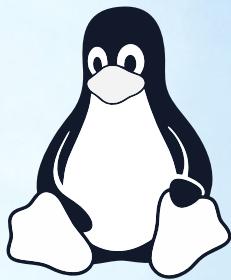


COLEÇÃO: EU QUERO UMA CARREIRA - UTAH

E-BOOKS - PREPARATÓRIO PARA CERTIFICAÇÃO

Linux



Linux
Professional
Institute

LPIC

LINUX ESSENTIALS

E-Book 02

O FENÔMENO - OPEN SOURCE

Caminhe com
os gigantes

Sumário

■ <i>Introdução - Distribuições Linux</i>	05
■ <i>Sistemas Embarcados, Android</i>	06
■ <i>Exercícios - LPI Essentials</i>	07
■ <i>Introdução - Pacotes de Softwares</i>	09
■ <i>Instalação de Pacotes</i>	09
■ <i>Remoção de Pacotes</i>	13
■ <i>Navegadores e Multimedia</i>	14
■ <i>Servidores</i>	15
■ <i>Administração de Rede</i>	16
■ <i>Linguagem de Programação</i>	17
■ <i>Aplicativos Para Desktop</i>	18
■ <i>Exercícios - LPI Essentials</i>	19
■ <i>Definição de Software Livre de Código Aberto</i>	21
■ <i>Open Source vs Free Software</i>	22
■ <i>Licenças</i>	23
■ <i>CopyLeft</i>	23
■ <i>Definição de Código Aberto e Licenças Permissivas</i>	24
■ <i>Creative Commons</i>	25
■ <i>Modelos de Negócio em Software Livre</i>	25
■ <i>Exercícios - LPI Essentials</i>	26

Parabéns. Agora você caminha com os Gigantes!

Olá!

Meu nome é **Fábio Pires**, sou CEO do **Grupo Utah**.

Para nós é uma imensa satisfação, poder contribuir para o seu crescimento pessoal e profissional.

Nosso propósito como empresa é transformar vidas, através de um **aprendizado focado no aluno**, levando-os a um patamar **acima da média**, utilizando para isso, treinamentos, baseados em certificações **internacionalmente reconhecidas** no Mercado de T.I.

Esperamos que este conteúdo sirva para apoiar você, na sua jornada rumo a umas das Certificações mais cobiçadas do Mercado de Infra Estrutura, **a LPIC**.

Sinta-se a vontade para nos contactar e descobrir como, uma metodologia diferenciada com foco no aprendizado do aluno, pode levar você a ter resultados além do esperado.

A pergunta que fica é:

Você quer ter, uma carreira ou um emprego?

Se a resposta for, **CARREIRA**, meus parabéns este material foi desenvolvido para você.

Aproveite. Neste momento, começa sua **caminhada ao lado de Gigantes!**

Fábio Pires





Peso	2
Descrição	Conhecimento do desenvolvimento Linux e das principais distribuições
Áreas de conhecimento	<ul style="list-style-type: none"><input type="checkbox"/> Distribuições<input type="checkbox"/> Sistemas Embarcados<input type="checkbox"/> Linux na Nuvem
Itens relacionados	<ul style="list-style-type: none"><input type="checkbox"/> Debian, Ubuntu (LTS)<input type="checkbox"/> CentOS, openSUSE, Red Hat, SUSE<input type="checkbox"/> Linux Mint, Scientific Linux<input type="checkbox"/> Raspberry Pi, Raspbian<input type="checkbox"/> Android



■ Introdução

O Linux é um dos sistemas operacionais mais populares. Seu desenvolvimento foi iniciado em 1991 por Linus Torvalds. O sistema operacional foi inspirado no Unix, outro sistema operacional desenvolvido na década de 1970 pela AT&T Laboratories.

O Unix era voltado para computadores pequenos. Na época, computadores "pequenos" eram considerados máquinas que não precisam de um salão inteiro com ar condicionado e custam menos de um milhão de dólares. Mais tarde, foram consideradas as máquinas que podem ser levantadas por duas pessoas. Naquela época, o Unix não estava disponível em computadores pequenos, como computadores de escritório baseados na plataforma x86. Portanto, Linus, que era estudante na época, começou a implementar um sistema operacional semelhante ao Unix, que deveria rodar nesta plataforma.

Principalmente, o Linux usa os mesmos princípios e idéias básicas do Unix, mas o próprio Linux não contém código Unix, pois é um projeto independente. O Linux não é suportado por uma empresa individual, mas por uma comunidade internacional de programadores. Como está disponível gratuitamente, pode ser usado por qualquer pessoa sem restrições..

■ Distribuições Linux

Uma distribuição Linux é um pacote configurável que consiste em um kernel Linux e em uma seleção de aplicativos mantidos por uma empresa ou comunidade de usuários. O objetivo de uma distribuição é otimizar o kernel e os aplicativos executados no sistema operacional para um determinado caso de uso ou grupo de usuários. As distribuições geralmente incluem ferramentas específicas da distribuição para instalação de software e administração do sistema. É por isso que algumas distribuições são usadas principalmente para ambientes de desktop em que precisam ser fáceis de usar, enquanto outras são usadas principalmente para executar em servidores para usar os recursos disponíveis da maneira mais eficiente possível.

Outra maneira de classificar distribuições é referindo-se à família de distribuição à qual eles pertencem. As distribuições da família de distribuição Debian usam o gerenciador de pacotes dpkg para gerenciar o software que é executado no sistema operacional. Os pacotes que podem ser instalados com o gerenciador de pacotes são mantidos por membros voluntários da comunidade da distribuição. Os mantenedores usam o formato do pacote deb para especificar como o software é instalado no sistema operacional e como é configurado por padrão. Assim como uma distribuição, um pacote é um pacote de software e uma configuração e documentação correspondentes que facilitam a instalação, atualização e uso do software.

A distribuição Debian GNU / Linux é a maior distribuição da família de distribuição Debian. O Projeto Debian GNU / Linux foi lançado por Ian Murdock em 1993. Hoje milhares de voluntários estão trabalhando no projeto. O Debian GNU / Linux visa fornecer um sistema operacional muito confiável. Também promove a visão de Richard Stallman de um sistema operacional que respeita as liberdades do usuário para executar, estudar, distribuir e melhorar o software. É por isso que ele não fornece nenhum software proprietário por padrão.

O Ubuntu é outra distribuição baseada no Debian que vale a pena mencionar. O Ubuntu foi criado por Mark Shuttleworth e sua equipe em 2004, com a missão de criar um ambiente de desktop Linux fácil de usar. A missão do Ubuntu é fornecer um software gratuito para todos em todo o mundo, bem como reduzir o custo de serviços profissionais. A distribuição tem uma liberação programada a cada seis meses, com uma liberação de suporte a longo prazo a cada 2 anos.

Red Hat é uma distribuição Linux desenvolvida e mantida pela empresa de software com nome idêntico, adquirida pela IBM em 2019. A distribuição Red Hat Linux foi iniciada em 1994 e renomeada em 2003 para o Red Hat Enterprise Linux, frequentemente abreviada como RHEL. Ele é fornecido às empresas como uma solução corporativa confiável, suportada pela Red Hat, e vem com um software que visa facilitar o uso do Linux em ambientes de servidores profissionais. Alguns de seus componentes exigem assinaturas ou licenças com base em taxas.



O projeto CentOS usa o código-fonte disponível gratuitamente do Red Hat Enterprise Linux e o compila em uma distribuição disponível gratuitamente, mas, em troca, não é fornecida com suporte comercial.

O RHEL e o CentOS são otimizados para uso em ambientes de servidor. O projeto Fedora foi fundado em 2003 e cria uma distribuição Linux voltada para computadores desktop. A Red Hat iniciou e mantém a distribuição do Fedora desde então. O Fedora é muito progressivo e adota novas tecnologias muito rapidamente e às vezes é considerado um banco de testes para novas tecnologias que posteriormente poderão ser incluídas no RHEL. Todas as distribuições baseadas no Red Hat usam o formato de pacote rpm.

A empresa SUSE foi fundada em 1992 na Alemanha como um provedor de serviços Unix. A primeira versão do SUSE Linux foi lançada em 1994. Ao longo dos anos, o SUSE Linux ficou conhecido principalmente por sua ferramenta de configuração do YaST. Essa ferramenta permite aos administradores instalar e configurar software e hardware, instalar servidores e redes. Semelhante ao RHEL, o SUSE lança o SUSE Linux Enterprise Server, que é sua edição comercial. É lançado com menos frequência e é adequado para implantação empresarial e de produção. É distribuído como servidor e ambiente de desktop, com pacotes adequados à finalidade. Em 2004, o SUSE lançou o projeto openSUSE, que abriu oportunidades para desenvolvedores e usuários testarem e desenvolverem ainda mais o sistema. A distribuição do openSUSE está disponível gratuitamente para download.

Distribuições independentes foram lançadas ao longo dos anos. Alguns deles são baseados no Red Hat ou no Ubuntu, outros são projetados para melhorar uma propriedade específica de um sistema ou hardware. Existem distribuições criadas com funcionalidades específicas, como o QubesOS, um ambiente de desktop muito seguro, ou o Kali Linux, que fornece um ambiente para explorar vulnerabilidades de software, usadas principalmente por testadores de penetração. Recentemente, várias distribuições Linux super pequenas foram projetadas para serem executadas especificamente em contêineres Linux, como o Docker. Também existem distribuições criadas especificamente para componentes de sistemas embarcados e até dispositivos inteligentes.

■ *Sistemas Embarcados*

Os sistemas incorporados são uma combinação de hardware e software de computador projetados para ter uma função específica em um sistema maior. Geralmente eles fazem parte de outros dispositivos e ajudam a controlar esses dispositivos. Os sistemas incorporados são encontrados em aplicações automotivas, médicas e até militares. Devido à sua grande variedade de aplicativos, uma variedade de sistemas operacionais baseados no kernel Linux foi desenvolvida para ser usada em sistemas embarcados. Uma parte significativa dos dispositivos inteligentes possui um sistema operacional baseado em kernel Linux em execução.

Portanto, com sistemas embarcados vem o software embarcado. O objetivo deste software é acessar o hardware e torná-lo utilizável. As principais vantagens do Linux sobre qualquer software incorporado proprietário incluem compatibilidade, desenvolvimento, suporte e sem plataforma de plataforma entre fornecedores. Dois dos projetos de software incorporado mais populares são o Android, usado principalmente em telefones celulares em vários fornecedores e o Raspbian, usado principalmente no Raspberry Pi.

■ *Android*

O Android é principalmente um sistema operacional móvel desenvolvido pelo Google. O Android Inc. foi fundado em 2003 em Palo Alto, Califórnia. A empresa criou inicialmente um sistema operacional para rodar em câmeras digitais. Em 2005, o Google comprou o Android Inc. e o desenvolveu para ser um dos maiores sistemas operacionais móveis.

A base do Android é uma versão modificada do kernel Linux com software de código aberto adicional. O sistema operacional é desenvolvido principalmente para dispositivos touchscreen, mas o Google desenvolveu versões para TV e relógios de pulso. Diferentes versões do Android foram desenvolvidas para consoles de jogos, câmeras digitais e computadores.



O Android está disponível gratuitamente em código aberto como Android Open Source Project (AOSP). O Google oferece uma série de componentes proprietários, além do núcleo de código aberto do Android. Esses componentes incluem aplicativos como o Google Agenda, Google Maps, Google Mail, o navegador Chrome e a Google Play Store, o que facilita a instalação fácil de aplicativos. A maioria dos usuários considera essas ferramentas parte integrante de sua experiência com o Android. Portanto, quase todos os dispositivos móveis enviados com Android na Europa e América incluem software proprietário do Google.

O Android em dispositivos incorporados tem muitas vantagens. O sistema operacional é intuitivo e fácil de usar com uma interface gráfica do usuário, possui uma comunidade de desenvolvedores muito ampla, portanto, é fácil encontrar ajuda para o desenvolvimento. Também é suportado pela maioria dos fornecedores de hardware com um driver Android, portanto, é fácil e econômico prototipar um sistema inteiro.

■ *Raspbian e o Raspberry Pi*

O Raspberry Pi é um computador de baixo custo, do tamanho de um cartão de crédito, que pode funcionar como um computador de mesa com todas as funcionalidades, mas pode ser usado em um sistema Linux incorporado. É desenvolvido pela Raspberry Pi Foundation, uma instituição de caridade educacional sediada no Reino Unido. Ele tem principalmente o objetivo de ensinar jovens a aprender a programar e entender a funcionalidade dos computadores. O Raspberry Pi pode ser projetado e programado para executar tarefas ou operações desejadas que fazem parte de um sistema muito mais complexo.

As especialidades do Raspberry Pi incluem um conjunto de pinos GPIO (General Purpose Input-Output) que podem ser usados para conectar dispositivos eletrônicos e placas de extensão. Isso permite usar o Raspberry Pi como uma plataforma para desenvolvimento de hardware. Embora tenha sido destinado a fins educacionais, o Raspberry Pis é usado hoje em vários projetos de bricolage, bem como em protótipos industriais no desenvolvimento de sistemas embarcados.

■ *Exercícios:*

- 01 Qual a diferença entre o Debian GNU / Linux e o Ubuntu? Cite dois aspectos.
- 02 - Quais são os ambientes / plataformas mais comuns para os quais o Linux é usado? Nomeie três ambientes / plataformas diferentes e nomeie uma distribuição que você pode usar para cada um.
- 03 Você está planejando instalar uma distribuição Linux em um novo ambiente. Cite quatro coisas que você deve considerar ao escolher uma distribuição.
- 04 Cite três dispositivos nos quais o sistema operacional Android é executado, exceto os smartphones.
- 05 Explique três grandes vantagens da computação em nuvem.



Peso	3
Descrição	Conhecer as principais aplicações, bem como seus usos e desenvolvimento.
Áreas de conhecimento	<ul style="list-style-type: none">▫ Aplicativos de Desktop▫ Aplicações de Servidor▫ Linguagens de Programação▫ Ferramentas e repositórios de gerenciamento de
Itens relacionados	<ul style="list-style-type: none">▫ OpenOffice.org, LibreOffice, Thunderbird, Firefox, GIMP▫ Nextcloud, ownCloud▫ Apache HTTPD, NGINX, MariaDB, MySQL, NFS, Samba▫ C, Java, JavaScript, Perl, shell, Python, PHP



■ Introdução

Um aplicativo é um programa de computador cuja finalidade não está diretamente ligada ao funcionamento interno do computador, mas com tarefas executadas pelo usuário. As distribuições Linux oferecem muitas opções de aplicativos para executar uma variedade de tarefas, como aplicativos de escritório, navegadores da Web, players e editores de multimídia, etc. Geralmente, há mais de um aplicativo ou ferramenta para executar um trabalho específico. Cabe ao usuário escolher o aplicativo que melhor se adapta às suas necessidades.

■ Pacotes de Software

Quase todas as distribuições Linux oferecem um conjunto pré-instalado de aplicativos padrão. Além desses aplicativos pré-instalados, uma distribuição possui um repositório de pacotes com uma vasta coleção de aplicativos disponíveis para instalação por meio de seu gerenciador de pacotes. Embora as várias distribuições ofereçam aproximadamente as mesmas aplicações, existem vários sistemas de gerenciamento de pacotes diferentes para várias distribuições.

Por exemplo, Debian, Ubuntu e Linux Mint usam as ferramentas dpkg, apt-get e apt para instalar pacotes de software, geralmente chamados de pacotes DEB. Distribuições como Red Hat, Fedora e CentOS usam os comandos rpm, yum e dnf, que por sua vez instalam pacotes RPM. Como o pacote de aplicativos é diferente para cada família de distribuição, é muito importante instalar pacotes do repositório correto projetado para a distribuição específica. O usuário final geralmente não precisa se preocupar com esses detalhes, pois o gerente de pacotes da distribuição escolherá os pacotes certos, as dependências necessárias e as atualizações futuras. Dependências são pacotes auxiliares necessários aos programas. Por exemplo, se uma biblioteca fornecer funções para lidar com imagens JPEG usadas por vários programas, essa biblioteca provavelmente será empacotada em seu próprio pacote, do qual todos os aplicativos que usam a biblioteca dependem.

Os comandos dpkg e rpm operam em arquivos de pacotes individuais. Na prática, quase todas as tarefas de gerenciamento de pacotes são executadas pelos comandos apt-get ou apt em sistemas que usam pacotes DEB ou yum ou dnf em sistemas que usam pacotes RPM. Esses comandos funcionam com catálogos de pacotes, podem baixar novos pacotes e suas dependências e verificar se há versões mais recentes dos pacotes instalados.

■ Instalação de Pacotes

Suponha que você tenha ouvido falar sobre um comando chamado figlet, que imprime texto ampliado no terminal e deseja testá-lo. No entanto, você recebe a seguinte mensagem após executar o comando figlet:

```
$ figlet  
-bash: figlet: command not found
```

Isso provavelmente significa que o pacote não está instalado no seu sistema. Se sua distribuição funcionar com pacotes DEB, você poderá procurar em seus repositórios usando o `apt-cache search package_name` ou o `apt search package_name`. O comando `apt-cache` é usado para procurar pacotes e listar informações sobre pacotes disponíveis. O comando a seguir procura por quaisquer ocorrências do termo "figlet" nos nomes e descrições do pacote:

```
$ apt-cache search figlet  
figlet - Make large character ASCII banners out of ordinary text
```



A pesquisa identificou um pacote chamado **figlet** que corresponde ao comando ausente. A instalação e remoção de um pacote requer permissões especiais concedidas apenas ao administrador do sistema: o usuário chamado **root**. Nos sistemas de **desktop**, os usuários comuns podem instalar ou remover pacotes, acrescentando o comando **sudo** aos comandos de instalação / remoção. Isso exigirá que você digite sua senha para continuar. Para pacotes DEB, a instalação é executada com o comando **apt-get install package_name** ou **apt install package_name**:

```
$ sudo apt-get install figlet
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  figlet
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
```

Nesse ponto, o pacote será baixado e instalado no sistema. Quaisquer dependências necessárias para o pacote também serão baixadas e instaladas.

```
Need to get 184 kB of archives.
After this operation, 741 kB of additional disk space will be used.
Get:1 http://archive.raspbian.org/raspbian stretch/main armhf figlet armhf 2.2.5-2
Fetched 184 kB in 0s (213 kB/s)
Selecting previously unselected package figlet.
(Reading database ... 115701 files and directories currently installed.)
Preparing to unpack .../figlet_2.2.5-2_armhf.deb ...
Unpacking figlet (2.2.5-2) ...
Setting up figlet (2.2.5-2) ...
update-alternatives: using /usr/bin/figlet-figlet to provide /usr/bin/figlet (figlet)
Processing triggers for man-db (2.7.6.1-2) .
```

Suponha que você tenha ouvido falar sobre um comando chamado **figlet**, que imprime texto ampliado no terminal e deseja testá-lo. No entanto, você recebe a seguinte mensagem após executar o comando **figlet**:

```
$ figlet
-bash: figlet: command not found
```

Após a conclusão do download, todos os arquivos são copiados para os locais apropriados, qualquer configuração adicional será executada e o comando ficará disponível:

```
$ figlet Awesome!
```



Nas distribuições baseadas nos pacotes RPM, as pesquisas são realizadas usando `yum search package_name` ou `dnf search package_name`. Digamos que você queira exibir algum texto de maneira mais irreverente, seguido por uma vaca de desenho animado, mas não tem certeza do pacote que pode executar essa tarefa. Assim como nos pacotes DEB, os comandos de pesquisa do RPM aceitam termos descritivos:

```
$ yum search speaking cow
Last metadata expiration check: 1:30:49 ago on Tue 23 Apr 2019 11:02:33 PM -03.
=====
Name & Summary Matched: speaking, cow =====
cowsay.noarch : Configurable speaking/thinking cow
```

Depois de encontrar um pacote adequado no repositório, ele pode ser instalado com o `yum install package_name` ou o `dnf install package_name`:

```
$ sudo yum install cowsay
Last metadata expiration check: 2:41:02 ago on Tue 23 Apr 2019 11:02:33 PM -03.
Dependencies resolved.
=====
Package           Arch      Version       Repository      Size
=====
Installing:
cowsay           noarch   3.04-10.fc28   fedora        46 k
Transaction Summary
=====
Install 1 Package

Total download size: 46 k
Installed size: 76 k
Is this ok [y/N]: y
```

Mais uma vez, o pacote desejado e todas as suas dependências possíveis serão baixados e instalados:

```
Downloading Packages:
cowsay-3.04-10.fc28.noarch.rpm          490 kB/s | 46 kB     00:00
=====
Total                                         53 kB/s | 46 kB     00:00

Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing          :                               1/1
Installing         : cowsay-3.04-10.fc28.noarch    1/1
Running scriptlet: cowsay-3.04-10.fc28.noarch    1/1
Verifying          : cowsay-3.04-10.fc28.noarch    1/1

Installed:
cowsay.noarch 3.04-10.fc28

Complete!
```



O comando `cowsay` faz exatamente o que seu nome indica:

```
$ cowsay "Utah Treinamentos, pioneira em Linux no Brasil!"  
< Utah Treinamentos, pioneira em Linux no Brasil! >  
-----  
 \ ^__^  
  (oo)\_____  
   (__)\       )\/\  
    ||----w |  
    ||     ||
```

Embora possam parecer inúteis, os comandos `figlet` e `cowsay` fornecem uma maneira de chamar a atenção de outros usuários para informações relevantes.

```
$ sudo yum install cowsay  
Last metadata expiration check: 2:41:02 ago on Tue 23 Apr 2019 11:02:33 PM -03.  
Dependencies resolved.  
=====  
 Package           Arch      Version       Repository      Size  
=====  
 Installing:  
  cowsay          noarch    3.04-10.fc28    fedora        46 k  
  
Transaction Summary  
=====  
Install 1 Package  
  
Total download size: 46 k  
Installed size: 76 k  
Is this ok [y/N]: y
```

Mais uma vez, o pacote desejado e todas as suas dependências possíveis serão baixados e instalados:

```
Downloading Packages:  
cowsay-3.04-10.fc28.noarch.rpm          490 kB/s | 46 kB   00:00  
=====  
Total                                         53 kB/s | 46 kB   00:00  
  
Running transaction check  
Transaction check succeeded.  
Running transaction test  
Transaction test succeeded.  
Running transaction  
Preparing : 1/1  
Installing  : cowsay-3.04-10.fc28.noarch 1/1  
Running scriptlet: cowsay-3.04-10.fc28.noarch 1/1  
Verifying   : cowsay-3.04-10.fc28.noarch 1/1  
  
Installed:  
cowsay.noarch 3.04-10.fc28  
  
Complete!
```



■ Remoção de Pacotes

Os mesmos comandos usados para instalar pacotes são usados para removê-los. Todos os comandos aceitam a palavra-chave `remove` para desinstalar um pacote instalado: `apt-get remove package_name` ou `apt remove package_name` para pacotes DEB e `yum remove package_name` ou `dnf remove package_name` para pacotes RPM. O comando `sudo` também é necessário para executar a remoção. Por exemplo, para remover o `figlet` do pacote instalado anteriormente de uma distribuição baseada em DEB:

```
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing           : 1/1
Erasing            : cowsay-3.04-10.fc28.noarch 1/1
Running scriptlet: cowsay-3.04-10.fc28.noarch 1/1
Verifying          : cowsay-3.04-10.fc28.noarch 1/1

Removed:
  cowsay.noarch 3.04-10.fc28

Complete!
```

Os arquivos de configuração dos pacotes removidos são mantidos no sistema, para que possam ser usados novamente se o pacote for reinstalado no futuro.

```
$ sudo yum remove cowsay
Dependencies resolved.
=====
 Package      Arch      Version       Repository      Size
 =====
 Removing:
  cowsay      noarch   3.04-10.fc28    @fedora        76

Transaction Summary
=====
 Remove 1 Package
 Freed space: 76 k
 Is this ok [y/N]: y
```

Da mesma forma, uma confirmação é solicitada e o pacote é apagado do sistema:



■ Navegadores

Para a maioria dos usuários, o principal objetivo de um computador é fornecer acesso à Internet. Atualmente, as páginas da web podem atuar como um aplicativo completo, com a vantagem de ser acessível de qualquer lugar, sem a necessidade de instalação de software extra. Isso faz do navegador a aplicação mais importante do sistema operacional, pelo menos para o usuário médio.

Dica:

Uma das melhores fontes para aprender sobre desenvolvimento web é o MDN Web Docs, disponível em <https://developer.mozilla.org/>. Mantido pela Mozilla, o site está cheio de tutoriais para iniciantes e materiais de referência nas mais modernas tecnologias da web.

Os principais navegadores da web no ambiente Linux são o Google Chrome e o Mozilla Firefox. O Chrome é um navegador da web mantido pelo Google, mas é baseado no navegador de código aberto chamado Chromium, que pode ser instalado usando o gerenciador de pacotes da distribuição e é totalmente compatível com o Chrome. Mantido pela Mozilla, uma organização sem fins lucrativos, o Firefox é um navegador cujas origens estão vinculadas ao Netscape, o primeiro navegador da web popular a adotar o modelo de código aberto. A Mozilla Foundation está profundamente envolvida com o desenvolvimento de padrões abertos subjacentes à web moderna.

A Mozilla também desenvolve outros aplicativos, como o cliente de email Thunderbird. Muitos usuários optam por usar o webmail em vez de um aplicativo de email dedicado, mas um cliente como o Thunderbird oferece recursos extras e se integra melhor a outros aplicativos na área de trabalho.

■ Multimedia

Comparado aos aplicativos da Web disponíveis, os aplicativos de desktop ainda são a melhor opção para a criação de conteúdo multimídia. As atividades relacionadas a multimídia, como a renderização de vídeo, geralmente exigem grandes quantidades de recursos do sistema, que são melhor gerenciados por um aplicativo de desktop local. Alguns dos aplicativos multimídia mais populares para o ambiente Linux e seus usos estão listados abaixo.

Blender	Um renderizador 3D para criar animações. O Blender também pode ser usado para exportar objetos 3D a serem impressos por uma impressora 3D.
GIMP	Um editor de imagens completo, que pode ser comparado ao Adobe Photoshop, mas possui conceitos e ferramentas próprios para trabalhar com imagens. O GIMP pode ser usado para criar, editar e salvar a maioria dos arquivos de bitmap, como JPEG, PNG, GIF, TIFF e muitos outros.
Inkscape	Um editor de gráficos vetoriais, semelhante ao Corel Draw ou Adobe Illustrator. O formato padrão do Inkscape é SVG, que é um padrão aberto para gráficos vetoriais. Os arquivos SVG podem ser abertos por qualquer navegador da web e, devido à sua natureza como gráfico vetorial, podem ser usados em layouts flexíveis de páginas da web.
Audacity	Um editor de áudio. O Audacity pode ser usado para filtrar, aplicar efeitos e converter entre diversos formatos de áudio, como MP3, WAV, OGG, FLAC, etc.
ImageMagick	O ImageMagick é uma ferramenta de linha de comando para converter e editar a maioria dos tipos de arquivos de imagem. Também pode ser usado para criar documentos PDF a partir de arquivos de imagem e vice-versa.

Existem também muitos aplicativos dedicados à reprodução de multimídia. O aplicativo mais popular para reprodução de vídeo é o VLC, mas alguns usuários preferem outras alternativas, como o smplayer. A reprodução de música local também tem muitas opções, como Audacious, Banshee e Amarok, que também podem gerenciar uma coleção de



■ Servidores

Quando um navegador da Web carrega uma página de um site, na verdade ele se conecta a um computador remoto e solicita uma informação específica. Nesse cenário, o computador executando o navegador da Web é chamado de cliente e o computador remoto é chamado de servidor.

O computador servidor, que pode ser um computador comum ou hardware especializado, precisa de um programa específico para gerenciar cada tipo de informação que ele fornecerá. Em relação à veiculação de páginas da web, a maioria dos servidores em todo o mundo implementa programas de servidor de código aberto. Este programa de servidor específico é chamado de servidor HTTP (**HTTP significa Hyper Text Transfer Protocol**) e os mais populares são **Apache**, **Nginx** e **lighttpd**.

Até páginas da web simples podem exigir muitas solicitações, que podem ser arquivos comuns- chamado conteúdo estático- ou conteúdo dinâmico renderizado de várias fontes. A função de um servidor HTTP é coletar e enviar todos os dados solicitados de volta ao navegador, que organiza o conteúdo conforme definido pelo documento HTML recebido (**HTML significa Hyper Text Markup Language**) e outros arquivos de suporte. Portanto, a renderização de uma página da web envolve operações executadas no lado do servidor e operações executadas no lado do cliente. Ambos os lados podem usar scripts personalizados para realizar tarefas específicas. No lado do servidor HTTP, é bastante comum usar a linguagem de script PHP. JavaScript é a linguagem de script usada no lado do cliente (o navegador da web).

Os programas de servidor podem fornecer todos os tipos de informações. Não é incomum ter um programa de servidor solicitando informações fornecidas por outros programas de servidor. Esse é o caso quando um servidor HTTP requer informações fornecidas por um servidor de banco de dados.

■ Servidor WEB

Por exemplo, quando uma página dinâmica é solicitada, o servidor HTTP normalmente consulta um banco de dados para coletar todas as informações necessárias e envia o conteúdo dinâmico de volta ao cliente. De maneira semelhante, quando um usuário se registra em um site, o servidor HTTP reúne os dados enviados pelo cliente e os armazena em um banco de dados.

■ Servidor Banco de Dados

Um banco de dados é um conjunto organizado de informações. Um servidor de banco de dados armazena o conteúdo de forma formatada, possibilitando a leitura, gravação e vinculação de grandes quantidades de dados de maneira confiável e em alta velocidade. Os servidores de banco de dados de código aberto são usados em muitos aplicativos, não apenas na Internet. Até aplicativos locais podem armazenar dados conectando-se a um servidor de banco de dados local. O tipo mais comum de banco de dados é o banco de dados relacional, onde os dados são organizados em tabelas predefinidas. Os bancos de dados relacionais de código aberto mais populares são o MariaDB (originário do MySQL) e o PostgreSQL.

■ Servidor de Armazenamento de Dados

Nas redes locais, como as encontradas em escritórios e residências, é desejável que os computadores não apenas possam acessar a Internet, mas também que possam se comunicar. Às vezes, um computador atua como servidor, às vezes, o mesmo computador como cliente. Isso é necessário quando se deseja acessar arquivos em outro computador na rede - por exemplo, acessar um arquivo armazenado em um computador desktop a partir de um dispositivo portátil- sem a necessidade de copiá-lo para uma unidade USB ou algo semelhante.



■ Servidor de Armazenamento de Dados

Entre máquinas Linux, o NFS (sistema de arquivos de rede) é frequentemente usado. O protocolo NFS é a maneira padrão de compartilhar sistemas de arquivos em redes equipadas apenas com máquinas Unix / Linux. Com o NFS, um computador pode compartilhar um ou mais de seus diretórios com computadores específicos na rede, para que eles possam ler e gravar arquivos nesses diretórios. O NFS pode até ser usado para compartilhar uma árvore de diretórios de todo o sistema operacional com clientes que o usarão para inicializar. Esses computadores, chamados thin clients, geralmente são usados em grandes redes para evitar a manutenção de cada sistema operacional individual em cada máquina.

Se houver outros tipos de sistemas operacionais conectados à rede, é recomendável usar um protocolo de compartilhamento de dados que possa ser entendido por todos eles. Este requisito é cumprido pelo Samba. O Samba implementa um protocolo para compartilhar arquivos na rede originalmente criado para o sistema operacional Windows, mas hoje é compatível com todos os principais sistemas operacionais. Com o Samba, os computadores na rede local não apenas podem compartilhar arquivos, mas também impressoras.

Em algumas redes locais, a autorização concedida no login em uma estação de trabalho é concedida por um servidor central, chamado controlador de domínio, que gerencia o acesso a vários recursos locais e remotos. O controlador de domínio é um serviço fornecido pelo Active Directory da Microsoft. As estações de trabalho Linux podem se associar a um controlador de domínio usando o Samba ou um subsistema de autenticação chamado SSSD. A partir da versão 4, o Samba também pode funcionar como um controlador de domínio em redes heterogêneas.

Se o objetivo é implementar uma solução de computação em nuvem capaz de fornecer vários métodos de compartilhamento de dados baseados na Web, duas alternativas devem ser consideradas: ownCloud e Nextcloud. Os dois projetos são muito parecidos porque o Nextcloud é um spin-off do ownCloud, o que não é incomum entre os projetos de código aberto. Tais spin-offs são geralmente chamados de garfo. Ambos oferecem os mesmos recursos básicos: compartilhamento e sincronização de arquivos, espaços de trabalho colaborativos, calendário, contatos e correio, através de interfaces de desktop, móveis e web. O Nextcloud também fornece conferência de áudio / vídeo privada, enquanto o ownCloud está mais focado no compartilhamento e integração de arquivos com software de terceiros. Muitos outros recursos são fornecidos como plug-ins, que podem ser ativados posteriormente, conforme necessário.

O ownCloud e o Nextcloud oferecem uma versão paga com recursos extras e suporte estendido. O que os diferencia de outras soluções comerciais é a capacidade de instalar o Nextcloud ou o ownCloud em um servidor privado, gratuitamente, evitando manter dados confidenciais em um servidor desconhecido. Como todos os serviços dependem da comunicação HTTP e são gravados em PHP, a instalação deve ser executada em um servidor Web configurado anteriormente, como o Apache. Se você considerar instalar o ownCloud ou Nextcloud em seu próprio servidor, ative também o HTTPS para criptografar todas as conexões com sua nuvem.

■ Administração de Redes Linux

A comunicação entre computadores só é possível se a rede estiver funcionando corretamente. Normalmente, a configuração de rede é feita por um conjunto de programas em execução no roteador, responsável por instalar e verificar a disponibilidade da rede. Para isso, são utilizados dois serviços básicos de rede: DHCP (Dynamic Host Configuration Protocol) e DNS (Domain Name System).

O DHCP é responsável por atribuir um endereço IP ao host quando um cabo de rede está conectado ou quando o dispositivo entra em uma rede sem fio. Ao se conectar à Internet, o servidor DHCP do ISP fornecerá um endereço IP ao dispositivo solicitante. Um servidor DHCP também é muito útil em redes locais, para fornecer automaticamente endereços IP a todos os dispositivos conectados. Se o DHCP não estiver configurado ou se não estiver funcionando corretamente, seria necessário configurar manualmente o endereço IP de cada dispositivo conectado à rede. Não é prático definir manualmente os endereços IP em redes grandes ou mesmo em redes pequenas; é por isso que a maioria dos roteadores de rede vem com um servidor DHCP pré-configurado por padrão.



■ Administração de Redes Linux

O endereço IP é necessário para se comunicar com outro dispositivo em uma rede IP, mas nomes de domínio como www.lpi.org têm muito mais probabilidade de serem lembrados do que um número IP como 203.0.113.165. O nome de domínio por si só, no entanto, não é suficiente para estabelecer a comunicação através da rede. É por isso que o nome do domínio precisa ser traduzido para um endereço IP por um servidor DNS. O endereço IP do servidor DNS é fornecido pelo servidor DHCP do ISP e é usado por todos os sistemas conectados para converter nomes de domínio em endereços IP.

As configurações de DHCP e DNS podem ser modificadas inserindo a interface da web fornecida pelo roteador. Por exemplo, é possível restringir a atribuição de IP apenas a dispositivos conhecidos ou associar um endereço IP fixo a máquinas específicas. Também é possível alterar o servidor DNS padrão fornecido pelo ISP. Alguns servidores DNS de terceiros, como os fornecidos pelo Google ou OpenDNS, às vezes podem fornecer respostas mais rápidas e recursos extras.

■ Linguagens de Programação

Todos os programas de computador (programas de cliente e servidor, aplicativos de desktop e o próprio sistema operacional) são criados usando uma ou mais linguagens de programação. Os programas podem ser um único arquivo ou um sistema complexo de centenas de arquivos, que o sistema operacional trata como uma sequência de instruções a serem interpretadas e executadas pelo processador e outros dispositivos.

Existem inúmeras linguagens de programação para propósitos muito diferentes e os sistemas Linux fornecem muitas delas. Como o software de código aberto também inclui as fontes dos programas, os sistemas Linux oferecem aos desenvolvedores condições perfeitas para entender, modificar ou criar software de acordo com suas próprias necessidades.

Todo programa começa como um arquivo de texto, chamado código fonte. Este código fonte é escrito em uma linguagem mais ou menos amigável para o ser humano que descreve o que o programa está fazendo. Um processador de computador não pode executar diretamente esse código. Em linguagens compiladas, o código fonte é, portanto, convertido em um arquivo binário que pode ser executado pelo computador. Um programa chamado compilador é responsável por fazer a conversão do código fonte para o formato executável. Como o binário compilado é específico para um tipo de processador, o programa pode precisar ser recompilado para ser executado em outro tipo de computador.

Nas linguagens interpretadas, o programa não precisa ser compilado anteriormente. Em vez disso, um intérprete lê o código fonte e executa suas instruções toda vez que o programa é executado. Isso torna o desenvolvimento mais fácil e rápido, mas, ao mesmo tempo, os programas interpretados tendem a ser mais lentos que os programas compilados.

Aqui estão algumas das linguagens de programação mais populares:

JavaScript

JavaScript é uma linguagem de programação usada principalmente em páginas da web. Em sua origem, os aplicativos JavaScript eram muito simples, como rotinas de validação de formulário. Hoje, o JavaScript é considerado uma linguagem de primeira classe e é usado para criar aplicativos muito complexos, não apenas na Web, mas em servidores e dispositivos móveis.

C

A linguagem de programação C está intimamente relacionada aos sistemas operacionais, particularmente ao Unix, mas é usada para gravar qualquer tipo de programa em quase qualquer tipo de dispositivo. As grandes vantagens de C são flexibilidade e velocidade. O mesmo código fonte escrito em C pode ser compilado para ser executado em diferentes plataformas e sistemas operacionais, com pouca ou nenhuma modificação. Depois de compilado, no entanto, o programa será executado apenas no sistema de destino.

Java

O principal aspecto do Java é que os programas escritos nessa linguagem são portáteis, o que significa que o mesmo programa pode ser executado em diferentes sistemas operacionais. Apesar do nome, Java não está relacionado ao JavaScript.



```
$ sudo yum remove cowsay
Dependencies resolved.
=====
 Package          Arch      Version       Repository      Size
 =====
 Removing:
 cowsay           noarch   3.04-10.fc28  @fedora        76
 Transaction Summary
 =====
 Remove 1 Package
 Freed space: 76 k
 Is this ok [y/N]: y
```

■ Aplicativos para Desktop

Os aplicativos do Office são usados para editar arquivos, como textos, apresentações, planilhas e outros formatos comumente usados em um ambiente de escritório. Esses aplicativos geralmente são organizados em coleções chamadas suítes de escritório.

Por um longo tempo, o pacote de escritório mais usado no Linux foi o pacote OpenOffice.org. O OpenOffice.org era uma versão de código aberto do pacote StarOffice lançado pela Sun Microsystems. Alguns anos depois, a Sun foi adquirida pela Oracle Corporation, que por sua vez transferiu o projeto para a Apache Foundation e o OpenOffice.org foi renomeado para Apache OpenOffice. Enquanto isso, outro pacote de escritório baseado no mesmo código-fonte foi lançado pela Document Foundation, que o chamou de LibreOffice.

Os dois projetos têm os mesmos recursos básicos e são compatíveis com os formatos de documento do Microsoft Office. No entanto, o formato de documento preferido é o Open Document Format, um formato de arquivo totalmente aberto e padronizado ISO. O uso de arquivos ODF garante que os documentos possam ser transferidos entre sistemas operacionais e aplicativos de diferentes fornecedores, como o Microsoft Office. Os principais aplicativos oferecidos pelo OpenOffice / LibreOffice são:

Writer	Editor de Texto
Calc	Planilhas
Impress	Apresentações
Draw	Desenho Vetorial
Math	Fórmulas Matemáticas
Base	Base de Dados

O LibreOffice e o Apache OpenOffice são softwares de código aberto, mas o LibreOffice é licenciado sob LGPLv3 e o Apache OpenOffice é licenciado sob a Apache License 2.0. A distinção de licenciamento implica que o LibreOffice pode incorporar melhorias feitas pelo Apache OpenOffice, mas o Apache OpenOffice não pode incorporar melhorias feitas pelo LibreOffice. Isso e uma comunidade mais ativa de desenvolvedores são a razão pela qual a maioria das distribuições adota o LibreOffice como suíte de escritório padrão.



Python

Python é uma linguagem de programação muito popular entre estudantes e profissionais não diretamente envolvidos com a ciência da computação. Embora tenha recursos avançados, o Python é uma boa maneira de começar a aprender programação por sua abordagem fácil de usar.

PHP

O PHP é mais usado como uma linguagem de script no servidor para gerar conteúdo para a web. A maioria das páginas HTML online não são arquivos estáticos, mas conteúdo dinâmico gerado pelo servidor a partir de várias fontes, como bancos de dados. Programas PHP- às vezes chamados apenas de páginas PHP ou scripts PHP- são frequentemente usados para gerar esse tipo de conteúdo. O termo LAMP vem da combinação de um sistema operacional Linux, um servidor HTTP Apache, um banco de dados MySQL (ou MariaDB) e programação PHP. Servidores LAMP são uma solução muito popular para a execução de servidores web. Além do PHP, todas as linguagens de programação descritas anteriormente também podem ser usadas para implementar esses aplicativos.

Shell

O shell, particularmente o shell Bash, não é apenas uma linguagem de programação, mas uma interface interativa para executar outros programas. Os programas shell, conhecidos como scripts shell, podem automatizar tarefas complexas ou repetitivas no ambiente da linha de comandos.

■ Exercícios:

- 01 Para cada um dos seguintes comandos, identifique se está associado ao sistema de empacotamento Debian ou ao sistema de empacotamento Red Hat:

dpkg	=
rpm	=
apt-get	=
yum	=
dnf	=
aptitude	=

- 02 Qual comando pode ser usado para instalar o Blender no Ubuntu? Após a instalação, como o programa pode ser executado?

- 03 Qual aplicativo do pacote LibreOffice pode ser usado para trabalhar com planilhas eletrônicas?

- 04 Qual navegador de código aberto é usado como base para o desenvolvimento do Google Chrome?

- 05 SVG é um padrão aberto para gráficos vetoriais. Qual é o aplicativo mais popular para editar arquivos SVG em sistemas Linux?

- 06 Para cada um dos seguintes formatos de arquivo, escreva o nome de um aplicativo capaz de abrir e editar o arquivo correspondente:

png	=
doc	=
xls	=
ppt	=
wav	=
svg	=

- 07 Qual pacote de software permite o compartilhamento de arquivos entre máquinas Linux e Windows pela rede local?



Peso	1
Descrição	Comunidades abertas e licenciamento de software de código aberto para empresas.
Áreas de conhecimento	<ul style="list-style-type: none">▫ Filosofia de Código Aberto▫ Licenciamento de Código Aberto▫ Fundação do Software Livre (FSF), Open Source Initiative (OSI)
Itens relacionados	<ul style="list-style-type: none">▫ Copyleft, permissivo▫ GPL, BSD, Creative Commons▫ Software Livre, Software de Código Aberto, FOSS, FLOSS▫ Modelos de negócios de código aberto



■ Introdução

Embora os termos software livre e software de código aberto sejam amplamente utilizados, ainda existem alguns conceitos errados sobre seu significado. Em particular, o conceito de "liberdade" precisa de um exame mais detalhado. Vamos começar com a definição dos dois termos

■ Definição de Software Livre e de Código Aberto

Critérios de Software Livre

Antes de tudo, "livre" no contexto do software livre não tem nada a ver com "gratuitamente", ou como o fundador da Free Software Foundation (FSF), Richard Stallman, coloca sucintamente:

Para entender o conceito, você deve pensar em "livre" como em "liberdade de expressão", não como em "cerveja grátis".
- Richard Stallman

■ O que é Software Livre

Independentemente de você ter que pagar pelo software ou não, existem quatro critérios que constituem o software livre. Richard Stallman descreve esses critérios como "as quatro liberdades essenciais", cuja contagem começa do zero:

LIBERDADE 0

"A liberdade de executar o programa como desejar, para qualquer finalidade"
Onde, como e com que finalidade o software é usado não pode ser prescrito nem restrito.

LIBERDADE 1

"A liberdade de estudar como o programa funciona, e alterá-lo para que você faça a sua computação como desejar. O acesso ao código-fonte é uma condição prévia para isso. Todos podem alterar o software de acordo com suas idéias e necessidades. Por sua vez, isso pressupõe que o chamado código-fonte, ou seja, todos os arquivos dos quais um software consiste, deve estar disponível em um formato legível pelos programadores. E, é claro, esse direito se aplica a um único usuário que pode querer adicionar um único recurso, bem como a empresas de software que constroem sistemas complexos, como sistemas operacionais de smartphones ou firmware de roteador.

LIBERDADE 2

"A liberdade de redistribuir cópias para que você possa ajudar outras pessoas "
Essa liberdade incentiva explicitamente cada usuário a compartilhar o software com os outros. Portanto, é uma questão da distribuição mais ampla possível e, portanto, da comunidade mais ampla possível de usuários e desenvolvedores que, com base nessas liberdades, desenvolvem e melhoram ainda mais o software para o benefício de todos.

LIBERDADE 4

"A liberdade de distribuir cópias de suas versões modificadas para outras pessoas.
Ao fazer isso, você pode dar a toda a comunidade a chance de se beneficiar de suas alterações. O acesso ao código-fonte é uma condição prévia para isso. "

Não se trata apenas da distribuição de software livre, mas da distribuição de software livre modificado. Qualquer pessoa que faça alterações no software livre tem o direito de disponibilizá-las para outras pessoas. Se o fizerem, também são obrigados a fazê-lo livremente, ou seja, não devem restringir as liberdades originais ao distribuir o software, mesmo se o modificaram ou estenderam. Por exemplo, se um grupo de desenvolvedores tiver idéias diferentes sobre a direção de um software específico que os autores originais, ele poderá dividir seu próprio ramo de desenvolvimento (chamado de fork) e continuar como um novo projeto. Mas, é claro, todas as obrigações associadas a essas liberdades permanecem.



■ Open Source vs Free Software

Para muitos, software livre e software de código aberto são sinônimos. A abreviação frequentemente usada FOSS para Software Livre e de Código Aberto enfatiza esse ponto em comum. O software FLOSS for Free / Libre e Open Source é outro termo popular, que enfatiza inequivocamente a idéia de liberdade também para outros idiomas além do inglês. No entanto, se considerarmos a origem e o desenvolvimento de ambos os termos, uma diferenciação vale a pena.

O termo software livre, com a definição das quatro liberdades descritas, remonta a Richard Stallman e ao projeto GNU fundado por ele em 1985- quase 10 anos antes do surgimento do Linux. O nome “GNU não é Unix” descreve a intenção com um piscar de olhos: o GNU começou como uma iniciativa para desenvolver uma solução tecnicamente convincente- ou seja, o sistema operacional Unix- a partir do zero, para disponibilizá-lo ao público em geral e melhorar continuamente com o público em geral. A abertura do código fonte era apenas uma necessidade técnica e organizacional para isso, mas, em sua auto-imagem, o movimento do software livre ainda é um movimento social e político- alguns também dizem ideológico.

Com o sucesso do Linux, as possibilidades colaborativas da Internet e os milhares de projetos e empresas que surgiram nesse novo cosmos de software, o aspecto social se voltou cada vez mais para segundo plano. A abertura do próprio código-fonte mudou de um requisito técnico para um recurso definidor: assim que o código-fonte ficou visível, o software foi considerado “código-fonte aberto”. Os motivos sociais deram lugar a uma abordagem mais pragmática do desenvolvimento de software.

Software livre e software de código aberto funcionam da mesma maneira, com os mesmos métodos e em uma comunidade mundial de indivíduos, projetos e empresas. Mas desde que eles se uniram a partir de diferentes direções- uma social e outra técnica pragmática-, às vezes existem conflitos. Esses conflitos surgem quando os resultados do trabalho conjunto não correspondem aos objetivos originais de ambos os movimentos. Isso acontece especialmente quando o software revela suas fontes, mas não respeita as quatro liberdades do software livre ao mesmo tempo, por exemplo, quando há restrições à divulgação, alteração ou conexão com outros componentes de software.

A licença sob a qual o software está disponível determina a quais condições um software está sujeito em relação ao uso, distribuição e modificação. E como os requisitos e motivos podem ser muito diferentes, inúmeras licenças diferentes foram criadas na área de software livre. Devido à abordagem muito mais fundamental do movimento do software livre, não é surpreendente que ele não reconheça muitas licenças de código aberto como "gratuitas" e, portanto, as rejeite. Por outro lado, esse não é o caso, devido à abordagem de código aberto muito mais pragmática.

Vamos dar uma breve olhada na área realmente complexa das licenças abaixo.

■ Licenças

Ao contrário de uma geladeira ou carro, o software não é um produto físico, mas um produto digital. Assim, uma empresa não pode realmente transferir a propriedade de um produto vendendo-o e alterando a posse física- em vez disso, transfere os direitos de uso para esse produto e o usuário concorda contratualmente com esses direitos de uso. Quais direitos de uso são e, acima de tudo, não são registrados na licença do software e, portanto, torna-se compreensível a importância dos regulamentos contidos nela.

Embora grandes fornecedores de software proprietário, como Microsoft ou SAP, possuam suas próprias licenças adaptadas com precisão aos seus produtos, os defensores do software livre e de código aberto se esforçaram desde o início por clareza e validade geral de suas licenças, porque, afinal, todo usuário deve entendê-los e, se necessário, usá-los para seus próprios desenvolvimentos.

No entanto, não se deve esconder que esse ideal de simplicidade dificilmente pode ser alcançado porque muitos requisitos específicos e internacionalmente nem sempre os entendimentos legais compatíveis impedem isso. Para dar apenas um exemplo: as leis de direitos autorais alemãs e americanas são fundamentalmente diferentes. Segundo a lei alemã, há uma pessoa como autor (mais precisamente: Urheber), cuja obra é sua propriedade intelectual. Embora o autor possa conceder permissão para usar sua obra, ele não pode atribuir ou desistir de sua autoria. Este último é estranho à lei americana. Aqui também existe um autor (que, no entanto, também pode ser uma empresa ou uma instituição), mas ele só possui direitos de exploração que ele pode transferir em parte ou na íntegra e, portanto, se destaca completamente de seu trabalho. Uma licença válida internacionalmente deve ser interpretada com relação a diferentes legislações.

As consequências são numerosas e, às vezes, muito diferentes licenças de software livre. Pior, ainda, são versões diferentes de uma licença ou uma mistura de licenças (dentro de um projeto, ou mesmo ao conectar vários projetos) que podem causar confusão ou até disputas legais.



■ *Licenças*

Tanto os representantes do software livre quanto os defensores do movimento de código aberto claramente orientado para a economia criaram suas próprias organizações, que hoje são decisivamente responsáveis pela formulação de licenças de software de acordo com seus princípios e apoiam seus membros em sua aplicação.

■ *CopyLeft*

A já mencionada Free Software Foundation (FSF) formulou a GNU General Public License (GPL) como uma das licenças mais importantes para o software livre, usada por muitos projetos, por exemplo, o kernel do Linux. Além disso, lançou licenças com personalizações específicas de cada caso, como a Licença Pública Geral Menor GNU (LGPL), que governa a combinação de software livre com menos componentes livres, a Licença Pública Geral GNU Affero (AGPL), que abrange vendas acesso ao software hospedado, ou a GNU Free Documentation License (FDL), que estende os princípios de liberdade à documentação do software. Além disso, a FSF faz recomendações a favor ou contra licenças de terceiros, e projetos afiliados, como GPL-Violations.org, investigam suspeitas de violações de licenças gratuitas.

A FSF chama o princípio segundo o qual uma licença gratuita também se aplica a variantes modificadas do copyleft de software - em contraste com o princípio de direitos autorais restritivos que ele rejeita. A idéia, portanto, é transferir os princípios liberais de uma licença de software o mais irrestritamente possível para futuras variantes do software, a fim de evitar restrições subsequentes.

O que parece óbvio e simples, no entanto, leva a complicações consideráveis na prática, razão pela qual os críticos costumam chamar o princípio copyleft de "viral", uma vez que é transmitido para versões subsequentes.

Pelo que foi dito, segue-se, por exemplo, que dois componentes de software licenciados sob diferentes licenças copyleft podem não ser combináveis entre si, pois ambas as licenças não podem ser transferidas para o produto subsequente ao mesmo tempo. Isso pode até se aplicar a diferentes versões da mesma licença!

Por esse motivo, as licenças mais recentes ou versões de licença geralmente não compreendem mais o copyleft com tanto rigor. A mencionada Licença Pública Geral Menor (LGPL) do GNU já é, neste sentido, uma concessão para poder conectar software livre com componentes "não-livres", como é frequentemente feito com as chamadas bibliotecas. As bibliotecas contêm sub-rotinas ou rotinas, que por sua vez são usadas por vários outros programas. Isso leva à situação comum em que o software proprietário chama essa sub-rotina de uma biblioteca gratuita.

Outra maneira de evitar conflitos de licença é o licenciamento duplo, em que um software é licenciado sob licenças diferentes, por exemplo uma licença gratuita e uma licença proprietária. Um caso de uso típico é uma versão gratuita de um software que só pode ser usada quando respeitar as restrições de copyleft e a oferta alternativa de obter o software sob uma licença diferente que libera o licenciado de determinadas restrições em troca de uma taxa que pode ser usada para financiar o desenvolvimento do software.

Portanto, deve ficar claro que a escolha da licença para projetos de software deve ser feita com muita cautela, uma vez que a cooperação com outros projetos, a combinabilidade com outros componentes e também o design futuro do próprio produto dependem disso. O copyleft apresenta aos desenvolvedores desafios especiais a esse respeito.

■ *Definição de Código Aberto e Licenças Permissivas*

No lado do código aberto, é a Open Source Initiative (OSI), fundada em 1998 por Eric S. Raymond e Bruce Perens, que se preocupa principalmente com questões de licenciamento. Também desenvolveu um procedimento padronizado para verificar as licenças de software quanto à conformidade com sua Definição de Código Aberto. Atualmente, mais de 80 licenças de código aberto reconhecidas podem ser encontradas no site da OSI.

Aqui, eles também listam as licenças como "aprovadas pela OSI" que contradizem explicitamente o princípio copyleft, especialmente o grupo de licenças BSD. O Berkeley Software Distribution (BSD) é uma variante do sistema operacional Unix originalmente desenvolvido na Universidade de Berkeley, que mais tarde deu origem a projetos gratuitos como o NetBSD, FreeBSD e OpenBSD. As licenças subjacentes a esses projetos são frequentemente denominadas permissivas. Ao contrário das licenças copyleft, elas não têm o objetivo de estabelecer os termos de uso de variantes modificadas. Em vez disso, a liberdade máxima deve ajudar o software a ser o mais amplamente distribuído possível, deixando os editores sozinhos a decidir como proceder com as edições - se, por exemplo, eles também os liberam ou os tratam como fonte fechada e distribuem comercialmente.



■ Definição de Código Aberto e Licenças Permissivas

A Licença BSD de 2 Cláusulas, também chamada Licença Simplificada BSD ou Licença FreeBSD, prova o quanto reduzida pode ser uma licença permissiva. Além da cláusula de responsabilidade padronizada, que protege os desenvolvedores de reivindicações de responsabilidade decorrentes de danos causados pelo software, a licença consiste apenas nas duas regras a seguir:

A redistribuição e uso nas formas de origem e binárias, com ou sem modificação, são permitidas desde que as seguintes condições sejam atendidas:

As redistribuições do código-fonte devem manter o aviso de direitos autorais acima, esta lista de condições e o aviso de isenção de responsabilidade a seguir.

As redistribuições em formato binário devem reproduzir o aviso de direitos autorais acima, esta lista de condições e o aviso de isenção de responsabilidade a seguir na documentação e / ou outros materiais fornecidos com a distribuição.

■ Creative Commons

O conceito de desenvolvimento bem-sucedido do FLOSS e o progresso tecnológico associado levaram a tentativas de transferir o princípio do código aberto para outras áreas não técnicas. A preparação e o fornecimento de conhecimento, bem como a cooperação criativa na resolução de tarefas complexas, são agora considerados evidências do princípio estendido do código aberto relacionado ao conteúdo.

Isso levou à necessidade de criar fundações confiáveis ​​também nessas áreas, segundo as quais os resultados do trabalho podem ser compartilhados e processados. Como as licenças de software disponíveis dificilmente eram adequadas para isso, houve inúmeras tentativas de converter os requisitos específicos do trabalho científico em obras de arte digitalizadas "no espírito do código aberto" em licenças igualmente úteis.

De longe, a iniciativa mais importante desse tipo hoje é o Creative Commons (CC), que resume suas preocupações da seguinte forma:

A Creative Commons é uma organização global sem fins lucrativos que permite compartilhar e reutilizar a criatividade e o conhecimento através do fornecimento de ferramentas legais gratuitas.

- <https://creativecommons.org/faq/#what-is-creative-commons-and-what-do-you-do>

Com o Creative Commons, o foco da atribuição de direitos remonta ao distribuidor e ao autor. Um exemplo: na publicação tradicional, um autor geralmente transfere todos os direitos de publicação (impressão, tradução etc.) para um editor, que, por sua vez, garante a melhor distribuição possível da obra. Os canais de distribuição significativamente alterados da Internet agora colocam o autor em posição de exercer muitos desses direitos de publicação e decidir por si próprio como seu trabalho pode ser usado. O Creative Commons oferece a oportunidade de determinar isso de maneira simples e legal, mas o Creative Commons quer mais: os autores são incentivados a disponibilizar seus trabalhos como uma contribuição para um processo geral de intercâmbio e cooperação.

Diferentemente dos direitos autorais tradicionais, que concedem ao autor todos os direitos que eles podem transferir para os outros, conforme necessário, a abordagem Creative Commons adota a abordagem oposta: o autor disponibiliza seu trabalho para a comunidade, mas pode escolher dentre um conjunto de recursos aqueles que precisam a ser considerado ao usar o trabalho - quanto mais recursos ela escolher, mais restritiva será a licença.

E, assim, o princípio "Escolha uma licença" do CC solicita a um autor passo a passo as propriedades individuais e gera a licença recomendada, que o autor pode atribuir pela última vez ao trabalho como texto e ícone.

Para uma melhor compreensão, aqui está uma visão geral das seis combinações e licenças possíveis oferecidas pela CC:

CC BY ("Atribuição")

A licença gratuita que permite a qualquer pessoa editar e distribuir o trabalho, desde que o nome seja do autor.

CC BY-SA ("Atribuição-Compartilhamento pela mesma licença")

Como CC BY, exceto que o trabalho modificado só pode ser distribuído sob a mesma licença. O princípio lembra o copyleft, porque a licença é "herdada" aqui também.



■ Creative Commons

CC BY-ND ("Atribuição-Não-Derivadas")

Como o CC BY, exceto que o trabalho só pode ser transmitido sem modificações.

CC BY-NC ("Atribuição-Não-Comercial")

O trabalho pode ser editado e distribuído com a nomeação do autor, mas somente em condições não comerciais.

CC BY-NC-SA ("Atribuição-Uso Não-Comercial-Compartilhamento pela mesma Licença")

Como BY-NC, exceto que o trabalho só pode ser compartilhado sob as mesmas condições (ou seja, uma licença semelhante a copyleft).

CC BY-NC-ND ("Atribuição-Não-Comercial-Não-Derivadas")

A licença mais restritiva: a distribuição é permitida com a atribuição do autor, mas apenas inalterada e em condições não comerciais.

■ Modelos de Negócios em Código Aberto

Em retrospecto, o triunfo do FLOSS atua como um movimento de base de idealistas tecnófilos que, independentemente de restrições econômicas e livres de dependências monetárias, colocam seu trabalho a serviço do público em geral. Ao mesmo tempo, empresas no valor de bilhões foram criadas no ambiente FLOSS; para citar apenas um, a empresa norte-americana Red Hat fundada em 1993 com vendas anuais de mais de 3 bilhões de dólares (2018), que foi adquirida pela gigante de TI IBM em 2018.

Então, vamos dar uma olhada na tensão entre a distribuição gratuita e gratuita de software de alta qualidade e os modelos de negócios para seus criadores, porque uma coisa deve ficar clara: os inúmeros desenvolvedores altamente qualificados de software livre também devem ganhar o dinheiro e o ambiente FLOSS originalmente puramente não comercial devem, portanto, desenvolver modelos de negócios sustentáveis para preservar seu próprio cosmos.

Uma abordagem comum, especialmente para projetos maiores na fase inicial, é o chamado crowdfunding, ou seja, a coleta de doações em dinheiro por meio de uma plataforma como o Kickstarter. Em troca, os doadores recebem um bônus predefinido dos desenvolvedores em caso de sucesso, ou seja, se objetivos previamente definidos forem atingidos, seja acesso ilimitado ao produto ou recursos especiais.

Outra abordagem é o licenciamento duplo: o software livre é oferecido em paralelo sob uma licença mais restritiva ou até proprietária, o que garante ao cliente serviços mais abrangentes (tempos de resposta em caso de erros, atualizações, versões para determinadas plataformas etc.). Um exemplo entre muitos é o ownCloud, que está sendo desenvolvido sob a GPL e oferece aos clientes comerciais uma "Business Edition" sob uma licença proprietária.

Tomemos também o ownCloud como exemplo de outro modelo de negócios amplamente difundido do FLOSS: serviços profissionais. Muitas empresas não possuem o conhecimento técnico interno necessário para configurar e operar software complexo e crítico de maneira confiável e, acima de tudo, segura. É por isso que eles compram serviços profissionais, como consultoria, manutenção ou suporte técnico, diretamente do fabricante. As questões de responsabilidade também desempenham um papel nessa decisão, pois a empresa transfere os riscos de operação para o fabricante.

Se um software consegue se tornar bem-sucedido e popular em seu campo, são possibilidades de monetização periférica, como merchandising ou certificados que os clientes adquirem, e, portanto, apontam seu status especial ao usar esse software. A plataforma de aprendizado Moodle oferece a certificação de treinadores, que documentam seus conhecimentos para clientes em potencial, por exemplo, e este é apenas um exemplo entre muitos outros.

O software como serviço (SaaS) é outro modelo de negócios, especialmente para tecnologias baseadas na Web. Aqui, um provedor de nuvem executa um software como um CRM (Customer Relationship Management) ou um CMS (Content Management System) em seus servidores e concede a seus clientes acesso ao aplicativo instalado. Isso economiza a instalação e manutenção do software pelo cliente. Em troca, o cliente paga pelo uso do software de acordo com vários parâmetros, por exemplo, o número de usuários. Disponibilidade e segurança desempenham um papel importante como fatores críticos para os negócios.

Por fim, mas não menos importante, o modelo de desenvolvimento de extensões específicas do cliente em software livre por pedido é particularmente comum em projetos menores. Geralmente, cabe ao cliente decidir como proceder com essas extensões, ou seja, se ele também as libera ou as mantém trancadas e fechadas como parte de seu próprio modelo de negócios.



■ Modelos de Negócios em Código Aberto

Uma coisa deveria ter ficado clara: embora o software livre esteja geralmente disponível gratuitamente, vários modelos de negócios foram criados em seu ambiente, que são constantemente modificados e ampliados por inúmeros freelancers e empresas em todo o mundo de uma forma muito criativa, o que, em última análise, também garante a existência continuada de todo o movimento FLOSS.

■ Exercícios

01 Quais são- em poucas palavras- as “quatro liberdades”, conforme definidas por Richard Stallman e pela Free-Software Foundation?

Liberdade 0

Liberdade 1

Liberdade 2

Liberdade 3

02 O que significa a abreviação FLOSS?

03 Você desenvolveu software livre e deseja garantir que o próprio software, mas também todos os trabalhos futuros baseados nele, também permaneçam livres. Qual licença você escolhe?

a) CC BY =

b) GPL Versão 3

c) Licença BSD de 2 Cláusulas

d) LGPL =

04 Qual das licenças abaixo você chamaria de permissiva, e qual você chamaria de CopyLeft?

BSD Simplificada

GPL Versão 3

CC BY

CC BY-SA

05 05- Você escreveu um aplicativo da Web e o publicou sob uma licença gratuita. Como você pode ganhar dinheiro com seu produto? Cite três possibilidades.