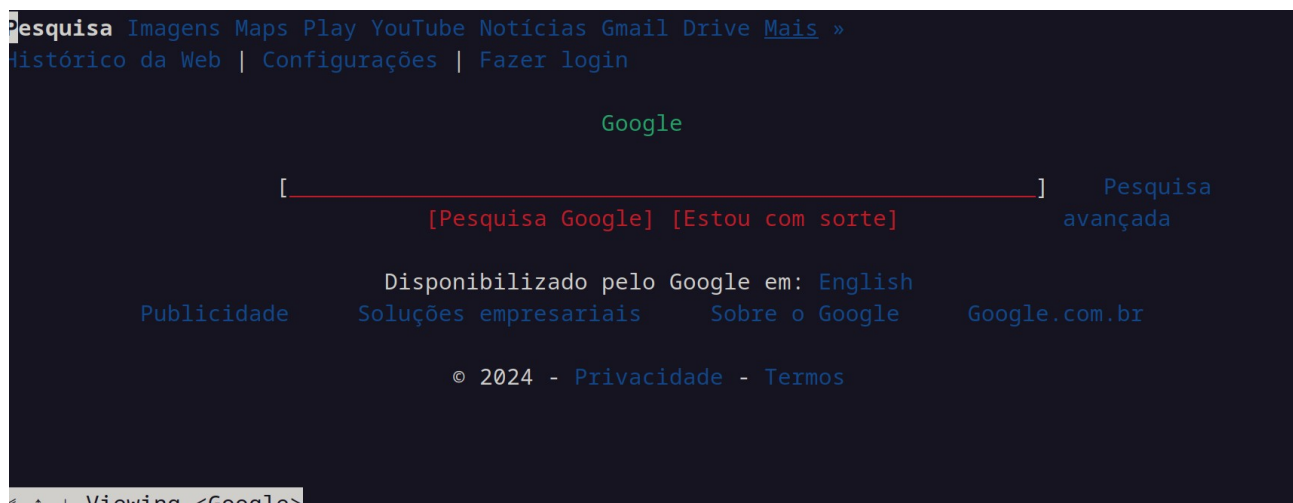
1.3.4 – Navegador para o modo texto

No linux temos alguns navegadores que funcionam no terminal.

Um deles é w3m

```
sudo apt install w3m w3m-img
```

Vejamos ele acessando o Google



Para sair
Shift+Q

1.4 - Permissões de arquivos

As permissões dos arquivos no Linux formam um tópico bem rico e um dos grandes responsáveis pela segurança do sistema operacional.

Temos:

- Permissões para ler, escrever, executar, acessar
- Dono e grupo

Existem muitos bons tutoriais sobre o assunto e vou apenas citar com alguns comentários um script que sempre instalo nos servidores que administro para gerenciar as permissões dos arquivos da pasta web, geralmente em /var/www

No caso eu adiciono o meu usuário ao grupo do Apache ou do Nginx usando

```
sudo adduser ribafs www-data
```

Então crio um arquivo chamado perms em /usr/local/bin

```
sudo nano /usr/local/bin/perms
```

Contendo

```
#!/bin/sh
clear;
echo "Aguarde enquanto configuro as permissões do /var/www/$1";
echo "";
chown -R ribafs:www-data /var/www/$1;
find /var/www/$1 -type d -exec chmod 775 {} \;
find /var/www/$1 -type f -exec chmod 664 {} \;
echo "";
echo "Concluído!";
```

Executando:

Varrer toda a pasta /var/www

```
sudo perms
```

Ao final ele me diz

Aguarde enquanto configuro as permissões do /var/www/

Concluído!

Varrer e ajustar os arquivos da pasta /var/www/site

```
sudo perms site
```

1.5 – Cron

O Cron permite que usuários Linux e Unix executem comandos ou scripts em uma determinada data e hora. Você pode agendar scripts para serem executados periodicamente. Cron é uma das ferramentas mais úteis em sistemas operacionais Linux ou UNIX. Geralmente é usado para tarefas de administrador de sistemas, como backups ou limpeza do diretório /tmp e muito mais. O serviço cron (daemon) é executado em segundo plano e verifica constantemente o arquivo /etc/crontab e os diretórios /etc/cron.*/. Ele também verifica o diretório /var/spool/cron/.

Você precisa usar o comando crontab para editar, criar, instalar, desinstalar ou listar os cron jobs. Cada usuário pode ter seu próprio arquivo crontab e, embora sejam arquivos em /var/spool/cron/crontabs, eles não devem ser editados diretamente. Você precisa usar o comando crontab para editar ou configurar seus próprios cron jobs.

Tipos de arquivos de configuração cron

Existem diferentes tipos de arquivos de configuração:

1. O crontab do sistema UNIX/Linux: Normalmente, usado por serviços do sistema e tarefas críticas que requerem privilégios de root. O sexto campo (veja abaixo a descrição do campo) é o nome de um usuário para o qual o comando será executado. Isso dá ao crontab do sistema a capacidade de executar comandos como qualquer usuário.

2. O usuário crontabs: O usuário pode instalar seus próprios cron jobs usando o comando crontab. O sexto campo é o comando a ser executado e todos os comandos são executados como o usuário que criou o crontab

Nota: Este faq apresenta implementações cron escritas por Paul Vixie e incluídas em muitas distribuições Linux e sistemas semelhantes ao Unix, como na popular 4ª edição BSD. A sintaxe é compatível com várias implementações de crond.

Como faço para instalar, criar ou editar meus próprios cron jobs?

Para editar ou criar seu próprio arquivo crontab, digite o seguinte comando no prompt do shell UNIX/Linux:

```
crontab -e
```

Devo reiniciar o cron após alterar o arquivo crontable?

Não. O Cron examinará o tempo de modificação em todos os crontabs e recarregará aqueles que foram alterados. Portanto, o cron não precisa ser reiniciado sempre que um arquivo crontab é modificado.

Sintaxe do crontab (descrição do campo)

A sintaxe é:

```
1 2 3 4 5 /path/to/command arg1 arg2
```

ou

```
1 2 3 4 5 /root/backup.sh
```

Onde

- 1: Minute (0-59)
- 2: Hours (0-23)
- 3: Day (0-31)
- 4: Month (0-12 [12 == December])
- 5: Day of the week(0-7 [7 or 0 == sunday])
- /path/to/command – Script or command name to schedule

Fácil de lembrar o formato

* * * * * comando a ser executado a cada minuto

Seu cron job é o seguinte para jobs do sistema:

```
1 2 3 4 5 USERNAME /path/to/command arg1 arg2
```

ou

```
1 2 3 4 5 USERNAME /path/to/script.sh
```

Exemplo: execute o script do cron job de backup

Se você deseja que um script denominado /root/backup.sh seja executado todos os dias às 3 da manhã, sua entrada no crontab será semelhante à seguinte. Primeiro, instale seu cronjob executando o seguinte comando:

```
crontab -e
```

Adicione

```
0 3 * * * /root/backup.sh
```

Salve e feche

Mais exemplos

Para executar /path/to/command cinco minutos depois da meia-noite, todos os dias, digite:

```
5 0 * * * /path/to/command
```

Execute /path/to/script.sh às 14h15 do primeiro dia de cada mês, digite:

```
15 14 1 * * /path/to/script.sh
```

Execute `/scripts/phpscript.php` às 22h nos dias de semana, digite:
`0 22 * * 1-5 /scripts/phpscript.php`

Execute `/root/scripts/perl/perlscript.pl` 23 minutos após a meia-noite, 2h, 4h ..., todos os dias, digite:
`23 0-23 / 2 * * * /root/scripts/perl/perlscript.pl`

Execute `/path/to/command` às 5 após 4 de cada domingo, digite:
`5 4 * * sun /path/to/command`

Além disso, o cron lê os arquivos no diretório `/etc/cron.d/`. Normalmente, o daemon do sistema como `sa-update` ou `sysstat` coloca seu cronjob aqui. Como um usuário `root` ou superusuário, você pode usar os seguintes diretórios para configurar tarefas cron. Você pode soltar seus scripts diretamente aqui. O comando `run-parts` executa scripts ou programas em um diretório por meio do arquivo `/etc/crontab`:

Directory	Description
<code>/etc/cron.d/</code>	Put all scripts here and call them from <code>/etc/crontab</code> file.
<code>/etc/cron.daily/</code>	Run all scripts once a day
<code>/etc/cron.hourly/</code>	Run all scripts once an hour
<code>/etc/cron.monthly/</code>	Run all scripts once a month
<code>/etc/cron.weekly/</code>	Run all scripts once a week

Como faço para usar os diretórios acima para colocar meus próprios scripts ou trabalhos?

Aqui está um script de shell de amostra chamado `clean.cache`. Este script é criado para limpar arquivos em cache a cada 10 dias. Este script é criado diretamente no diretório `/etc/cron.daily/`. Em outras palavras, crie um arquivo de texto chamado `/etc/cron.daily/clean.cache` da seguinte maneira.

```
#!/bin/bash
# A sample shell script to clean cached file from lighttpd web server
CROOT="/tmp/cachelighttpd/"

# Clean files every $DAYS
DAYS=10

# Web server username and group name
USER="lighttpd"
LGROUP="lighttpd"

# Okay, let us start cleaning as per $DAYS
/usr/bin/find ${CROOT} -type f -mtime +${DAYS} | xargs -r /bin/rm

# Failsafe
# if directory deleted by some other script just get it back
if [ ! -d $CROOT ]
then
    /bin/mkdir -p $CROOT
    /bin/chown ${USER}:${LGROUP} ${CROOT}
fi
```

Salve e feche e ajuste as permissões

```
chmod +x /etc/cron.daily/clean.cache
```

Como faço backup das entradas de cron jobs instaladas?

Basta digitar o seguinte comando para fazer backup de seus cronjobs em um servidor NAS montado no diretório /nas01/backup/cron/users.root.bakup:

```
crontab -l > /nas01/backup/cron/users.root.bakup  
crontab -u userName -l > /nas01/backup/cron/users.userName.bakup
```

arquivo de registro de tarefas crond e cron

Você pode usar os comandos abaixo para visualizar o arquivo de log crond. Por exemplo, em um CentOS 7 Linux on pode usar os seguintes comandos:

```
cat /var/log/cron  
tail -f /var/log/cron  
grep "my-script.sh"  
tail -f /var/log/cron
```

Descubra se os trabalhos de backup diários estão em execução ou não no servidor FreeBSD Unix:

```
sudo grep '/usr/local/bin/rsnapshot daily' /var/log/cron
```

Numa distribuição Linux moderna, pode-se usar o comando systemctl ou o comando journalctl:

```
sudo systemctl status cron  
sudo journalctl -u cron  
sudo journalctl -u cron | grep backup-script.sh
```

Leia as man pages
run-parts(8)

Referências

<https://www.cyberciti.biz/faq/how-do-i-add-jobs-to-cron-under-linux-or-unix-oses/>

=====
CRONTAB – o agendador de tarefas do linux
4 de janeiro de 2012 By Ubuntu Dicas 15 Comentários

por Rodolfo Silveira

Olá pessoal, estou aqui de volta fazendo um pequeno tutorial sobre o crontab o agendador de tarefas do Linux, lembrando que o crontab existe em qualquer versão do linux.

Com o crontab é possível especificar horários como “todos os dias às 5 da manhã” ou “a cada meia hora”, “de dez em dez minutos”.

Para a maioria das tarefas pouco importa a hora que vai ocorrer mas sim a frequência em que ela vai ser executada, como diariamente ou semanalmente. Para isso já existe 4 diretórios especiais, que basta o administrador botar o script lá dentro, eles já serão executados na periodicidade desejada.

```
/etc/cron.daily diário
/etc/cron.hourly a cada hora
/etc/cron.monthly mensal
/etc/cron.weekly semanal
```

Mas caso você mesmo queira fazer um período específico, com hora e tudo mais, basta editar o arquivo:

```
/etc/crontab
```

Então vamos lá. Escolha um editor de sua escolha, no meu caso o VIM e abra um terminal e digite:

```
sudo vim /etc/crontab
```

Notem no conteúdo:

```
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.
```

```
SHELL=/bin/sh
```

```
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
```

```
# m h dom mon dow user command
17 * * * * root cd / && run-parts --report /etc/cron.hourly
25 6 * * * root test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6 * * 7 root test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6 1 * * root test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
#
```

Notem que já existe algumas entradas justamente dos diretórios especiais, para adicionar sua própria tarefa temos que seguir o seguinte padrão de linha:

```
minuto hora dia mês dia da semana usuário comando
```

Notem que para cada espaço, se avança um campo e os campos seguem o padrão existente que conhecemos, por exemplo no campo mês não pode ter 15 pois temos de 1-12 e o campo da semana usamos de 0 a 7 onde zero é domingo 1 é segunda-feira, 2 terça-feira e assim por diante.

Temos também alguns caracteres que facilitam a vida:

Caractere	Exemplo	Siginificado
Hífen	2-4	intervalo de 2 a 4
virgula	2,4,6,8	os números 2,4,6 e 8
barra	*/10	de dez em dez

asterisco * todas as opções possíveis
Vamos botar a mão na massa:

Imagine que temos uma rotina de backup que comprime o o /home e o /var/log e queremos que esta rotina aconteça diariamente as 23:35;

Se usarmos o tar por exemplo, poderíamos usar duas linhas de comando ou não, por isso recomendo que crie um script e no arquivo do CRON nós vamos apontar para o script.

Criando o script:

```
vim /scriptbkp.sh
```

O conteúdo do script, onde comprime a pasta home e a pasta log no hd externo de modo que o nome do arquivo fique com a data do dia da execução e ainda criando um arquivo de log localizado no /scriptbkplog.log para posterior análise:

```
#!/bin/sh  
tar -cvzf /media/hdexterno/bkphome$(date +%Y_%m_%d).tar.gz /home >> /scriptbkplog.log  
tar -cvzf /media/hdexterno/bkplog$(date +%Y_%m_%d).tar.gz /var/log >> /scriptbkplog.log
```

Salve e saia.

Vamos dar permissão de execução também:

```
sudo chmod 755 /scriptbkp.sh
```

Agora vamos lá no crontab:

```
sudo vim /etc/crontab
```

Adicione a seguinte linha:

```
35 23 * * * root sh /scriptbkp.sh
```

Salve e saia. Pronto, todos os dias da semana, todos os meses, todos os dias do mês, na hora 23 e minuto 35 ele vai executar o script e enviar o backup para o hd externo do exemplo.

Mas e se quisermos a atividade de segunda e sabado as 09:27. A linha ficará assim:

```
27 09 * * 1,6 root sh /scriptbkp.sh
```

Se quisermos toda hora, de 08 as 18 de segunda a sexta:

```
00 8-18 * * 1-5 root sh /scriptbkp.sh
```

Se quisermos fazer mensalmente independente da hora, feche o crontab e mova o arquivo para a pasta especial, com o seguinte comando:

```
sudo mv /scriptbkp.sh /etc/cron.monthly
```

FIM! Se tiverem dúvida podem entrar em contato valeu!

Rodolfo Silveira

@rodolfo_tec

e-mail: ro_dolfo14@hotmail.com

Crédito

<https://www.ubuntudicas.com.br/2012/01/crontab-o-agendador-de-tarefas-do-linux/>

1.6 - Calendário (ncal) e Calculadora (bc)

Existe um ótimo calendário para o terminal

cal ou ncal

Caso não esteja instalado instalar com

sudo apt install ncal

Executar

Mostrar o calendário com o mês atual

cal

```
~$-Lc-> cal
      Agosto 2024
do se te qu qu se sá
           1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31
```

Mostrar o calendário de 08 de 1956

cal 08 1956

```
~$-Lc-> cal 08 1956
      Agosto 1956
do se te qu qu se sá
           1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31
```

Ajuda

cal --help

man cal

1.7 – Aplicativos para o modo gráfico

Atualmente uso a distribuição Debian, na versão 12 mas a distribuição que mais utilizei foi o Linux Mint, onde se pode fazer praticamente tudo sem usar o terminal. Mantenho aqui uma boa relação de aplicativos que já acompanham o Linux Mint. Uma relação invejável:

Seleção de alguns dos inúmeros softwares disponíveis para instalação no Linux Mint 20.1. Atualmente está na versão 22.

Através do Gerenciador de Aplicativos no Modo Gráfico

Selecionei apenas alguns que me chamaram mais a atenção e para o modo gráfico (a maioria)

Acessórios

mc - Midnight Commander. Gerenciador de arquivos para o terminal, com muitos e bons recursos. Já abre com dois painéis, como posso fazer no nemo
Luckybackup, Timeshift e Deja-dup - Softwares para backup
ClamTk - Cliente gráfico para o Clamav (varrer virus de windows em máquinas linux)
Psensor - temperatura hardware
Ksysguard - monitor de processos
Kcalc - Calculadora
Lshw-gtk - Informações sobre configurações de hardware
Kfind, Catfish, Searchmonkey (procura arquivos, inclusive pelo conteúdo e em ODF e OOXML)
Look (procurar string em ODF e OOXML)
Tiled - editor de mapas de tiles
Jstest - Teste e configuração de joystick
Wallch - mudar papel de parede
Nemo, Spacefm, Thunar, Tuxcmd e Doublecmd-qt, tuxcmd, konqueror - Gerenciadores de arquivos
Glogg - Log explorer com qt
Gworldclock - date e hora com timezone
Kdf - info de discos
Grub-customizer
Bomaga - Pré-visualizar impressão
Keyboard-configuration
wget - baixar arquivos e continuar download se cair. Baixa um site completo com camadas, etc

Ciência e Educação

Stellarium - Ver o céu
Gnumeric - Planilha
Tuxpaint - editor de imagens para jovens
Klavaro - tutor de digitação
Tuxtype - jogo de datilografia
Calculator - calculadora científica
Arduino - IDE e bibliotecas
Bibletime - Estudar a bíblia
Solfege - treinamento de ouvido
Maxima, Wxmaxima, Mathomatic, Kálgebra - álgebra

Tuxmath - Jogo de matemática para crianças
Scilab - cálculos numéricos científico
Marble - Mapa do globo
Vym - Mapas mentais
Gelemental - Tabela periódica
Gchenpain - Editor de estruturas químicas
Qgis - info geográficas
Saga - Análise científica
Planets - Simulação de gravidade em planetas
Algebox - intro algoritmos
Avogrado - Modelagem e gráficos moleculares
Scratch - Ambiente de programação apra qualquer idade. <https://scratch.mit.edu/>
Aeskulap - Visualizados de imagens médicas
Grace - Plotagem de gráficos XY (plano cartesiano)
Lybniz - plotar gráficos matemáticos
Biogenesis - simulador de evolução de organismos (vida artificial)
Jelic - desenvolvimento de atividades educacionais multimídia
Julia - Linguagem de programação de alta eprformance
Tilem - emulador de calculadora do Z80
Turtle - Ambiente de programação educacional (logo?)
Retext - Markdown
Sigil - editor de ebooks
Notepadqq - Similar ao notepad++ no linux
Gnome-builder - IDE para programar para Gnome
Aghermann - Pesquisa experimental sobre o sono

Desenvolvimento

Netbeans - IDE para Java, PHP, JavaScript, etc
Codeblocks - IDE para C/C++
Sublime Text
QtCreator
Lazarus, Lazarus-ide - IDE para free pascal
Geogebra - Programa de matemática para estudantes e educadore
Sweethome3d - Software de desenho arquitetônico com inteirores em 2D e visualização em 3D
Pychess - Xadrez
Spyder, Spyder3 - IDE em Python para cientistas
Eric - IDE para Python
Glade - IDE para GTK+
Sqlitebrowser - GUI para o SQLite
GAMBAS 3 - Programação gráfica para linux que usa VB como referência
Kdevelop - IDE para C/C++
Kdevelop-php - IDE para PHP
Devhelp - Criador de help do Gnome
Regexxer - Expressões regulares
Geany, Bluefish - IDE para PHP
Localize - tradução
Qtqr - Gerar e decodificar QRCode
Kimagemapeditor - edição de mapas de iamgem HTML

Escritório

LibreOffice - Pacote office, contendo processador de textos, Planilha, Apresentação, Desenho, Bancos de Dados, Fórmulas Matemáticas, Documento HTML e XML, Etiquetas, Cartões de Visitas, Modelos. Plugins de integração com Colibre. Converte para PDF em um clique

Kate - editor de texto com muitos recursos

jEdit - editor para programação

Frescobaldi - editor musical

Freeplane - mapas mentais

Juffed - editor de textos

Ferramentas de Sistema

Wine - emulador de programas windows

Guake - terminal gráfico?

Synaptic - Gerenciador gráfico de pacotes

Gedit - editor de textos oficial do Gnome

Terminator - terminal com múltiplas janelas

Gufw - Interface gráfica para o firewall UFW

USB-creator-gtk

USB-creator-kde

Boot-repair

Mintdrivers

Galternatives - Configuração para alternativas do sistema

PackageSearch - pesquisa e informações de pacotes

Fontes

Birdfont, Fontforge - Editor de fontes

Fontmanager - gerenciador de fontes

Gráficos

dia e dia shapes - diagramas

VSCoDe

huguin - fotos panorâmicas

Kcolorchooser

OBS-studio, Kdenlive, Kino, Gopshop - editor de vídeos

Editores de imagens: Kolourpaint, Pinta, MyPaint, Tuxpaint, Krita, Gimp, Inkscape, Mtpaint

Darktable - mesa de luz e sala escura virtual para fotógrafos

Shotwell, Kphotoalbum - organizador de fotos

Kcolorchooser, Gpick - color picker

Xchm, Kchmviewer - visualizador de arquivos CHM de help do windows

Librecad - CAD para linux

Kruler - Régua na tela

Mythtv - gravador de vídeo (client e server)

Tupi - Ferramenta de criação de animações 2D

Hasciicam - vídeos como texto?

Qelectrotech - Editor de esquemas elétricos

Internet

Firefox, Chromium, Google Chrome, Opera, Midori, Epiphany
Filezilla, Gftp - transferência segura de arquivos entre desktop e servidor
Pidgin - Cliente gráfico para diversos messengers
Kopete - instant messenger
Transmission - baixar torrents
Wireshark - análise do tráfego da rede
Putty - Cliente de SSH
Uget - gerenciador de downloads
WhatsApp
Quitterss - News feed
Krdc - cliente para desktop remoto
Google Earth
Htttrach, Htttraqt - Copiar sites para o desktop para navegação em off
Owncloud-client - sincronização com o servidor do ownCloud
Skype
Gtk-gnutella - compartilhar arquivos via peer to peer pela rede

Jogos

Wine, Payonlinux
Frozen-bubble - estoure as bolhas (o único jogo que joga as vezes)
Pingus - clone do lemmings

Raciocínio

Gbrainy - jogo de quebra cabeças
Einstein - inspirado no enigma de Einstein
Minetest - caixa de areia em blocos
Gnome-tetravex - Combinar azulejos

Pychess, Gnome-chess, Xboard, Chessx - jogo de xadrez
Minecraft
Ascijump - com arte ascii
Hatari - emulador do atari
Steam
E muitos outros

Multimidia

VLC - player multimidia com muitos recursos
Smplayer, Mplayer, Mpv, Xplayer - player de áudio/vídeo
Audacity, Rosegarden, Sweep - Editor de áudio multiplataforma
Cheese - Capturar fotos e vídeo da webcam
Spotfy
Multitube, Smtube - Navegador de vídeos do Youtube
Kazam, Vokoscreen, SimpleScreenRecorder - Filmar o desktop com áudio
Tuxguitar, Nted - editor de tablaturas para violão e guitarra
Aegisub - editor de legendas
Photofilmstrip - Criador de slides show

Hidrogen-drumkits
Espeak - Sintetizador de voz multi-idioma

Flatpak

Telegram
Gimp
Discord
Video Downloader

Editores de vídeo:

Kdenlive
Shotcut
Olive

Krita
Postman
Visual Studio Code
Remmina - cliente de desktop remoto
Minitube
Peek - gravador de tela que gera gif animado
WPS Office
Spotify
Slack
OpenTodoist
Avidemux
Retroarch
Vorta - backup
Calibre - criação de ebook
VueScan - mais de 6000 scanners
Snes9x - emulador de super nintendo
Freetube
GnuCash - Gerenciamento de finanças, contas e investimentos
Tor
Master PDF Editor
Flameshot - screenshots
Onlyoffice Desktop Editors
OpenShot
Marker - markdown
Minder
Android Studio
IntelliJ IDEA Community
Scribus
Gnome Mapas - mapa do mundo
Tarefas - gerenciador de tarefas
Jitsi Meet - Vídeo conferência
Kolourpaint
OpenBoard - quadro branco para escolas
EasySSH
Itopia Remote Desktop Client

Draw.io - criar e compartilhar diagramas
Unity HUB
Warpinator - compartilhar arquivos pela rede
RedNotebook - diário e jornal
Arduino IDE
Sequeler - cliente SQL
Spice-up - criar apresentações
WebStorm - IDE para JS
Pixelorama - Editor de sprites 2D com suporte a animação
Sweet Home 3D
VSCodium
Scratch
DBeaver Community - Universal database manager

Qelectrotek
Calibre - editor de ebooks
Shutter - Captura de tela com muitos recursos
Scribus
vlc - player de multimídia com muitos e bosn recursos
Gimp - Editor de imagens que é o mais perto do Photoshop para linux
Inkscape - Desenho vetorial, como o Corel
audacity - editor de áudio
Aa3d - arte ascii e Ansilove
Blender - Modelagem e renderização 3D

Wine
Synaptic
Terminator - terminal que divide a janela na vert e hor
Gufw
USB-creator-gtk
USB-creator-kde
Boot-repair
PackageSearch - pesquisar pacotes e infos

Firefox
Opera
Filezilla
Pidgin
TRansmission
Wireshark
Putty
Uget
Midori - navegador leve e rápido
WhatsApp
Krdc - Cliente de conexão remota do desktop
Google Earth
Httrach
Owncloud-client
Google-chrome
Skype

Gtk-gnutella - compartilhar arquivos
Httraqt
Kimagemapeditor

Jogos

Frozenbubble
Steam
Gnome-sudoku
Tuxtype
Tuxmath
Minecraft
Gnome-chess
Flight-of-the-amazon-queen
Einstein
Xboard
Asciijump

Multimídia

Cheese
Kazam
Vokoscreen
SimpleScreenResorD
Tuxguitar
Kino
Pitivi
Nted - editor de partituras
Timidity-daemon
Espeak - sintetizador de voz

Preferidos

Xed - meu editor de texto puro/código preferido. Pela sua leveza. Para mais recursos uso o VSCode
VSCode - Muitos e bons recursos. Similar as IDEs Eclipse e Netbeans, mas muito mais leve e com uma infinidade de extensões
LibreOffice - Pesado mas bom, para trabalhos mais elaborados
Firefox - Meu navegador preferido. Acabo de instalar o Edge para sites que são compatíveis somente com ele
Postman - Preferido para APIs
Apache, PHP e PostgreSQL
CakePHP
Joomla
Virtualbox
Vagrant
Github
Docker
SO - Linux
Distros Server - Alpine, Debian e Ubuntu
Distro Desktop - Linux Mint

Capturador de telas - Shutter ou gnome-screenshot --interactive
Editor de imagens - Koorpaint
Gravar desktop - Kazam
SearchMonkey

Sugestão: caso não tenha encontrado algum programa nesta relação para uma específica atividade sugiro que instale o Linux Mint última versão e verifique sua lista de softwares disponíveis para instalação pelo modo gráfico (pois a lista acima não é completa). Existem também outras distribuições linux que trazem uma enorme lista de programas para instalação.

1.8 – Diferenças entre Linux e Windows

Aqui apenas relato as diferenças que eu percebi.

Especialmente no que tange à programação para PHP

A comparação foi usando Windows 10 e Linux Mint 20.1

Para quem nunca usou os dois ambientes não tem como comparar, apenas acredita ou não.

Me parece que quem programa em PHP, cuja grande maioria usa Windows, deveria pelo menos experimentar um Linux, pois a maioria dos servidores web usa Linux, sem contar que aprender um novo sistema operacional valoriza o profissional

Comparação

- Windows é um sistema operacional pago enquanto Linux é gratuito e de código fonte aberto
- Operações com arquivos são sensivelmente mais lentas em Windows
- As operações com internet são um pouco mais rápidas no Linux
- As atualizações do windows algumas vezes ocupam o computador com exclusividade por muito tempo, nos impedindo de usá-lo. E não devemos interromper a atualização desligando ou reiniciando o computador, sob pena de correr o risco de nem mais iniciar
- A usabilidade em geral no Linux Mint é superior a do Windows. Diversos aspectos foram pensados pelos criadores para facilitar a vida do usuário. Um dos mais importantes é o Mint vir com um menu como vinha no Windows 7. Isso ajuda a encontrar todos os programas gráficos instalados sem precisar lembrar seu nome.
- O Windows 10 está vindo com o WSL, que se propõe a oferecer uma distribuição linux no prompt. Mas isso ainda está longe de ser algo usável. Atualmente é mais produtivo usar um pacote como o Xampp.
- O Windows 10 é gordo e pesado exigindo um computador mais caro e robusto do que o necessário para usar o Linux Mint 20.1. E isso é por que o Linux Mint já vem com muita coisa tornando o seu uso mais produtivo. Existem diversas distribuições Linux ainda mais leves que o Mint.
- Para se desinstalar um programa no Linux Mint apenas abrimos o menu, mostramos o nome do programa e clicamos nele com o botão direito - Desinstalar e confirmamos.
- A maioria dos softwares que instalamos no Linux vem dos repositórios oficiais (que ajuda na segurança), que torna a vida do usuário bem mais prática tanto no terminal quanto no gerenciador de aplicativos. Isso é algo que não deve interessar para a MS, tendo em vista que ela nunca adotou algo do tipo.

- Como o Linux tem código fonte aberto já temos centenas de distribuições (<http://distrowatch.org/> e https://en.wikipedia.org/wiki/List_of_Linux_distributions), com usos específicos para diversas necessidades, como educação, segurança, servidores, leves, etc. Qual o sistema operacional é oriundo do Windows? Não pode, pois seu código é fechado. Ele é estéril.
- Um ponto forte do Linux é a liberdade de customizações.
- O desenvolvimento em comunidade, o fato de ser gratuito e de código fonte aberto possibilitam a correção de bugs e de falhas de segurança com mais agilidade.
- Como o Windows tem quase todos os recursos no modo gráfico torna o uso do sistema muito fácil. O Linux até já conta com distribuições que focam no modo gráfico, como o Mint, mas continua tendo muitos recursos importantes no terminal o que torna o usuário que o domina alguém com um perfil geralmente superior ao usuário comum do Windows. Estes e mais outros motivos criam uma comunidade de usuários mais ativa e inclusive maior que a comunidade de usuários do Windows.
- Como em termos de sistema para desktop o Windows não tem concorrente, a MS não se preocupa em melhorar alguns aspectos do sistema. Melhora apenas o que interessa para ele.
- O Linux evoluiu tanto que atualmente chega a detectar e instalar automaticamente drivers do hardware ainda mais que o Windows.
- Somente o fato de ser o sistema mais usado já seria suficiente para atrair sobre si a atenção dos crackers, mas não é somente isso que o torna o sistema menos seguro.
- Quando se abre o Windows explorer (gerenciador de arquivos) muitas vezes ele demora muito para ficar disponível para uso, pois fica scanneando o sistema de arquivos. Para agilizar eu fecho e abro novamente.
- Tanto para ligar quanto para desligar o Linux Mint 20.1 é mais rápido que o Windows 10. Deve ser por conta da gordura em excesso. É como se a MS fosse parceira dos fabricantes de hardware e forçasse um upgrade.
- A instalação do Linux é de longe mais simples e rápida que a do Windows.
- Algo marcante no Linux é a estabilidade e robustez. Praticamente não se vê um Linux travado.

Seria ingenuidade afirmar que o Linux é o sistema operacional ideal para todos. Ele até já pode ser usado por todos, atendendo muitas das necessidades, mas algumas necessidades o Linux ainda não atende como o Windows: alguns jogos pesados, programas de edição de imagens como o Photoshop, CorelDraw, AutoCAD, alguns de edição de vídeo. Estas áreas ainda prendem usuários no Linux. O Linux já evoluiu muito nestas áreas mas ainda falta evoluir para conseguir atender a todos os usuários. O que acontece é que os fabricantes destes programas ainda não se voltaram para o Linux, que ainda tem poucos usuários em relação aos do Windows.

Me parece que todos os 3 principais sistemas para desktop, Windows, Mac e Linux caminham para um sistema comum, com recursos dos 3. Ainda temos um novo sistema surgindo, que é o ChromeOS. Algo que mostra esta tendência é a implantação do WSL, que tenta atrair usuários do Linux.

Algumas pequenas diferenças (Testar no Windows 10):

- Selecionar um arquivo no gerenciador de arquivos e teclar F2 para renomeá-lo (veja os qual sistema pensa mais no usuário, facilitando sua vida):
 - Linux Mint 20.1
 - Windows 10

1.8 - Recuperando um linux do windows

Algo que aconteceu comigo e acredito vale o relato.

Ontem desinstalei alguns programas do meu Linux Mint 20.1 com cinnamon através do menu. Botão direito sobre o programa - Desinstalar. Desinstalei vários que não estava precisando para aliviar o HD.

Lembro que um deles pediu confirmação para remover um pacote com cinnamon. Confirmei sem prestar muita atenção.

Achei estranho o fato de após desinstalar dezenas de programas o espaço livre em disco não alterar.

O problema foi quando reiniciei o computador. Ele me avisa que não encontrou o cinnamon e numa mensagenzinha como aqueles window manager bem antigos e sem qualquer menu. Apenas com OK e Cancel. Desliguei e fui dormir, já me preparando para reinstalar tudo. Sei que não existe sistema perfeito mas me parece que o Mint não deveria ter permitido que o seu ambiente gráfico fosse removido. Embora saiba que eu assinei em baixo mas tudo bem. Talvez qualquer outro distro tenha o mesmo comportamento. Estes acontecimentos me fazem refletir e repensar algumas coisas, como a seguir.

Então me veio a mente um comentário de um colega sobre a estabilidade do Debian. Será que é hora de sacrificar todo o conforto do Mint pela estabilidade e segurança do Debian? Passei algum tempo refletindo sobre isso. Pela manhã, ao acordar, ainda na cama, pensei isso "cada distro tem seus recursos, características e um perfil de usuário que a usa". O Mint é bem adequado para usuários iniciantes no Linux, pois é bem amigável e com muitos recursos no modo gráfico. Ideal para usuários que vem do Windows mas não somente. Já o Debian requer um pouco mais de conhecimento e de preferência um usuário com boa experiência com linux e o terminal. Talvez para ter maior estabilidade e segurança precise abdicar de alguns recursos de que gosto, ou melhor, consegui-los de outra forma não tão confortável. Talvez. O Mint vai continuar no meu coração mas talvez seja o momento de dar uma chance ao Debian, que há muito tem minha admiração e respeito.

Podemos perceber pelo instalador. O do Debian pede a senha do root além da do user comum. A do Mint/Ubuntu somente do user comum. Sempre que uso Ubuntu eu atribui senha para o root pois já tive problema por não ter esta senha. Numa recuperação do sistema somente com senha de root.

Levantei e fui para o computador dos filhos, que tem o Windows. Nem lembrei que ele também tem o Mint em dual boot. Antes disso liguei o meu notebook e deixei ele chegar até aquela pequena mensagem. No computador dos filhos fui acessar o Facebook, mas não lembrei a senha, que está salva em meu notebook e que também tenho em um arquivo texto nele. E agora? Acessei o git que havia instalado e executei

```
ssh ribafs@192.168.0.xx
```

Uma maravilha. Lá estava eu, no meu notebook. O sistema estava inteiro, ou melhor, faltando apenas o ambiente gráfico, mas todo o restante funcionando. Então verifiquei no arquivo das senhas, mesmo sem abrir:

```
cat /backup/Senhas.txt | grep face
```

Ele me mostrou e eu acessei o Facebook.

Resolvi restaurar o grub do notebook via ssh. Restaurei e reiniciei mas não era o grub. A mensagem que eu recebia já havia passado pelo grub.

Foi quando lembrei do cinnamon, então, via ssh, executei no notebook:

```
sudo apt update
```

```
sudo apt install cinnamon
```

Bem, estou agora digitando este texto em meu notebook e no Linux Mint 20.1 com cinnamon e tudo. Legal, não?

Claro que tudo isso poderia ser feito de qualquer outro sistema operacional, inclusive de um celular com um aplicativo com SSH. Acabo de instalar um aqui num celular com Android para testar e é uma beleza. Vale muito a pena ter o SSH, este canivete suíço, instalado em seu celular. Sugestão "Mobile SSH". É pequeno, simples, configurável e resolve.

Mas o LMDE 4 está sendo baixado no computador dos filhos. Vou usar o rufus para instalar num pendrive e dar um crédito ao Debian e ainda vou usar algo do Mint e com cinnamon, usando o LMDE. Afinal gostaria de ficar com o melhor dos dois mundos e vou tentar. Bola pra frente.

Como a versão atual está bem mais lenta que antes, isso me faz lembrar que o Debian é bem mais leve que os Ubuntu.

Escrevi este artigo usando o Linux Mint 20.1 com cinnamon, mas esta linha já estou encrevendo com o Xed no LMDE 4. Muita coisa fica realmente mais ágil, com melhor desempenho que no Mint, principalmente se for o 20.1 com cinnamon. Uma beleza, ganha-se estabilidade e desempenho, sem perder o Cinnamon e parte do Mint. O melhor dos 3 mundos. :) Tem mais, a esposa nem vai perceber que mudei de sistema, alias, é muito difícil, mesmo para quem entende, perceber diferenças.

Muita coisa do que eu tinha no Mint 20.1 continua disponível no LMDE 4, com pouca diferença: o Mint traz o driver da minha impressora (Canon G3110), o LMDE não traz mas uma rápida busca e ok. O Mint 20.1 traz o PHP 7.4 (trabalho com PHP) e o LMDE 4, que usa Debian 10, traz o 7.3. Eu preferia o 7.4, mas o 7.3 vai me atender. E mais algumas poucas diferenças, mas agora estou melhor. Pelo menos mudei e estou com algo novo (igual criança por novidade).

Ainda bem que demorei alguns dias para publicar este relato, pois agora me apareceu um problema muito grande: o som não está prestando. Baixei o Debian e estou criando uma VM para testar. Instalei com Gnome e Cinnamon. Embora pretenda usar o Cinnamon, ele usa algo do Gnome. Se o som ficar legal irei instalar o Debian puro. Imagino que o desempenho será ainda melhor. Não terei um look bonitão, mas tá bom.

Ufa, anets mesmo de testar o Debian na VM percebi que o problema no som era apenas no fone de ouvido. Beleza. Mas mesmo assim vou fazer uns testes com o Debian.

O desempenho não dá para ser percebido numa VM, a não ser que concedamos bons recursos para ela. Ideal é criar uma partição para ele para boot. Mas não preciso fazer isso, pois já fiz isso algumas vezes e sei que a performance do Debian é superior a do Ubuntu e Mint.

Fatalidade

Algo me empurrou para o Debian puro. Eu removi meu HD com o Mint e coloquei outro que estava planejando comprar. Como não deu certo coloquei o meu de volta. Então veio o empurrão, que até agora não entendi. Sem eu mexer em nada no HD ele perdeu o grub. Como o outro não foi imediatamente detectado pela BIOS, ela ficou fazendo uns testes. Talvez tenha acontecido alguma alteração na BIOS que não enxergou o Grub. RESumindo: agora já estou terminando este artigo usando o Gedit no Debian. Preferia estar usando o Xed do Mint, mas ele já estava ficando bem pesadinho. Então é melhor me acostumar e ficar por aqui, pois também não quero ficar usando muito do meu tempo apenas trocando de distro.

Porque Ubuntu ganha terreno

Durante a instalação do Debian, na fase da criação das partições, que sempre escolho manual para criar:

- UEFI - 1GB
- swap - 4GB
- / - uns 60GB
- /backup - restante

Estou com um HD bem pequeno, apenas 120GB. Apenas porque é SSH. De vez em quando tenho que estar apagando coisas e até formatando e instalando tudo novamente, para aliviar. Hoje estarei indo comprar um maior para acabar com isso. Agora, após instalar o Debian, instalar o AMP, alguns softwares e configurar como geralmente faço, estou com 45GB livres dos 60. Não lembro quanto restava livre no Mint após instalar, mas logo eu estava com 20 e até 6GB livres e isso me incomoda não ter espaço para mexer. A questão aqui é ser SSD, pois tenho um de 320, mas SATA e não aguento, depois de provar o SSD. O original SATA de 1TB queimou. Sem pró. Até mais tarde.

Apesar de já ter feito isso, instalar Debian, algumas vezes, quase desisti da instalação, pois não estava lembrando como proceder. Nem lembrava mais que durante esta fase ele usava o Gparted. Já estava ficando viciado com a facilidade do instalador do Ubuntu e o do Mint é ainda melhor. Isso justifica que a cada dia o Ubuntu e derivadas ganha ainda mais vantagem em relação ao Debian e a outras. Sabe o que é sintomático e ainda mais grave? É que em minha última pesquisa sobre distros voltadas para servidor, vi que o Ubuntu é o mais popular e disparado em relação ao segundo. Para servidor ao meu ver não deveria, mas mostra que praticidade, facilidade conquista.

Realmente o Gedit abre bem mais rápido no Debian que o Xed no Mint. Que fique registrado, pois tomara que alguém do Mint escute e mude isso.

Caramba, se eu lembrasse que a diferença de desempenho era tanta e em praticamente tudo (até agora exceto com libreoffice, que continua pesado. Paciência) eu já teria mudado há muito. Existem pequenas diferenças que estou contornando, depois passarei detalhes. Espero me segurar, pois detendo ficar esperando pelo computador. Até o boot é sensivelmente mais rápido. Agora, olhei agora para a barra de tarefas abaixo e neste aspecto, look e na usabilidade o Mint ganha disparado. Aquela história, precisamos ter consciência sobre o que é mais importante para nós.

Esperarei alguns dias, até que tenha instalado e testado tudo que uso para somente então publicar este artigo, com a apreciação final.

Como todas as mensagens são moderadas no grupo Mint Brasil não acredito que este artigo seja liberado. Pelo menos integralmente, não. Então cortarei e publicarei somente até a parte do LMDE.

Que reviravolta. Comecei este artigo no Linuc Mint, mudei para o LMDE, depois para o Debian e agora estou no Ubuntu 20.10. Pequeno esclarecimento, pois pretendo continuar usando o Ubuntu. De fato, o Mint com Cinnamon foi a distribuição mais completa e confortável que já usei. Mas aconteceu um problema, além dela estar muito pesada e isso me fez refletir. Refletindo fui parar no LMDE, que é ainda um Mint com Cinnamon, com todos os seus ótimos recursos e look mas não usando o Ubuntu e sim o Debian. Mas algo me levou a ir em frente e fui para o a Debian puro, na versão 10. Tive que reinstalar e me deu bastante trabalho. Enquanto isso, fiz algumas pesquisas, assisti um vídeo do Diolinux sobre algumas distros e isso me levou a usar o Ubuntu. Ele não tem o desempenho do Debian mas é bem superior ao do Mint. Veja o que me levou a usar o Ubuntu. Fiquei pensando, se não existisse o Mint nem o Ubuntu e todos tivessem que usar o Debian. Ao meu ver bem poucas pessoas usariam Linux, pois o Debian exige um perfil de usuário mais experiente. Isso mostra realmente que o Ubuntu entrega um ótimo sistema e é democrático, pois é muito amigável. Olha o que me veio a mente agora, enquanto escrevia isso: o Mint é ainda mais amigável que o Ubuntu, inclusive estou usando pequena parte dele, que é o Cinnamon, mas ele também está deixando de ser democrático e exige um computador mais potente que o Ubuntu. Independente disso, o Ubuntu foi quem desencadeou tudo isso. Se não fosse ele, atualmente estaríamos usando o Debian ou um dos descendentes do RedHat. Praticamente temos duas grandes raízes: Debian e RedHat.

O interessante é que este raciocínio me mostrou que nem sempre o melhor é entregue por uma organização sem fins lucrativas mas pode sim ser entregue por uma empresa, como é o caso da Canonical. Lembrando, se a empresa realmente for idônea ela tem mais recursos financeiros para bancar as ações que uma organização. Termine citando a Wikipédia, outra grande democrata:

"A proposta do Ubuntu é oferecer um sistema que qualquer pessoa possa utilizar sem dificuldades, independentemente de nacionalidade, nível de conhecimento ou limitações físicas. O sistema deve ser constituído principalmente por software livre e deve também ser isento de qualquer taxa.

O nome "Ubuntu" IPA: [u'buntu] deriva do conceito sul africano de mesmo nome, diretamente traduzido como "humanidade com os outros" ou "sou o que sou pelo que nós somos todos".

"Uma pessoa com Ubuntu está aberta e disponível para outros, apoia os outros, não se sente ameaçada quando outros são capazes e bons, baseada em uma autoconfiança que vem do conhecimento que ele ou ela pertence a algo maior e é diminuída quando os outros são humilhados ou diminuídos, quando os outros são torturados ou oprimidos."

Esse nome busca passar a ideologia do projeto, baseada nas liberdades do software livre e no trabalho comunitário de desenvolvimento. O sistema é muito comumente chamado "Ubuntu Linux", porém, oficialmente a Canonical, desenvolvedora do sistema, usa apenas o nome "Ubuntu", uma vez que o sistema ao ser portado para outros núcleos livres para além do Linux recebe outros nomes (por exemplo, o Ubuntu implementado sobre o OpenSolaris recebe o nome de "Nexenta") - ao contrário do Debian, por exemplo, que recebe este nome independentemente do núcleo usado."

Isso fechou. Essa filosofia, esta generosidade me conquistou. É esta distro que eu quero usar e devo apoiar.

O default do Ubuntu é bem bonito, bem mais que o Cinnamon nele, mas o Cinnamon é mais prático. Não preciso saber de cor o nome dos programas instalados. Apenas clico no menu e digito o nome do programa ou parte e ele já aparece. Além de alguns recursos que ele traz.

3 - Shell Scripts

Existem diversos interpretadores de comando usados nos terminais do Linux. O mais popular é o Bash, que estarei abordando aqui;

O Bash contém recursos que aparecem em outros shells populares e alguns recursos que só aparecem no Bash. Alguns dos shells dos quais o Bash pegou conceitos emprestados são o Bourne Shell ('sh'), o Korn Shell ('ksh') e o C-shell ('csh' e seu sucessor, 'tcsh'). O menu a seguir divide os recursos em categorias, observando quais recursos foram inspirados por outros shells e quais são específicos do Bash.

Estes interpretadores de comando tem muitos recursos de programação, como estruturas condicionais, como if, else, laços for e muito mais.

Não pretendo ensinar sobre a criação de scripts com bash, primeiro por que não sei. Os scripts que crio são sob demanda. Quando preciso procuro algo semelhante ou então começo e saio tirando as dúvidas.

Scripts

Comecei criando scripts e colocando na pasta /usr/local/bin para poder acessar de qualquer lugar. Especialmente porque aprendi a passar parâmetro para os scripts. Sabe como passo um parâmetro para um script? Assim

```
nano criadir.sh
```

```
#!/bin/bash
# Criar pastas na pasta do user
mkdir $1
```

Ao executar este script ele espera o nome da pasta como parâmetro:

```
criadir.sh nomepasta
```

Então o primeiro parâmetro passado é o \$1. Se quiser passar dois devem ser \$1 e \$2.

3.1 - aliases

Depois descobri que o alias aceita parâmetro, somente um, mas aceita e precisa estar no final do alias.

Usando o exemplo acima para criar um alias

```
cd /home/ribafs
```

```
nano .bashrc
```


Ao final adiciono:

```
alias criadir="mkdir"
```

```
source .bashrc
```

Executando

```
criadir nomepasta
```

O alias aceita somente um único parâmetro e ao final do alias. Quando preciso de mais de um parâmetro então uso um script e o crio em /usr/local/bin e depois dou permissão de execução.

Usuários do Windows

Atualmente existe a opção de usar o linux no WSL2. Para aqueles cujo computador não suporte o WSL2 existe a alternativa dos arquivos de lote/batch.

Veja os meus aliases que ficam ao final do .bashrc.

```
# Meus aliases
alias c="clear"
alias maiores='du -cks * | sort -rn | head'
alias rc="nano /home/ribafs/.bashrc"
alias rcs="source /home/ribafs/.bashrc"
alias apti="sudo update;sudo upgrade -y;sudo apt install -y"
alias aptr="sudo apt remove --purge -y"
alias apts="sudo aptitude search"
alias apr="sudo service apache2 restart"
alias phpI="sudo nano /etc/php/8.2/apache2/php.ini;apr"
alias up="sudo apt update;sudo apt upgrade -y"
alias ni="npm install;npm run build"
alias ci="composer install"
alias l11="composer create-project --prefer-dist laravel/laravel"
alias l10="composer create-project --prefer-dist laravel/laravel:^10.0"
alias l9="composer create-project --prefer-dist laravel/laravel:^9.0"
alias as="php artisan serve"
alias am="php artisan migrate"
alias ams="php artisan migrate --seed"
alias ak="php artisan key:generate"
alias a="php artisan"
alias ce="cp .env.example .env"
alias lc="php artisan cache:composer du;clear;php artisan config:cache;php artisan route:cache;php artisan optimize"
alias ne="nano .env"
alias lr="mysql -uroot -e 'drop database laravel';mysql -uroot -e 'create database laravel';echo database laravel recreated;"
alias p="php -S localhost:8000"
```

Isso facilita muito o meu trabalho, especialmente porque faço muitos testes, muitas instalações.

3.2 – Exemplos de scripts

3.2.1 – Ajustando o .env

O laravel 11 cria um aplicativo e no .env usa o sqlite e as demais linhas da conexão comentadas, Assim

```
DB_CONNECTION=sqlite
# DB_HOST=127.0.0.1
# DB_PORT=3306
# DB_DATABASE=laravel
# DB_USERNAME=root
# DB_PASSWORD=
```

Eu quero que fique assim:

```
DB_CONNECTION=mariadb
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=laravel
DB_USERNAME=root
DB_PASSWORD=
```

E quero que um script faça isso para mim, já que um alias não resolve. Pesquisei e testei bastante até chegar a este script, que chamei originalmente de addmaria e depois apenas ma:

```
#!/bin/bash
sed -i 's/DB_CONNECTION=sqlite/DB_CONNECTION=mariadb\nDB_HOST=127.0.0.1\nDB_PORT=3306\nDB_DATABASE=laravel\nDB_USERNAME=root\nDB_PASSWORD=/g' .env
sed -i.bak '/# DB_HOST=127.0.0.1/d' .env
sed -i.bak '/# DB_PORT=3306/d' .env
sed -i.bak '/# DB_DATABASE=laravel/d' .env
sed -i.bak '/# DB_USERNAME=root/d' .env
sed -i.bak '/# DB_PASSWORD=/d' .env
```

O sed é uma ótima ferramenta para procurar e sobrescrever strings.

3.2.2 – Backup incremental

Há muitos anos que faço backup dos meus arquivos em uma partição de backup. Mantenho sempre uma partição para o sistema e outra para o backup dos meus arquivos. Uma boa medida, mas já aconteceu várias vezes de acidentalmente apagar tudo.

Atualmente, como tenho um HT USB externo, resolvi fazer algo decente. Manter backup de tudo que tenho no HT externo, mas desta vez fazer algo melhor, com rsync fazendo backup incremental. Na primeira cópia ele copia tudo, é full, mas nas próximas ele copia somente o que mudou. Numa nova versão eu resolvi que se remover algo no original que ele remova também no backup.

```
#!/bin/bash
rsync -avh --progress /backup/usb/ /media/ribafs/HDBackup/backups/ --delete
# -avh is for --archive --verbose --human-readable
```

Apenas espeto o HD no micro e executo

```
df -h
```

E pego o endereço onde o HD foi montado para usar no final do script.

Uma beleza, uma simples linha e faz essa mágica.

Tomo o cuidado de geralmente quando faço mudanças no disco do micro executo uma atualização do backup.

Outro muito bom é o backup do meu diário, aplicativo criado com Laravel para isso. O backup está no crontab e roda a cada dia as 12 horas.

```
#!/bin/bash
zip -rq /backup/usb/pessoais/diario/diario_$(date +"%Y_%m_%d").zip /backup/usb/www/diario
mysqldump -uroot diario diarios > /backup/usb/pessoais/diario/diario_$(date +"%Y_%m_%d").sql
```

```
crontab -e
```

Adicionar ao final:

```
0 2 * * * /backup/usb/pessoais/diario/diario.sh
```

3.2.3 – Agenda com Shell Script

Uma boa aplicação do shell script e que é útil, uma agenda. Composta de dois arquivos, o script agenda.sh e o banco de dados, que é o agenda.txt

agenda.sh

```
#!/bin/bash
clear
echo ===== Agenda de Telefones e E-mails - Ribamar FS =====
echo ===== a - Adicionar registros =====
echo ===== c - Listar a Agenda =====
echo ===== s - Sair =====
read opcao
if [ $opcao = "a" ]; then
# Entrar com os dados
    echo Entre o nome
    read nome
    echo Entre com o e-mail
    read email
    echo Entre com o telefone ou ramal
    read telefone
    # Gravar os dados no arquivo agenda.txt
    echo "$nome - $email - $telefone" >> agenda.txt
    # Mostrar o arquivo na tela
    less agenda.txt
elif [ $opcao = "c" ]; then
    less agenda.txt
elif [ $opcao = "s" ]; then
    clear
    exit 0
fi
```

Executar com

sh agenda.sh

```
===== Agenda de Telefones e E-mails - Ribamar FS =====
===== a - Adicionar registros =====
===== c - Listar a Agenda =====
===== s - Sair =====
a
Entre o nome
Ribamar FS
Entre com o e-mail
ribafs@gmail.com
Entre com o telefone ou ramal
00000000
```

Agora veja o agenda.txt

Nomes

E-mails

Telefones

Ribamar FS - ribafs@gmail.com - 994407071
Ribamar FS - ribafs@gmail.com – 994405059

A primeira linha escrevi manualmente e em cada linha seguinte ele coloca um contato.

3.2.7 – Maiores arquivos do diretório atual

Apenas destacando este que para mim tem forte utilidade. Apenas acesso uma pasta pelo terminal e digito

maiores

```
/backup/usb/DevOps/GitHub$-Lc-> maiores
6087220    total
2403196    material
662732    javascript-main.zip
531044    gdevelop-main.zip
263856    laravel-projetos
240060    aplicativos-main.zip
209272    blog-main.zip
154776    joomla-main.zip
141436    postgresql-main.zip
130348    laravel-book-master.zip
```

Ele mostra o total da pasta e os arquivos listados pelo tamanho do maior para o menor

3.2.8 – Instalação do servidor web

Para finalizar vou mostrar um script que uso para instalar todos os pacotes do meu servidor web, nginx, php, mysql e cia. Iniciei criando um script usando a biblioteca semi-gráfica dialog, depois outros a abaixo são apenas comandos:

```
sudo apt install aptitude git composer mc curl phpunit ssh imagemagick zip unzip -y
sudo apt install php-fpm php-mysql php-gd php-cli php-curl php-mbstring php-zip php-opcache
php-xml php-bcmath php-pear php-imagick php-tidy php-xmlrpc php-intl php-xdebug php-apcu
php-redis -y
```