



FUNDAMENTOS DE DESIGN DE SISTEMAS

AULA 1



Prof. Vinicius Pozzobon Borin



CONVERSA INICIAL

O objetivo desta etapa é ensinar-lhe os princípios básicos sobre os sistemas operacionais baseados em Linux. Vamos explorar tópicos como histórico, distribuições, além de aprender serviços e ferramentas essenciais. Não importa se você é novo no mundo do Linux ou se possui alguma experiência prévia, pois este curso tem como objetivo ajudá-lo a se sentir confortável e confiante ao trabalhar com esse sistema operacional poderoso.

TEMA 1 – LINUX

Linux é um sistema operacional de código aberto criado por Linus Torvalds em 1991. Linus estudava na Universidade de Helsinque e precisava de um sistema operacional para seu computador pessoal. Ele decidiu criar o seu próprio sistema, inspirado no Unix, e chamou-o Linux.

Linus começou a trabalhar no projeto como um *hobby*, mas logo outros desenvolvedores se juntaram a ele e o projeto se expandiu rapidamente. O *kernel* do Linux foi lançado como *software* livre, o que significa que qualquer pessoa pode ver, modificar e distribuir o código fonte. Essa abertura permitiu que outros desenvolvedores contribuíssem para o projeto e ajudou a tornar o Linux uma das principais opções de sistema operacional para servidores e computadores pessoais até os dias atuais.

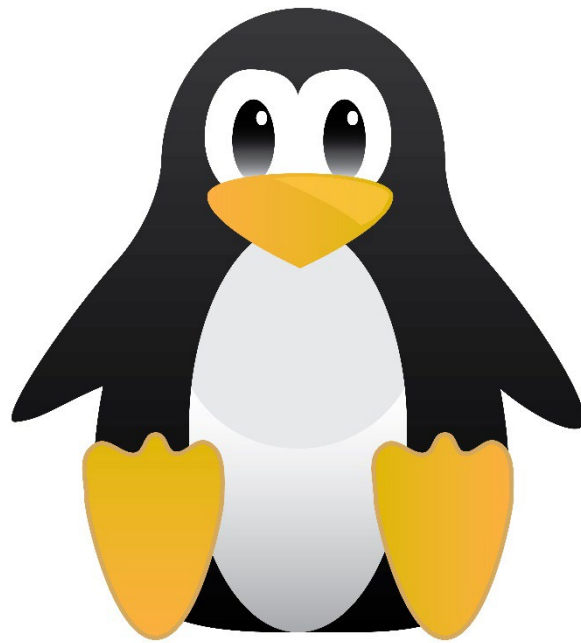
Uma frase famosa de Linus Torvalds é: “Se você não pode explicar algo de forma simples, você não entende o assunto o suficiente”. Isso reflete a filosofia de simplicidade e facilidade de uso que ele queria que o Linux tivesse.

O Linux inspirou-se no Unix. O Unix e o Linux são sistemas operacionais semelhantes, mas há algumas diferenças importantes. Enquanto o Unix é um sistema operacional proprietário, o Linux é de código aberto. Além disso, o Linux é considerado mais flexível e personalizável do que o Unix.

O mascote do Linux é conhecido como Tux, um pinguim (Figura 1). Ele foi criado por Larry Ewing em 1996 e se tornou um símbolo popular para representar o Linux. Frequentemente, ele é usado como logotipo de diversas distribuições Linux e é uma das imagens mais icônicas associadas ao sistema operacional.



Figura 1 – Mascote do Linux, o Tux



Crédito: Tae Mi/Shutterstock.

Linux é um sistema operacional de código aberto, o que significa que qualquer pessoa pode ver, modificar e distribuir o código fonte. Isso é possível graças à licença utilizada no Linux, a GPL (*General Public License*), criada por Richard Stallman, fundador do Projeto GNU (veremos isso melhor mais à frente).

Já o *kernel* Linux é o núcleo do sistema operacional, que gerencia as tarefas básicas do computador, como gerenciamento de memória e gerenciamento de arquivos. Porém, o *kernel* Linux por si só não é suficiente para ser utilizado como sistema operacional completo. Para isso, precisamos de *softwares* adicionais, como gerenciadores de janelas, programas de edição de texto e navegadores *web* etc. Esses *softwares* adicionais formam as distribuições Linux, como o Ubuntu, Debian e Fedora. Cada distribuição Linux tem suas próprias características e conjunto de *softwares* adicionais, permitindo que os usuários escolham a que melhor se adapta às suas necessidades.

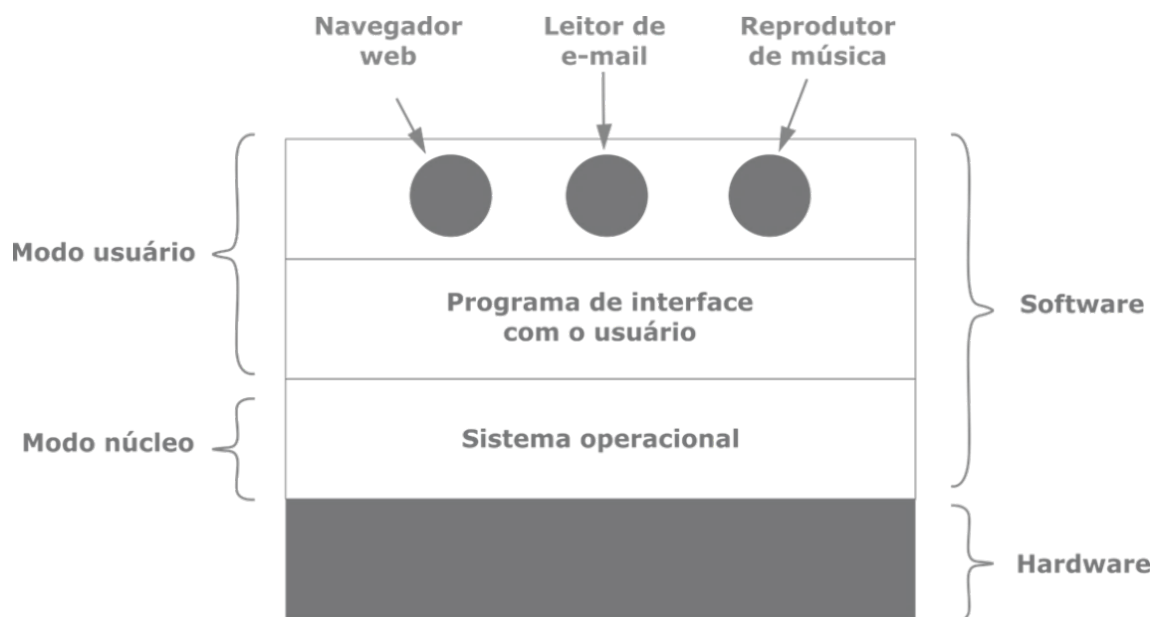
1.1 Kernel

Um *kernel* é o núcleo de um sistema operacional. As principais funcionalidades do *kernel* incluem:



- Gerenciamento de memória: o *kernel* é responsável por alocar e liberar memória para os programas, garantindo que cada programa tenha acesso à memória que precisa;
- Gerenciamento de processos: o *kernel* é responsável por criar, gerenciar e finalizar processos, ou seja, as tarefas que os programas realizam. Ele garante que cada processo tenha acesso à CPU e memória de forma equitativa;
- Gerenciamento de arquivos: o *kernel* é responsável por gerenciar os arquivos no sistema, incluindo a criação, leitura, escrita e exclusão de arquivos;
- Gerenciamento de dispositivos: o *kernel* é responsável por gerenciar os dispositivos conectados ao computador, como discos rígidos, impressoras e dispositivos USB. Ele também controla o acesso a esses dispositivos, garantindo que os programas possam usá-los de forma segura.

Figura 2 – Sistema operacional



Fonte: Tanenbaum, 2008, p. 1.

1.2 Projeto GNU

O Projeto GNU é um projeto de *software* livre criado em 1983 por Richard Stallman, programador, escritor e ativista político americano, considerado um dos principais defensores do *software* livre e da liberdade digital.



Stallman, além de fundador do Projeto GNU, é também o criador da *Fundação para o Software Livre*. O objetivo do projeto é desenvolver um sistema operacional completo, livre de restrições de uso e modificação, que possa ser usado por qualquer pessoa. O projeto foi iniciado com o objetivo de criar um sistema operacional semelhante ao Unix, mas com código fonte aberto.

O projeto GNU foi fundamental para o desenvolvimento do Linux, pois muitos dos componentes do Linux foram desenvolvidos como parte do projeto GNU. Existem distribuições Linux que são construídas inteiramente utilizando-se de componentes GNU, como o Debian, e são conhecidas como distribuições GNU/Linux. No entanto, nem todo o Linux é composto inteiramente de componentes GNU, algumas distribuições utilizam componentes de outros projetos de *software* livre e/ou proprietários.

1.3 Filosofia Linux

A filosofia do *software* livre é baseada no princípio de que este deve ser livre para ser usado, estudado, modificado e distribuído. Isso significa que os usuários não devem ser restritos em suas ações com o *software*, como é o caso do *software* proprietário, cujos usuários não podem ver o código fonte e não têm permissão para modificá-lo.

A filosofia do Linux é muito semelhante, pois o Linux é um sistema operacional de código aberto, o que significa que o código fonte está disponível para qualquer pessoa ver e modificar. Isso permite que outros desenvolvedores contribuam para o projeto e melhorem o sistema operacional.

Outra parte importante da filosofia do Linux e do *software* livre é a colaboração e a comunidade. Como o código fonte está disponível para todos, muitas pessoas trabalham juntas para melhorar o *software*. Isso cria uma comunidade de desenvolvedores, usuários e empresas que cooperam para melhorar o *software* e torná-lo mais acessível para todos.

1.4 Aplicações de Linux

O Linux é um sistema operacional versátil e utilizado em uma variedade de aplicações, desde dispositivos como *smartphones* e *Smart TVs*, até sistemas de grande porte como servidores e sistemas inteligentes.



Em servidores, devido à sua estabilidade, segurança e flexibilidade, muitas empresas e organizações utilizam o Linux como sistema operacional de servidor para aplicações como *web hosting*, *e-mail*, arquivos e banco de dados. Grandes empresas usam o Linux como sistema operacional para sistemas de nuvem, como o *Amazon Web Services* e o *Google Cloud*.

Outra aplicação comum do Linux é em dispositivos móveis, como *smartphones* e *tablets*. Muitos fabricantes de dispositivos móveis utilizam o Linux como sistema operacional em seus dispositivos, incluindo o *Android*, que é baseado no Linux. Além disso, o Linux é usado em *Smart TVs*, dispositivos de *streaming*, como o *Chromecast* e mesmo a *Raspberry Pi*.

O Linux também é usado em aplicações industriais, como sistemas de automação industrial, sistemas inteligentes transporte e sistemas de Internet das coisas (IoT). Isso se deve à sua estabilidade, segurança e capacidade de se adaptar às necessidades de diferentes aplicações.

1.5 Fundações ligadas ao Linux

A *Linux Foundation* e a *Free Software Foundation* (FSF) são duas organizações ligadas ao desenvolvimento e promoção do Linux.

A *Linux Foundation* é uma organização sem fins lucrativos que foi fundada em 2000. Ela tem como objetivo promover e proteger o Linux como um projeto de *software* livre e colaborativo. Ela patrocina o desenvolvimento do *kernel* Linux, organiza eventos para a comunidade Linux e fornece recursos para empresas que usam o Linux.

A *Free Software Foundation* (FSF) é a organização sem fins lucrativos fundada por Richard Stallman em 1985. Ela tem como objetivo promover a utilização de *software* livre e garantir a liberdade dos usuários. A FSF criou a licença GPL, e patrocina o desenvolvimento de vários projetos de *software* livre, incluindo o Projeto GNU. Além disso, a FSF luta pela liberdade do *software* e promove a filosofia do *software* livre.

TEMA 2 – LICENÇAS DE SOFTWARES

Licenças de *software* são contratos legais que regulamentam a utilização, distribuição e modificação de *software*. Elas definem as condições e restrições



sob as quais os usuários podem usar o *software*, bem como as responsabilidades dos desenvolvedores e distribuidores do *software*.

Existem diferentes tipos de licenças de *software*, incluindo licenças proprietárias, licenças gratuitas e licenças de *software* livre. Vamos agora conhecer um pouco mais as principais licenças existentes.

2.1 Software livre

Software livre é um tipo de *software* cujo código-fonte está disponível para ser modificado e distribuído livremente pelos usuários. Isso significa que os usuários podem fazer alterações no *software*, distribuí-lo e até mesmo vender suas próprias versões, desde que respeitem suas condições da licença. Além disso, os usuários têm acesso ao código-fonte, o que permite que eles entendam como o *software* funciona e possam resolver eventuais problemas.

Exemplos de *software* livre incluem o sistema operacional Linux, o browser Firefox, o editor de texto LibreOffice, o sistema de gerenciamento de banco de dados MySQL, entre outros. Alguns dos *softwares* mais conhecidos e utilizados no mundo são livres e são utilizados tanto por usuários comuns quanto por empresas e organizações governamentais. A filosofia do *software* livre é a de que todos devem ter acesso ao código-fonte e poder modificá-lo e distribuí-lo, sem restrições.

2.2 Open source

Softwares open source são aqueles cujos códigos-fontes estão disponíveis para serem estudados, modificados e distribuídos pelos usuários. A principal diferença entre *software* livre e *open source* é que o primeiro se baseia na liberdade de uso, estudo, modificação e distribuição, enquanto o segundo na disponibilidade do código-fonte para ser utilizado de forma colaborativa.

Embora o termo *open source* seja mais amplo e inclua *softwares* livres, nem todos os *softwares open source* são livres. Alguns deles podem ter restrições quanto às suas modificações e distribuições, enquanto outros podem ser vendidos como produtos comerciais. A filosofia do *open source* é a de que o compartilhamento do código-fonte permite a colaboração e o desenvolvimento de *software* mais inovador e de qualidade.



Exemplos de *softwares open source* incluem o sistema operacional Android, o *software* de edição de vídeo Blender, o gerenciador de projetos Redmine, entre outros. Eles são desenvolvidos e mantidos por uma comunidade global de desenvolvedores e usuários.

2.3 Copyleft

Licenças com *copyleft* são uma categoria especial de licenças de *software* livre que garantem que as modificações e distribuições derivadas do *software* se mantenham iguais. Isso é feito através da exigência de que qualquer pessoa que distribua o *software* ou suas modificações deve fornecer o código-fonte e licenciar o *software* sob os mesmos termos da licença original.

Exemplos de licenças com *copyleft* incluem a GPL v2 (*General Public License*), usada no Linux, a LGPL (*Lesser General Public License*) e a AGPL (*Affero General Public License*). Essas licenças são amplamente utilizadas em projetos de software livre, incluindo o Linux, o Firefox, o LibreOffice, o MySQL e muitos outros. Elas garantem que o *software* continue livre e acessível para todos, mesmo que seja modificado e distribuído por terceiros.

2.4 Freeware e shareware

Freeware é um tipo de *software* gratuito, que é distribuído sem custo algum e pode ser usado, copiado e distribuído livremente pelos usuários. No entanto, o código-fonte do *software* geralmente não está disponível, o que significa que os usuários não podem modificá-lo ou distribuí-lo como uma versão modificada. Exemplos de *freeware* incluem o navegador Firefox, o editor de texto Notepad ++ e o *software* antivírus Avast.

Já o *Shareware* é um tipo de *software* que é distribuído gratuitamente para teste e avaliação, mas que geralmente requer uma licença de uso paga para ser usado permanentemente. Esse tipo de *software* pode ser usado por um período limitado ou com certas restrições funcionais antes de precisar ser licenciado. Exemplos de *shareware* podem incluir jogos e *softwares* de grandes empresas, como o *software* de edição de fotos *Adobe Photoshop*.

Em resumo, a diferença entre *freeware* e *shareware* é que o primeiro é completamente gratuito para usar e distribuir, enquanto este é gratuito para



experimental, mas requer uma licença de uso paga para ser usado permanentemente.

2.5 Software proprietário

Software proprietário é um tipo de *software* que é controlado e licenciado por uma empresa ou indivíduo, que tem o direito exclusivo de distribuir, modificar e usar o *software*. Esse tipo de *software* geralmente vem com uma licença restritiva que proíbe a modificação, a distribuição e o uso compartilhado do *software* por terceiros. Além disso, o código-fonte do *software* geralmente não está disponível ao público.

Exemplos de *software* proprietário incluem o sistema operacional Windows da Microsoft, o *software* de edição de vídeo Adobe Premiere e o *software* de gerenciamento de projetos Microsoft Project. É importante notar que esses *softwares* são adquiridos mediante licenciamento, em que a compra da licença dá direito a uso, mas não à modificação do código fonte.

2.6 Software as a Service (SAAS)

Software as a Service (SAAS) é um modelo de entrega de *software* cujas aplicações são acessadas por meio da internet e geralmente são oferecidas por meio de assinatura. Isso significa que, em vez de instalar e gerenciar o *software* em seus próprios computadores, os usuários acessam a ele e a seus dados através da internet. A maioria dos provedores de SAAS oferecem acesso ao *software* por meio de um navegador *web*, ou aplicativos de *smartphones*, e os dados são armazenados em servidores remotos.

Exemplos de SAAS incluem o Google Apps, Salesforce, Dropbox, Netflix, Spotify e Slack. Esses serviços oferecem funcionalidades como *e-mails*, armazenamento de arquivos, gerenciamento de vendas, vídeos, música e comunicação de equipe, respectivamente, e são acessados via internet, permitindo o acesso a qualquer lugar e dispositivo com conexão. Este modelo de negócio vem sendo cada vez mais adotado mundo afora.

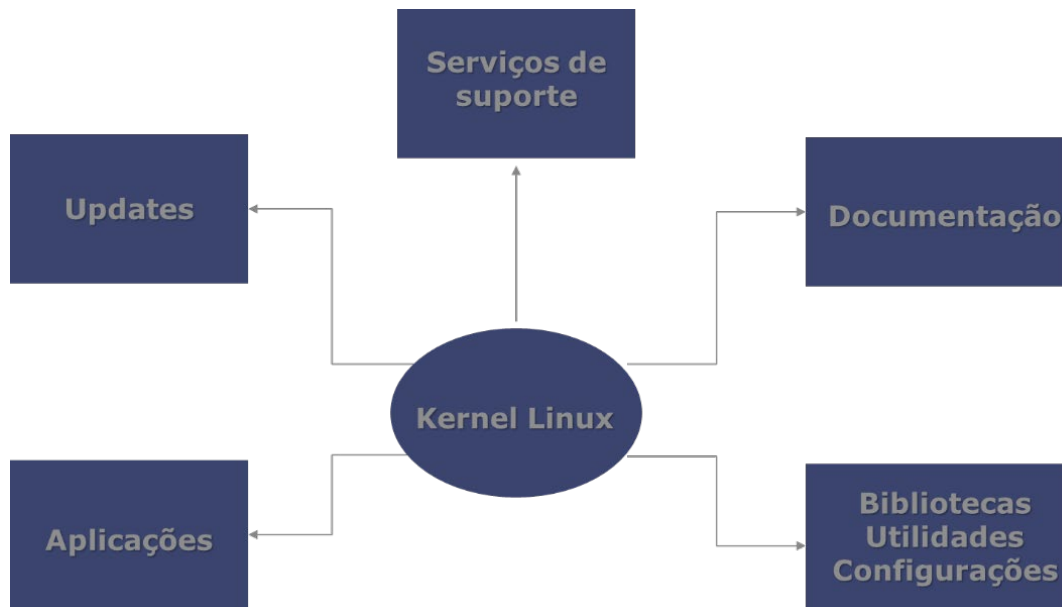
TEMA 3 – DISTRIBUIÇÕES LINUX

Uma distribuição Linux é uma versão do sistema operacional Linux que inclui um conjunto de ferramentas e programas adicionais. Elas são criadas por



diferentes organizações e comunidades, e cada uma tem seus próprios objetivos e características. Algumas distribuições Linux são projetadas para uso em *desktops* e *laptops*, enquanto outras são otimizadas para uso em servidores ou dispositivos embarcados. Veja, na Figura 3, o que compõe uma distribuição:

Figura 3 – Distribuições Linux



Crédito: Vinicius Pozzobon Borin.

Cada distribuição Linux tem sua própria filosofia e conjunto de ferramentas, mas todas usam o mesmo *kernel* Linux. Algumas distribuições Linux são mais fáceis de usar para iniciantes, enquanto outras são mais adequadas para usuários avançados ou para fins específicos, como desenvolvimento de *software* ou uso em servidores.

Algumas distribuições Linux são gratuitas e de código aberto, enquanto outras podem ser adquiridas comercialmente e oferecem suporte técnico e atualizações. Algumas distribuições Linux são desenvolvidas por comunidades, enquanto outras são desenvolvidas por empresas.

Saiba mais

Temos no mercado hoje centenas de distribuições. Acessando os *links* a seguir você encontra o catálogo delas, bem como os *links* para *download*:

LWN.net. Disponível em: <<https://lwn.net/>>. Acesso em: 22 fev. 2023.

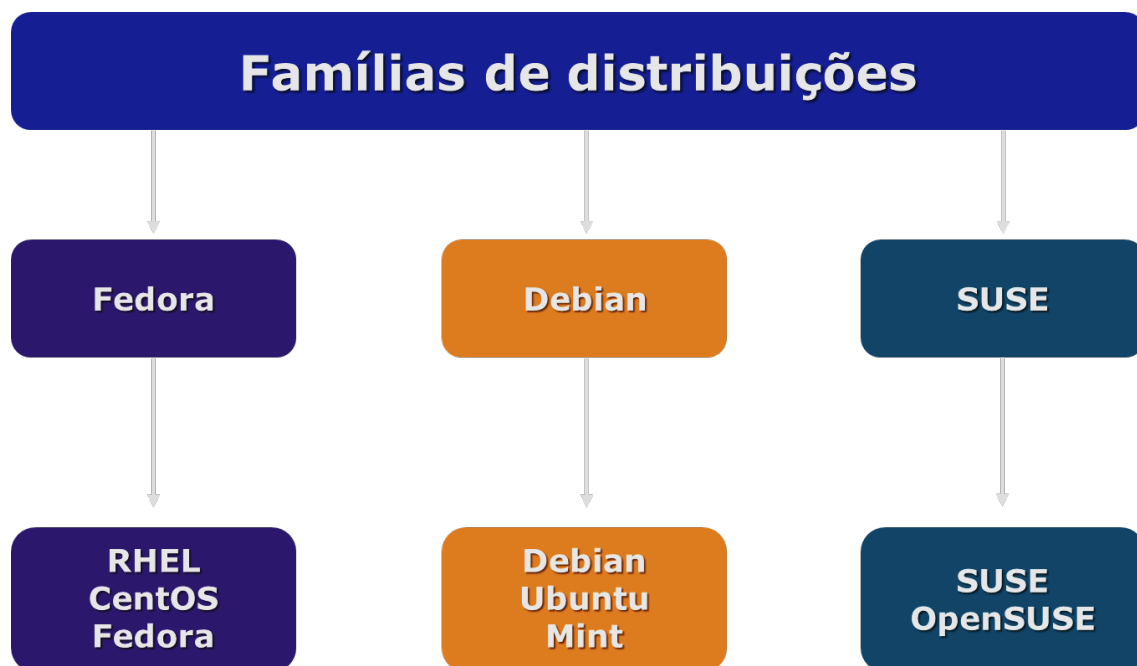
DISTROWATCH. Disponível em: <<https://distrowatch.com>>. Acesso em: 22 fev. 2023.

O *kernel* Linux, para criar sua própria distribuição, pode ser encontrado acessando-se o *link* a seguir:

MIRRORS.KERNEL.ORG. Disponível em: <<https://mirrors.edge.kernel.org/>>. Acesso em: 22 fev. 2023.

Na Figura 4 temos algumas das principais família e distribuições Linux.

Figura 4 – As principais famílias e distribuições



Crédito: Vinicius Pozzobon Borin.

3.1 Debian

A distribuição Debian é uma das mais antigas e respeitadas distribuições Linux. Ela foi criada em 1993 pelo programador Ian Murdock, com o objetivo de criar um sistema operacional livre e de código aberto que pudesse ser usado em várias arquiteturas de computador.

Uma das principais características da Debian é o seu gerenciador de pacotes, o APT (*Advanced Package Tool* - *.deb*), que permite que os usuários instalem, atualizem e removam facilmente pacotes de *software*. Isso torna a administração do sistema muito mais fácil e eficiente.



Outra característica interessante da Debian é o projeto Debian/Hurd, que visa criar uma versão da distribuição baseada no *kernel* Hurd (Sistema de núcleos da GNU), em vez do núcleo Linux. O Hurd é projetado para ser mais seguro e escalável que o Linux e oferece recursos avançados, como suporte a transações. Embora ainda não esteja completamente madura, a Debian/Hurd é uma ótima opção para aqueles interessados em experimentar novas tecnologias.

O Debian vem com diferentes interfaces gráficas, sendo a principal delas o GNOME. Além disso, a distribuição Debian deu origem a diferentes outras distribuições, como o Ubuntu, Mint e o Kurumin (distribuição brasileira). Outra variação é o *Raspbian*, otimizada para ser executada em dispositivos *Raspberry Pi*. Ele inclui pacotes específicos para *Raspberry Pi*, como o gerenciador de janelas PIXEL e o *software Scratch* para programação de dispositivos.

3.2 Ubuntu

A distribuição Ubuntu é um sistema operacional baseado em Linux desenvolvido pela empresa sul africana chamada *Canonical*. Ele é projetado para ser fácil de usar e acessível para usuários iniciantes, e bom para usuário avançados também.

O Ubuntu é baseado no Debian e usa o gerenciador de pacotes *apt* para instalar e gerenciar programas. O comando *sudo* surgiu com o Ubuntu. Ele é usado para dar privilégios de administrador ao usuário, permitindo que ele execute comandos que precisam de permissões elevadas.

Além disso, o Ubuntu também é compatível com o WSL (Subsistema Windows para Linux) do Windows 10, o que permite que os usuários executem comandos Linux diretamente no Windows sem precisar de uma máquina virtual ou dual boot. Isso pode ser útil para desenvolvedores que precisam alternar entre o Windows e o Linux.

3.3 Mint

A distribuição Linux Mint é uma versão baseada no Ubuntu que foi criada para fornecer uma experiência de usuário similar ao Windows, tornando-a uma boa opção para usuários que estão migrando do Windows para o Linux.

O Linux Mint vem acompanhada de um ambiente de *desktop* chamado *Cinnamon*, que é projetado para ser fácil de usar e intuitivo. Ele inclui uma barra



de tarefas e um menu iniciar semelhantes ao Windows, o que ajuda os usuários a se sentirem mais confortáveis enquanto se acostumam com o novo sistema operacional.

Além disso, o Linux Mint vem com uma variedade de aplicativos populares pré-instalados, como o LibreOffice (uma alternativa gratuita ao Microsoft Office) e o Firefox (navegador da web), para que os usuários possam começar a trabalhar imediatamente após a instalação.

3.4 Red Hat

A distribuição Red Hat é baseada em código aberto e é amplamente utilizada em empresas e organizações de todos os tamanhos.

Uma das principais características da distribuição Red Hat é o seu gerenciador de pacotes, o RPM (*Red Hat Package Manager*). Ele garante que todos os pacotes instalados estejam em conformidade com as políticas de segurança e estabilidade do sistema.

É importante mencionar que a distribuição Red Hat não é gratuita. Ela é disponibilizada sob uma licença comercial, o que significa que os usuários precisam pagar por suporte e atualizações de segurança. No entanto, os usuários podem baixar e usar a distribuição gratuitamente, sem suporte técnico ou atualizações de segurança.

TEMA 4 – TERMINOLOGIA LINUX

O universo do Linux apresenta terminologias bastante particulares e que precisam ser conhecidas por quem trabalha com ele. Já aprendemos termos como *kernel*, distribuição e GNU. Mas vamos conhecer mais alguns deles agora.

4.1 Processo de *boot*

O processo de inicialização de um computador, conhecido como *boot*, é uma sequência de etapas que acontecem quando você liga o seu computador. Ele começa com o carregamento do *firmware*, como o BIOS ou UEFI, que é um *software* básico armazenado na memória do computador que controla as funções básicas do *hardware*.

O *firmware* verifica se o disco rígido e outros dispositivos de armazenamento estão conectados e funcionando corretamente, e então procura

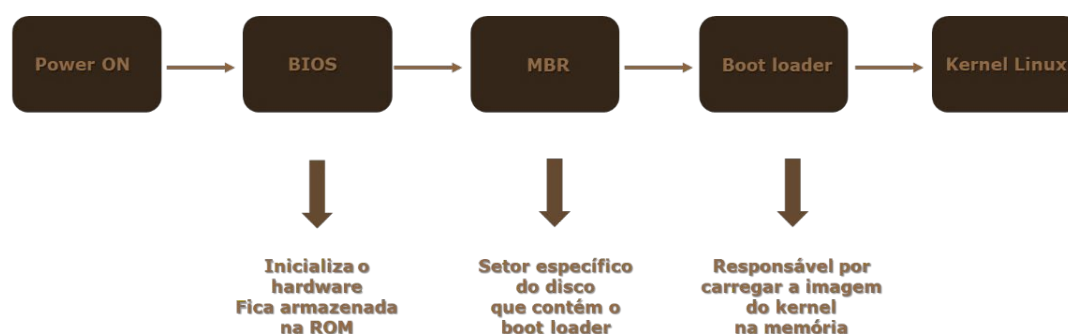


por um sistema operacional para carregar (*boot loader*). Se você estiver usando o Linux, o *firmware* irá procurar pelo arquivo de *boot loader*, se for no Linux, geralmente é o GRUB, que é responsável por carregar o *kernel* Linux.

Uma vez que o *kernel* é carregado, ele inicia o processo de inicialização do sistema operacional, carregando os módulos de dispositivos e configurando as configurações de *hardware*. Em seguida, ele inicia os serviços do sistema, como o gerenciador de *login*, e finalmente, inicia a interface gráfica ou o *prompt* de comando, dando a você acesso ao seu sistema.

No Linux, existem vários *boot loaders* disponíveis, mas os mais comuns são o GRUB, LILO e ISOLINUX. O GRUB (*Grand Unified Boot Loader*) é o *boot loader* padrão para a maioria das distribuições Linux. Ele oferece uma interface amigável para selecionar o sistema operacional a ser iniciado e também fornece recursos avançados como a capacidade de editar a linha de comando do *kernel* antes do seu início. A Figura 5 apresenta em um diagrama a explicação de inicialização dada acima:

Figura 5 – Processo de inicialização



Crédito: Vinicius Pozzobon Borin.

4.2 Sistema de arquivos

Sistemas de arquivos são estruturas de dados que organizam e gerenciam arquivos em um dispositivo de armazenamento, como um disco rígido. Eles são responsáveis por criar, ler, atualizar e excluir arquivos, bem como gerenciar permissões de acesso e outras informações importantes. No Linux, os sistemas de arquivos mais comuns incluem o Ext4, o XFS, e o Btrfs.

O Ext4 é amplamente utilizado e é o sistema de arquivos padrão para a maioria das distribuições Linux modernas. Ele oferece uma boa relação entre



desempenho e capacidade de armazenamento, além de suportar arquivos de tamanho de até 16 TB (TeraBytes).

O XFS é outro sistema de arquivos popular no Linux, especialmente para sistemas que trabalham com arquivos gigantescos. Ele oferece suporte a arquivos de até 9 EB (ExaBytes) e é usado em sistemas de arquivos de alta capacidade, como servidores de arquivos e sistemas de armazenamento de massa (*big data*).

Por fim, o Btrfs é um sistema de arquivos de nova geração, projetado para oferecer recursos avançados como *snapshots*, replicação e balanceamento de carga. Ele ainda não é tão amplamente utilizado quanto o Ext4 ou XFS, mas é considerado como uma boa opção para sistemas de arquivos de alta capacidade e que precisam de recursos avançados.

Quadro 1 – Diferenças entre os tipos sistemas de arquivos e também de nomenclatura entre o ambiente Windows e o Linux.

	Windows	Linux
Partição	Disk1	/dev/sda1
Tipo de Sistema de arquivos	NTFS/FAT	EXT3/EXT4...
Diretório base (onde o SO é armazenado)	C:\	/

Fonte: Borin, 2023.

4.3 X Window System e a Interface Gráfica Linux

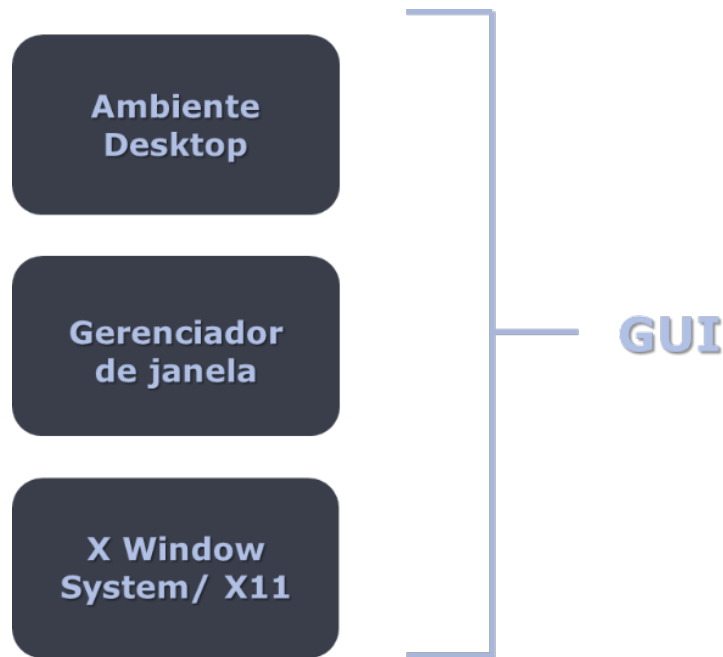
O *X Window System*, também conhecido como X ou X11, é um sistema de gerenciamento de janelas para sistemas Unix-like, como o Linux. Ele fornece uma camada de gerenciamento de janelas para aplicativos gráficos, permitindo que os usuários criem e gerenciem janelas, menus e outros elementos gráficos em uma interface gráfica de usuário.

Os aplicativos se comunicam com o servidor X através de uma API (*Application Programming Interface*), chamada *Xlib*, que permite que os aplicativos criem e gerenciem janelas, desenhem elementos gráficos e gerem eventos de entrada, como cliques de *mouse* e teclas pressionadas.

O *X Window System* é somente a camada inferior de toda a interface gráfica - *Graphic User Interface* (GUI), do Linux. Na Figura 6 vemos como a interface é construída:



Figura 6 – Graphic User Interface (GUI) do Linux



Crédito: Vinicius Pozzobon Borin.

Quando falamos de ambiente *desktop* e gerenciadores de janelas, precisamos citar Gnome e o KDE, que são os mais comuns de serem encontrados em distribuições Linux.

Falando um pouco do Gnome, ele fornece uma interface gráfica de usuário completa, incluindo um gerenciador de janelas, barra de tarefas, painel e uma coleção de aplicativos para tarefas comuns, como navegação de arquivos, visualização de imagens, edição de texto e muito mais. O Gnome foi criado com o objetivo de oferecer uma interface amigável e acessível para o sistema operacional Linux. Ele é projetado para ser intuitivo e fácil de usar, com uma interface limpa e consistente, e é uma das principais opções de ambiente de *desktop* utilizadas no Linux.

4.4 Linha de comando (terminal)

A linha de comando, também conhecida como *terminal*, é um meio de interagir com o sistema operacional por meio de comandos digitados em uma interface de texto. No Linux, a linha de comando é uma das formas mais poderosas e flexíveis de se trabalhar com o sistema operacional, permitindo que os usuários executem tarefas avançadas, como gerenciar arquivos, configurar o



sistema, instalar e desinstalar programas, entre outras, sem a necessidade de uma interface gráfica.

Além disso, os comandos digitados na linha de comando também podem ser automatizados através de scripts para realizar tarefas repetitivas ou complexas. A linha de comando é uma ferramenta importante para os administradores de sistemas e desenvolvedores, pois permite uma maior eficiência e automatização.

4.5 Shell

O *shell* é um interpretador de comandos que permite aos usuários interagir com o sistema operacional por meio da linha de comando. No Linux, existem vários tipos de *shells* disponíveis, cada um com suas próprias características e comandos. Alguns exemplos de *shells* comuns no Linux incluem o Bash (*Bourne-Again Shell*), o Zsh (*Z Shell*) e o Tcsh (*Tenex C Shell*).

O Bash é o *shell* padrão na maioria das distribuições Linux e é muito popular entre os usuários. Ele fornece uma variedade de recursos, como autocompletar comandos e nomes de arquivos, histórico de comandos e suporte para *scripts*.

TEMA 5 – SERVIÇOS NO LINUX E CERTIFICAÇÕES

Vamos agora finalizar nossa etapa sobre Linux conversando sobre alguns tópicos distintos dentro Linux que você precisa conhecer.

5.1 Serviços no Linux

Podemos simplificar a explicação e dizer que os serviços nada mais são do que programas que instalamos no Linux para desenvolver determinadas tarefas. Existem muitos serviços disponíveis para usuários de Linux, cada um com suas próprias características e funcionalidades únicas. Vamos citar alguns deles a seguir.

Um dos principais serviços disponíveis no Linux é o LibreOffice, que é um pacote de aplicativos de escritório de código aberto que inclui um processador de texto, planilha eletrônica, apresentação, editor de desenhos e muito mais. Ele é compatível com arquivos do Microsoft Office e é uma excelente opção para aqueles que precisam trabalhar com documentos e planilhas.



Outro serviço popular no Linux é o Mozilla Firefox. Este é um navegador *web* de código aberto que oferece uma variedade de recursos, como a capacidade de adicionar extensões e personalizar a interface do usuário. Além disso, o Firefox é conhecido por ser mais seguro e privado do que outros navegadores populares, o que o torna uma excelente escolha para usuários preocupados com a privacidade.

Podemos citar inúmeros outros serviços para Linux, e que também existem no Windows, como o editor de imagens GIMP, player de vídeos VLC e o gravador de vídeos, OBS Studio.

Figura 7 – Exemplos de serviços no Linux



tanuha2001 / Shutterstock



Fonset: <https://www.libreoffice.org/discover/libreoffice/>



3DMAVR / Shutterstock

Piotr Swat / Shutterstock



dizain / Shutterstock

5.2 Jogos no Linux

Um dos objetivos desta etapa é também instigar você a usar o Linux no seu dia a dia, mesmo que para entretenimento, até que se acostume com ele e talvez o adote como seu sistema operacional principal no futuro. Assim, nada mais justo do que comentar um pouco sobre como jogar *games* em Linux.

Jogos em Linux são uma ótima opção para aqueles que desejam jogar em um sistema operacional livre e de código aberto. Existem muitos jogos disponíveis para Linux, desde jogos indie até jogos AAA. No entanto, alguns jogos não são nativos para Linux e precisam ser executados utilizando-se de um *software* de emulação, como o Wine ou o Proton.



O Wine permite que você execute programas do Windows no Linux. Ele é compatível com uma ampla variedade de jogos e aplicativos, mas pode exigir alguma configuração e otimização para funcionar corretamente.

Já o Proton é uma versão do Wine desenvolvida pela Valve (criadora da Steam) para melhorar a compatibilidade de jogos do Windows com o Linux. Ele já vem com configurações otimizadas e é mais fácil de usar que o Wine.

5.3 Certificações Linux

As certificações e provas de certificação na área da computação são uma excelente maneira de demonstrar suas habilidades e conhecimentos em um determinado campo ou tecnologia. Elas são frequentemente exigidas por empregadores e podem ajudar a abrir portas para novas oportunidades de carreira. Além disso, as certificações podem ajudá-lo a se manter atualizado com as últimas tendências e desenvolvimentos em sua área de interesse.

Existem várias organizações que oferecem certificações em diferentes áreas da computação, como programação, segurança, banco de dados e redes. Alguns exemplos incluem a Microsoft, a Cisco e a Oracle. Cada organização tem seus próprios requisitos para a certificação, que podem incluir a realização de cursos de treinamento, experiência prévia e aprovação em uma prova de certificação.

A importância das certificações varia de acordo com a área de trabalho, mas geralmente é uma prova de sua competência e habilidade para trabalhar com tecnologias específicas. Isso pode ajudar a diferenciá-lo de outros candidatos em uma entrevista de emprego ou aumentar suas chances de promoção. Além disso, as certificações podem ajudá-lo a se manter atualizado com as últimas tendências e desenvolvimentos em sua área de interesse e aumentar sua confiança no seu trabalho.

A *Linux Professional Institute* (LPI) é uma organização sem fins lucrativos que oferece certificações para profissionais de sistemas Linux. A LPI é reconhecida mundialmente como líder em certificação Linux e tem como objetivo ajudar os profissionais a provar suas habilidades e conhecimentos em sistemas Linux.

A LPI oferece quatro níveis de certificação: Linux Essentials, LPIC-1, LPIC-2 e LPIC-3. A Linux Essentials não é obrigatória, e foi feita para quem não

conhece nada de Linux. O LPIC-1 é a certificação básica e abrange os conhecimentos necessários para instalar, configurar, usar o terminal e administrar sistemas Linux. O LPIC-2 é a certificação intermediária, e a LPIC-3 é a certificação avançada e abrange tópicos especializados, como virtualização e segurança avançada. Na Figura 8 vemos a sequência dessas certificações.

Figura 8 – Certificações Linux da LPI



FINALIZANDO

Ao longo desta etapa demos nossos primeiros passos com o Linux. Caso você nunca tenha utilizado esse sistema operacional, espero que tenha se interessado e instigado você a conhecê-lo melhor, pois é amplamente utilizado no mercado de tecnologia. Caso já tenha experiência com ele, espero que tenha trazido a você mais algum conhecimento que você ainda não tinha.



REFERÊNCIAS

LINUX.ORG. Disponível em: <<https://www.linux.org/>>. Acesso em: 22 fev. 2023.

SILVA, R. L. **Linux**: guia de instalação e administração. São Paulo: Novatec, 201

TANENBAUM, A. S. **Sistemas operacionais modernos**. 4. ed. São Paulo: Pearson Prentice Hall, 2008.