

Tales Araujo Mendonça  
Bruno Gonçalves Araujo

# **Linux**

## **Simplicidade ao Seu Alcance**



**editora**  
**VIENA**

1ª Edição  
Bauru/S.P.  
Editora Viena  
2012



# Sumário

<b>Lista de Siglas e Abreviaturas.....</b>	<b>17</b>
<b>1. Introdução.....</b>	<b>19</b>
1.1. Onde o Linux está? .....	21
1.2. A História.....	22
1.2.1. Como Tudo Começou.....	22
<b>2. Entenda os Conceitos .....</b>	<b>25</b>
2.1. Software e Hardware .....	27
2.2. Sistema Operacional, um Software Especial .....	27
2.2.1. Características de um Sistema Operacional Linux .....	27
2.3. Licença GPL .....	28
2.4. Projeto GNU .....	28
2.5. Software Livre.....	29
2.6. Interfaces Gráficas.....	30
2.6.1. KDE/Qt .....	30
2.6.2. Gnome/GTK.....	31
2.6.3. Outros.....	31
2.7. Padrões Abertos e Interoperabilidade .....	32
2.7.1. Software Legado .....	32
<b>3. Distribuição ou Distro Linux.....</b>	<b>35</b>
3.1. Escolha a Sua .....	37
3.1.1. Slackware.....	38
3.1.1.1. OpenSUSE.....	38
3.1.2. Gentoo Linux .....	39
3.1.2.1. Sabayon.....	39
3.1.3. Debian/GNU.....	40
3.1.3.1. Ubuntu .....	40
3.1.3.2. BigLinux.....	41
3.1.3.3. LinuxMint.....	41
3.1.4. Red Hat Linux .....	42
3.1.4.1. CentOS .....	42
3.1.4.2. Fedora .....	43
3.1.4.3. Mandriva .....	43
3.2. Dica .....	44
<b>4. Instalação.....</b>	<b>45</b>
4.1. Onde Encontrar a sua Distribuição .....	47
4.2. O que é Particionar .....	47
4.2.1. Organizar o Particionamento .....	48
4.2.2. Partição Swap .....	48
4.3. Instalação do Ubuntu, Xubuntu, Lubuntu e BigLinux .....	49
4.3.1. Preparando para Instalar o Ubuntu, Xubuntu, Lubuntu e BigLinux .....	50

4.3.1.1.	Instalação do Ubuntu, Xubuntu, Lubuntu e BigLinux – Caso 1, Instalação Padrão.....	51
4.3.1.2.	Instalação do Ubuntu, Xubuntu, Lubuntu e BigLinux – Caso 2, Modo Simples.....	52
4.3.1.3.	Instalação do Ubuntu, Xubuntu, Lubuntu e BigLinux – Caso 2, Modo Avançado..	53
4.3.2.	Onde Você Está? .....	58
4.3.3.	Layout do Teclado.....	59
4.3.4.	Quem é Você?.....	60
4.4.	Instalação do Fedora.....	61
4.4.1.	Escolha da Linguagem.....	61
4.4.2.	Definição do Teclado .....	62
4.4.3.	Dispositivo de Armazenamento .....	63
4.4.4.	Nome da Máquina .....	63
4.4.5.	Hora e Fuso Horário.....	64
4.4.6.	Senha do Superusuário.....	64
4.4.7.	Particionamento do Disco .....	65
4.4.8.	Perfil e Repositórios.....	66
4.4.9.	Configuração do Usuário .....	67
4.5.	Instalação do OpenSuse.....	68
4.5.1.	Linguagem e Teclado.....	70
4.5.2.	Hora e Fuso Horário.....	71
4.5.3.	Opções de Particionamento .....	72
4.5.4.	Criar Novo Usuário.....	73
4.5.5.	Configuração da Instalação.....	74
<b>5.</b>	<b>Interface Gráfica .....</b>	<b>75</b>
5.1.	Um Pouco Mais Sobre as Interfaces Gráficas .....	77
5.1.1.	Ambiente Unity .....	78
5.1.1.1.	Primeiras Impressões.....	78
5.1.1.1.1.	Área de Notificação .....	79
5.1.1.1.2.	Lançador.....	79
5.1.1.2.	Familiarize com o Ambiente .....	80
5.1.1.2.1.	Trabalhando com o Dash .....	80
5.1.1.2.2.	Incluir/Remover Aplicativos no Lançador.....	81
5.1.2.	Ambiente KDE.....	82
5.1.2.1.	Primeiras Impressões.....	83
5.1.2.2.	Familiarize com o Ambiente .....	83
5.1.3.	Ambiente Gnome-shell.....	85
5.1.3.1.	Primeiras Impressões.....	86
5.1.3.2.	Familiarize com o Ambiente .....	86
5.2.	Trabalhando com Aplicativos.....	87
5.2.1.	Gerenciando Janelas.....	87
5.2.1.1.	Personalizando o Tamanho de uma Janela Arrastando o Cursor.....	88
5.2.1.2.	Os Botões Minimizar, Maximizar/Restaurar e Fechar .....	89
5.2.1.3.	Mover a Janela .....	89
5.2.1.4.	Maximizando/Minimizando Janelas pela Barra de Tarefas de Títulos.....	90
5.3.	Aprendendo a Trabalhar com a sua Pasta Pessoal.....	90
5.3.1.	Criando uma Subpasta na Pasta Pessoal .....	91

5.3.2.	Abrindo um Arquivo ou Pasta .....	92
5.3.3.	Arrastando um Arquivo ou Pasta .....	92
5.3.4.	Copiando/Colando um Arquivo ou Pasta .....	93
5.3.5.	Recortando/Colando um Arquivo ou Pasta.....	96
5.3.6.	Excluindo um Arquivo ou Pasta.....	96
5.3.7.	Renomeando um Arquivo ou Pasta .....	97
5.4.	Teclas de Atalho.....	98
5.4.1.	<Alt+Tab> e <Alt+Shift+Tab>.....	98
5.4.2.	<Super> + Complementos.....	99
5.4.2.1.	Executar Aplicativos.....	99
5.4.2.2.	Exibir Janelas .....	100
5.4.2.3.	Exibir Áreas de Trabalho .....	100
5.4.3.	Alternância entre Áreas de Trabalho.....	101
5.4.4.	Outros Atalhos .....	102
5.5.	Instalando/Removendo Programas no Ubuntu.....	102
5.6.	Mantendo o Sistema Atualizado no Ubuntu .....	103
5.7.	Desligando o Computador.....	104
5.7.1.	Unity .....	105
5.7.2.	KDE.....	105
5.7.3.	Gnome-shell .....	106
5.8.	Administrando o Sistema.....	106
5.8.1.	Usuários.....	108
5.8.2.	Aparência.....	109
5.8.3.	Monitores.....	110
5.8.4.	Bluetooth .....	111
5.8.5.	Energia .....	111
5.8.6.	Mouse.....	111
5.8.7.	Teclado.....	112
5.8.8.	Som .....	114
5.8.9.	Data e Hora.....	115
5.8.10.	Impressora .....	116
5.8.10.1.	Impressora Local .....	116
5.8.10.2.	Impressora em Rede .....	119
5.8.11.	Rede.....	120
5.8.11.1.	Com Fio.....	121
5.8.11.2.	Sem Fio (Wireless).....	122
5.8.11.3.	3G.....	122
5.8.11.4.	DSL .....	123
<b>6.</b>	<b>Conhecendo os Programas Linux.....</b>	<b>125</b>
6.1.	Ferramentas de Internet .....	127
6.1.1.	Navegadores de Internet .....	127
6.1.1.1.	Mozilla Firefox .....	127
6.1.1.2.	Google Chrome/Chromium .....	128
6.1.2.	Comunicadores.....	129
6.1.2.1.	aMSN.....	129
6.1.2.2.	Empathy .....	129

6.1.2.3.	Pidgin .....	130
6.1.2.4.	Ekiga .....	130
6.1.3.	E-mail: Thunderbird.....	131
6.1.4.	Ferramentas para Downloads .....	131
6.1.4.1.	Jdownloader .....	131
6.1.4.2.	Kget.....	132
6.1.4.3.	qBittorrent .....	133
6.1.4.4.	aMule.....	134
6.1.4.5.	Frostwire.....	135
6.1.5.	Acesso Remoto ao Desktop.....	135
6.1.5.1.	TeamViewer.....	135
6.1.5.2.	Vino/Vinagre.....	136
6.1.6.	Outros Programas Relacionados .....	138
6.2.	Ferramentas de Escritório .....	138
6.2.1.	Editor de Texto: LibreOffice Writer .....	138
6.2.2.	Planilha Eletrônica: LibreOffice Calc.....	139
6.2.3.	Apresentações Gráficas: LibreOffice Impress.....	140
6.2.4.	Diagramação de Páginas: Scribus .....	141
6.2.5.	Leitor de PDF .....	142
6.2.5.1.	Okular .....	142
6.2.5.2.	Evince .....	143
6.2.6.	Não Perca seus Compromissos: Kalarm .....	144
6.3.	Gráficos .....	144
6.3.1.	Edição de Imagens Profissionais: Gimp.....	144
6.3.2.	Criação de Objetos Vetoriais: Inkscape .....	145
6.3.3.	Modelagem 3D: Blender.....	146
6.3.4.	Edição de Imagens: Showfoto .....	147
6.4.	Multimídia.....	148
6.4.1.	Reprodutor de Mídia.....	148
6.4.1.1.	SMPlayer.....	148
6.4.1.2.	VLC.....	149
6.4.1.3.	Totem.....	150
6.4.2.	Gravação e Cópia de CD e DVD.....	150
6.4.2.1.	K3b .....	150
6.4.2.2.	Brasero .....	151
6.4.3.	Conversor Simples de Áudio .....	151
6.4.3.1.	K3B.....	152
6.4.3.2.	SoundKonverter .....	152
6.4.4.	Conversor de Vídeo .....	153
6.4.4.1.	Big Converter Vídeo .....	153
6.4.4.2.	VLC.....	154
6.4.5.	Player de Música .....	155
6.4.5.1.	Amarok.....	155
6.4.5.2.	Tomahawk .....	156
6.4.5.3.	Audacious .....	157
6.4.5.4.	Rhythmbox.....	157

6.4.5.5.	Nightingale.....	158
6.4.6.	K9copy: Ripador, Conversor de DVD.....	158
6.4.7.	Edição de Vídeo Simples.....	159
6.4.7.1.	Avidemux.....	159
6.4.7.2.	OpenShot Video Editor.....	159
6.4.7.3.	KdenLive.....	160
6.4.8.	Criando Vídeos e Tirando Fotos com a Webcam: Cheese.....	161
6.4.9.	Edição de Áudio com Audacity.....	161
6.5.	CAD.....	162
<b>7.</b>	<b>Introdução à Árvore de Diretórios.....</b>	<b>165</b>
7.1.	O que são Diretórios?.....	167
7.2.	Entendendo a Estrutura de Diretórios.....	168
7.2.1.	Diretório Raiz (/).....	168
7.2.2.	Diretório dos Usuários (/home).....	168
7.2.3.	Acesso a Outras Partições e Dispositivos (/media, /run/media e /mnt).....	169
7.2.4.	Organização de Arquivos dos Programas.....	169
7.2.4.1.	Executáveis do Sistema (/bin, /usr/bin, /usr/sbin/, /usr/local/bin, /sbin /usr/games).....	169
7.2.4.2.	Entenda a Separação dos Diretórios Executáveis.....	169
7.2.5.	Configurações do Sistema (/etc).....	170
7.2.6.	Recursos do Sistema Unix (/usr).....	171
7.2.7.	Biblioteca dos Aplicativos (/lib, /usr/lib, /usr/local/lib).....	171
7.2.8.	Diretório Temporário (/tmp).....	171
7.2.9.	Variáveis do Sistema e Arquivos de Log (/var).....	172
7.2.10.	Diretórios especiais (/dev, /proc, /sys).....	173
<b>8.</b>	<b>Introdução aos Sistemas de Arquivos.....</b>	<b>175</b>
<b>9.</b>	<b>Introdução ao Terminal de Comandos.....</b>	<b>179</b>
9.1.	Entendendo o Shell.....	182
9.2.	Case Sensitive.....	182
9.3.	Interpretando os Comandos.....	183
9.4.	Preenchimento Automático de Comandos com a Tecla <Tab>.....	184
9.5.	Comandos do Terminal.....	184
9.5.1.	Listar Arquivos e Pastas (ls).....	185
9.5.2.	Acessar Pastas ou Diretórios (cd).....	186
9.5.3.	Descobrir a Pasta Corrente (pwd).....	186
9.5.4.	Cópia de Arquivos e Diretórios (cp).....	187
9.5.5.	Criar Pastas ou Diretórios (mkdir).....	188
9.5.6.	Criar Links ou Ponteiros (ln).....	189
9.5.7.	Criar e Alterar Senha do Usuário (passwd).....	189
9.5.8.	Exibir e Alterar a Data/Hora do Sistema (date).....	190
9.5.9.	Finalizar Sessão do Usuário (exit).....	190
9.5.10.	Limpar a Tela do Terminal de Comandos (clear).....	191
9.5.11.	Mover e Renomear, Arquivos e Pastas (mv).....	191
9.5.12.	Identificar Usuários Logados no Sistema (who).....	192
9.5.13.	Informações e Configurações da Rede (ifconfig).....	193
9.5.14.	Teste de Conectividade na Rede (ping).....	194

9.5.15.	Remover Arquivos e Diretórios (rm).....	195
9.5.16.	Obter Informações de Espaço dos Discos (df).....	196
9.5.17.	Obter o Espaço Ocupado por Arquivos e Pastas (du).....	196
9.5.18.	Trocar e Executar Aplicativos com Outro Usuário (su).....	197
9.5.19.	Informações sobre Memória, Buffer e Cache (free).....	198
9.5.20.	Obter Informações do Sistema com uname.....	198
9.5.21.	Compactação e Descompactação de Arquivos.....	199
9.5.21.1.	Compactação/Descompactação com tar.gz.....	199
9.5.21.2.	Compactação/Descompactação com zip/unzip.....	200
9.5.22.	Permissões de Arquivos e Pastas/Diretórios (chmod).....	201
9.5.23.	Permissões de Usuários e Grupos (chown).....	203
9.5.24.	Histórico de Comandos Digitados (history).....	203
9.6.	Gerenciar Arquivos do Sistema com Pacotes .deb.....	204
9.6.1.	Gerenciar Pacotes .deb com o dpkg.....	204
9.6.2.	Gerenciar Pacotes .deb com o APT.....	205
9.6.2.1.	Procurar Pacotes e Obter Informações de Pacotes com o APT.....	206
9.6.2.2.	Instalar e Remover Programas com o APT.....	206
9.6.2.3.	Atualizar com o APT.....	207
9.6.2.4.	Dicas para o APT.....	208
<b>10.</b>	<b>Aplicativos para Windows no Linux.....</b>	<b>211</b>
10.1.	Wine.....	213
10.1.1.	Play On Linux.....	214
10.1.2.	GameTree (Cedega).....	215
<b>11.</b>	<b>Exercícios.....</b>	<b>217</b>
11.1.	Questões do Capítulo 1 - Introdução.....	219
11.2.	Questões do Capítulo 2 - Entenda os Conceitos.....	220
11.3.	Questões do Capítulo 3 - Distribuição ou Distro Linux.....	222
11.4.	Questão do Capítulo 4 - Instalação.....	224
11.5.	Questões do Capítulo 7 - Introdução à Árvore de Diretórios.....	224
11.6.	Questões do Capítulo 8 - Introdução aos Sistemas de Arquivos.....	226
11.7.	Questões do Capítulo 9 - Introdução ao Terminal de Comandos.....	226
11.8.	Questões do Capítulo 10 - Aplicativos para Windows no Linux.....	234
	<b>Referências.....</b>	<b>235</b>
	<b>Glossário.....</b>	<b>237</b>

# Lista de Siglas e Abreviaturas

<i>↷</i>	<i>Indica que a linha de comando continua na linha abaixo, ou seja, as duas linhas são uma única linha de comando.</i>
<i>/usr</i>	<i>Unix System Resources.</i>
<i>APT</i>	<i>Advanced Packaging Tool.</i>
<i>ENIAC</i>	<i>Electronic Numerical Integrator and Computer.</i>
<i>EUA</i>	<i>Estados Unidos da América.</i>
<i>FSF</i>	<i>Fundação do Software livre ou Free Software Foundation.</i>
<i>FTP</i>	<i>File Transfer Protocol.</i>
<i>GPL</i>	<i>Licença Pública Geral.</i>
<i>ISO</i>	<i>International Organization for Standardization.</i>
<i>LVM</i>	<i>Logic Volume Manager.</i>
<i>LTS</i>	<i>Long Term Support ou Longo Tempo de Suporte.</i>
<i>MTU</i>	<i>Maximum Transmission Units.</i>
<i>ODF</i>	<i>Open Document Format for Office Applications.</i>
<i>PPD</i>	<i>PostScript Printer Description.</i>
<i>RPM</i>	<i>Red Hat Package Manager.</i>
<i>YUM</i>	<i>Yellow Updater Modified.</i>



# 1

## Introdução

Linux

- 1.1. **Onde o Linux Está?**
- 1.2. **A História**
  - 1.2.1. Como Tudo Começou



# 1. Introdução

É comum ouvirmos: “**Por que usar Linux?**” A indagação mais adequada, no entanto, deveria ser: “**Por que NÃO usar Linux?**”

Se existe uma opção gratuita, considerada mais segura e estável que as opções pagas, e, mais ainda, que você pode gerar cópias e dar aos seus amigos, que não está limitada aos conhecimentos de uma empresa (onde qualquer pessoa no mundo com conhecimento técnico pode alterar da forma que quiser e enviar para os responsáveis pelo projeto avaliarem), se é algo bom a ser incluído por padrão e disponível para todos. São razões para se usar o **Linux**.

O **Linux** necessita de menos capacidade de hardware para ser utilizado; por isso o tempo de utilização do equipamento, sem a necessidade de aquisição de um outro, será maior. Cabe acrescentar que essa durabilidade proporciona economia e protege o ambiente (gerando menos lixo).

O código fonte do **Linux** é disponível, daí a relação que muitas pessoas, empresas e governos têm de analisarem a fim de saber se não existem “armadilhas”. Isso ocorre, principalmente em áreas estratégicas como a de defesa nacional, pois é muito importante ter a certeza de que a empresa responsável pelo sistema não ocultou formas de ter acesso aos seus arquivos. Essas características não são exclusividade do **Linux**, elas são compartilhadas por grande parte dos **Softwares Livres**. Dessa forma, podemos concluir que a escolha por **Linux** e **Softwares Livres** deve ser a primeira opção. Se por algum motivo esses não possuírem os recursos necessários, então, aí sim, poderá ser vantajoso o uso de um **Software Proprietário**. O custo para criar esse recurso como um **Software Livre** pode não compensar. Dessa forma o uso de **Software Livre** para grandes empresas e governos se torna ainda mais importante.

Apesar de todas as evidências relatadas do uso do **Software Livre**, algumas pessoas alegam que o computador que compraram veio com um sistema operacional proprietário já incluso, mas não precisaram pagar por ele. O que ocorre na verdade é que o preço do sistema operacional já está incluso no preço do computador: o comprador pagou pelo computador, mas ao ligá-lo pela primeira vez será feita uma pergunta se concorda com a licença de uso; basta selecionar não e entrar em contato com a empresa responsável pelo sistema operacional para pedir a restituição do valor do software. Com isso, você poderá instalar outro sistema operacional e receber o valor que já estava incluso no preço do computador adquirido. Dessa forma, deixo mais uma vez a pergunta, por que não usar **Linux**?

## 1.1. Onde o Linux está?

Uma pesquisa realizada em 2010 apontou que dos 500 mais velozes supercomputadores do mundo, 470 estavam rodando **Linux**. Grande parte dos sites da internet estão hospedados em servidores **Linux**, inclusive sites gigantes como o Google e o Facebook.

Todos os celulares e tablets Android, que hoje são os mais vendidos no mundo, são **Linux**. Boa parte das televisões modernas que dispõem de recursos como acesso à internet e jogos, utilizam **Linux**. Diversos caixas eletrônicos, esses que utilizamos para sacar dinheiro e fazer outras transferências bancárias, são **Linux**.

Roteadores, filmadoras, aparelhos de DVD... O Linux está espalhado realmente por todos os lados, o uso no Desktop, que é o foco deste livro é apenas um dos inúmeros usos possíveis. Tudo isso ocorre somente por se tratar de um **Software Livre**, onde diversas pessoas e empresas criam novos módulos e outras melhorias para adaptá-lo às suas necessidades. Dessa forma, temos uma espécie de “canivete suíço” dos sistemas operacionais, pois tem a capacidade de funcionar, em geral, nas mais diversas necessidades, e ainda que não funcione é livre para ser adaptado e adquirir mais uma forma de uso.

## 1.2. A História

### 1.2.1. Como Tudo Começou

Antes de propriamente iniciar a história do Linux, faz-se necessário contar uma outra história. Nesse sentido, fatos ocorridos antes do surgimento do Linux serão abordados, pois é necessário considerar o passado: nele estão muitas respostas para os acontecimentos de hoje.

No ano de 1500, último do século XV, foi inventada a primeira máquina desenvolvida para efetuar operações matemáticas simples. Uma calculadora mecânica criada por Leonardo da Vinci.

Três séculos mais tarde, no ano de 1801, era criada a primeira máquina mecânica programável da história: um tear mecânico controlado por grandes cartões perfurados. A grandiosa e inovadora engenhoca foi criada pelo mecânico francês Joseph Marie Jacquard. Os mesmos padrões utilizados na máquina de tear foram usados pelos primeiros computadores eletrônicos desde a década de 40 até o surgimento de métodos de armazenamento mais confiáveis.

Alguns anos mais tarde, no ano de 1833, Charles Babbage, com o auxílio de sua esposa, Ada August Byron King, projetou a máquina analítica. Ada August foi a primeira programadora dessa máquina. Essa criação era constituída de unidade de controle de memória, aritmética, de entrada e de saída. Suas operações eram comandadas por cartões perfurados. Babbage investiu toda sua fortuna pessoal e a de seu filho na construção dessa máquina inovadora. Infelizmente, faleceu, em 1871, antes de ver o seu sonho tornar-se realidade; haviam muitas limitações tecnológicas naquela época. No entanto, chegou a esclarecer que uma máquina computacional deveria ser composta de um dispositivo de entrada, uma memória e um dispositivo de saída. Sua esposa, Ada Lovelace, tornou-se a pioneira da lógica de programação. Ela escreveu séries de instruções para a calculadora analítica; inventou o conceito de sub-rotinas; descobriu o valor das repetições, loops; e iniciou o desenvolvimento do desvio condicional.

Charles Babbage pode ser considerado o avô do computador digital. Apesar de não ter terminado sua máquina, criou a base para fabricação de equipamentos computacionais.

No ano de 1854, o matemático George Boole fez uma publicação propondo a álgebra binária booleana para realizar operações com números binários (0 ou 1), assumindo apenas dois valores, verdadeiro ou falso. Um ano mais tarde, em 1855, foi construído o primeiro computador mecânico, por George e Edvard Scheutz, baseado no trabalho de Babbage.

O primeiro “bug” de computador foi registrado por Grace Murray Hopper, oficial naval e matemático, em 9 de setembro de 1945. O “bug” na verdade era um inseto preso entre os relés do computador.

Um ano após a descoberta do primeiro “bug” de computador, em 14 de fevereiro de 1946, foi anunciado o primeiro computador elétrico digital de grande escala, de nome ENIAC (Electronic Numerical Integrator and Computer). Era um projeto dos EUA, utilizando apenas válvulas, tinha o objetivo de calcular trajetórias de mísseis com maior precisão. Sendo desativado após o fim da guerra.

Próximo da chegada da década de 70, em 1969, dois programadores dos laboratórios AT&T Bell, Ken Thompson e Denis Richie, desenvolveram o primeiro sistema operacional que poderia ser instalado em qualquer máquina, o sistema operacional UNIX.

Em janeiro de 1984, Richard Stallman, fundador da FSF (Fundação do Software livre), criou um projeto com o codinome GNU. Esse projeto propunha a criação de um sistema operacional totalmente livre e compatível com o sistema operacional UNIX, porém não deveria utilizar o código fonte do UNIX.

O nome GNU, escolhido por Stallman, foi referenciado a um manifesto, esse por sua vez, é um acrônimo recursivo de: GNU is Not Unix (em português: GNU não é Unix).

Stallman, em primeiro momento, não conseguiu sucesso com o que havia planejado, pois possuía alguns aplicativos livres, mas não um kernel (núcleo) do sistema operacional, responsável por fazer uma ponte de comunicação entre os seus aplicativos criados e o hardware (parte física, o computador em si) da máquina.

Houve uma época em que o UNIX era o sistema operacional predominante em faculdades e empresas. Na Universidade de Helsinki, na Finlândia, um universitário chamado Linus Torvalds, estudante do curso de Ciência da Computação, havia iniciado um projeto particular de um kernel, onde ele pudesse criar seus próprios drivers para os dispositivos e ter a liberdade de uso não conseguida por ele até então. Esse projeto foi iniciado com base no sistema Minix, um pequeno sistema UNIX, desenvolvido por Andrew S. Tanenbaum, que Linus simpatizava.

Iniciado o projeto, Linus decidiu distribuir o seu kernel (os códigos fontes) na faculdade, a fim de juntar colaboradores para ajudar no desenvolvimento do sistema. Nascia então o núcleo do sistema operacional.

Segue um trecho da mensagem que Linus Torvalds enviou para `comp.os.minix` com o objetivo de apresentar o seu projeto e encontrar pessoas dispostas a colaborar:

Você suspira pelos bons tempos do Minix-1.1, quando os homens eram homens e escreviam seus próprios “device drivers”? Você está sem um bom projeto em mãos e está desejando trabalhar num S.O. que você possa modificar de acordo com as suas necessidades? Está achando frustrante quando tudo funciona no Minix? Chega de noite ao computador para conseguir que os programas funcionem? Então, esta mensagem pode ser exatamente para você.

Como eu mencionei há um mês atrás, estou trabalhando numa versão independente de um S.O. similar ao Minix para



# 2

## Entenda os Conceitos

- 2.1. Software e Hardware**
- 2.2. Sistema Operacional, um Software Especial**
  - 2.2.1. Características de um Sistema Operacional Linux
- 2.3. Licença GPL**
- 2.4. Projeto GNU**
- 2.5. Software Livre**
- 2.6. Interfaces Gráficas**
  - 2.6.1. KDE/Qt
  - 2.6.2. Gnome/GTK
  - 2.6.3. Outros
- 2.7. Padrões Abertos e Interoperabilidade**
  - 2.7.1. Software Legado



## 2. Entenda os Conceitos

### 2.1. Software e Hardware

Explicando de forma simplificada o que é software e hardware, e com intuito de facilitar o entendimento do leitor, pode-se dizer que o software é a parte lógica, virtualizada, que será processada pelo hardware. Detalhando, temos que o hardware, em computação, é a parte física do computador, ou seja, são as placas, memória, processador, circuitos em geral. Esses por sua vez irão processar as instruções que foram escritas por alguma pessoa. O hardware pega as instruções que o programador criou e as interpreta, criando o que é conhecido como software, que nada mais é do que um programa computacional.

### 2.2. Sistema Operacional, um Software Especial

Como descrito no item anterior, software é um programa computacional, por exemplo: editor de textos, editor de imagens, tocador de músicas, compactador de dados, navegador de Internet, etc.

Um software geralmente está relacionado com tarefas específicas como citado, já o sistema operacional é também um software, no entanto ele possui uma característica especial, é ele que se comunica com o hardware (a parte física do computador), enquanto os outros programas se comunicam com o sistema operacional. É como se o sistema operacional fosse uma ponte que liga os outros softwares ao hardware.

Existem diversos sistemas operacionais, alguns mais populares são o Windows da Microsoft, MacOS da Apple e o Linux que é gerenciado por uma organização chamada The Linux Foundation que é um consórcio sem fins lucrativos dedicado a promover o crescimento do Linux, o sistema que explicaremos neste livro.

#### 2.2.1. Características de um Sistema Operacional Linux

O Linux é sistema operacional que apresenta algumas particularidades, como ser um sistema multiusuário e multitarefas.

Explicando primeiramente o que vem a ser um sistema multitarefas e afirmando que a maioria dos sistemas operacionais existentes atualmente são multitarefas, como: o Linux, o Unix, o FreeBSD, o MacOS e até mesmo o Windows da Microsoft. Um sistema multitarefas é um sistema que consegue executar várias tarefas de forma simultânea, como, por exemplo: editar um texto, navegar na internet e ouvir a uma música, ou seja, o sistema está executando várias tarefas ao mesmo tempo. Já num sistema monotarefas (mais encontrados em sistemas antigos e legados) você só conseguiria editar um texto, navegar na internet ou escutar uma música, realizando apenas uma tarefa por vez. Nesse caso, se você estivesse editando um texto e quisesse ouvir uma música, teria que fechar o seu editor para abrir o seu programa de escutar músicas.

Além do Linux ser um sistema multitarefas ele também é um sistema multiusuário, ao contrário do Microsoft Windows que é monousuário. Em um sistema multiusuário é permitido a utilização de vários usuários de forma simultânea, ou seja, vários usuários utilizando o mesmo sistema operacional ao mesmo tempo.

Para que um sistema operacional seja considerado multiusuário, é preciso que possua algumas características. O sistema deve controlar o que cada usuário está executando no sistema de forma individual, por exemplo: vários usuários acessando suas contas de e-mails ao mesmo tempo em um computador que está rodando o sistema operacional Linux e possui um servidor de e-mails configurado. O sistema operacional multiusuário deve entender, de forma individual, o que cada usuário está fazendo no sistema, de modo que um não pode interferir na utilização do outro, mesmo quando o uso é simultâneo.

Outro detalhe importante é que um sistema multiusuário deve entender o que cada usuário faz no sistema e deve associar essas funções a cada usuário de forma separada, por exemplo: o usuário que cria um documento de texto, uma planilha eletrônica, cria pastas, copia pastas de um lugar para o outro. Tudo isso é visto e entendido pelo sistema operacional de forma separada de modo que um usuário não interfira nas tarefas, permissões e processos do outro.

## 2.3. Licença GPL

A Licença Pública Geral GNU ou apenas GPL é a denominação de uma licença para o software livre, criada por Richard Stallman, seu idealizador, no final da década de 1980 na Free Software Foundation (FSF). A GPL é a licença que possui maior participação nos projetos de software livre, sendo boa parte ligada a projetos relacionados ao Linux.

Todo programa escrito sob a licença GPL pode ser alterado e comercializado, desde que siga os quatro princípios básicos ou as quatro liberdades:

- **Liberdade n° 0:** A liberdade de executar o programa, para qualquer propósito.
- **Liberdade n° 1:** A liberdade de estudar como o programa funciona e adaptá-lo para as suas necessidades. Acesso ao código fonte é um pré-requisito para esta liberdade.
- **Liberdade n° 2:** A liberdade de redistribuir cópias, de modo que você possa ajudar o próximo.
- **Liberdade n° 3:** A liberdade de aperfeiçoar o programa e liberar os seus aperfeiçoamentos, de modo que toda a comunidade se beneficie. Acesso ao código fonte é um pré-requisito para esta liberdade.

Um programa é tido como software livre se possuir todas essas liberdades. Portanto, você é livre para redistribuir cópias de softwares livres, com ou sem modificações, gratuitamente ou cobrando uma taxa pela distribuição e empacotamento do software, para qualquer um em qualquer lugar.

## 2.4. Projeto GNU

Richard Stallman, o fundador da FSF (Free Software Foundation), em janeiro de 1984 criou um projeto chamado **GNU**. Stallman visava criar um sistema operacional totalmente livre e que fosse compatível com o sistema Unix, mas não deveria conter ou utilizar o código fonte do Unix.

O nome **GNU**, escolhido por Stallman, foi referenciado ao manifesto Gnu e é um acrônimo recursivo de: GNU is Not Unix (em português: GNU não é Unix). O projeto de Stallman não foi bem sucedido até um certo ponto, pois ele tinha os aplicativos: compilador (gcc), editor de texto (emacs), bash (terminal de comandos), etc. Mas não tinha o principal, o kernel (núcleo) do sistema operacional. Foi em 1991 que o útil uniu-se ao agradável. Um jovem finlandês, chamado Linus Torvalds, tinha desenvolvido um kernel que poderia usar todas as ferramentas **GNU**, dando início ao projeto **GNU/Linux**.

## 2.5. Software Livre

O termo software livre veio do inglês Free Software. É exatamente o termo “free” que tem trazido grandes repercussões aos defensores dessa ideologia. “Free” pode ser interpretado ou traduzido tanto como “livre” no sentido de liberdade, como “livre” no sentido de ser gratuito.

Software livre, segundo a definição criada pela Fundação do Software Livre – FSF (Free Software Foundation), é qualquer programa de computador que pode ser usado, copiado, estudado, modificado e redistribuído sem nenhuma restrição. A liberdade de tais diretrizes é central ao conceito, o qual se opõe ao conceito de software proprietário, mas não ao software que é vendido almejando lucro (software comercial). A maneira usual de distribuição de software livre é anexar a ele uma licença de software livre e tornar o código fonte do programa disponível.

Sendo assim, nada impediria o usuário de Linux de copiar um CD da sua distribuição Linux favorita e disponibilizá-lo para um amigo, por exemplo. Do mesmo modo, encontramos distribuições vendidas em embalagens bem apresentadas, com manuais de instalação e documentação completa, vendidas em supermercados, revistarias e lojas especializadas.

O termo software livre, bem como as suas licenças, garantem apenas que o usuário tenha plenos acessos aos seus códigos fontes, tendo liberdade para estudá-lo, modificá-lo e copiá-lo sem pagar por isso.

Dentre as muitas licenças livres que encontramos no “mundo livre”, a que se destaca é a GPL (Licença Pública Geral). Essa licença, também escrita pela Free Software Foundation, visa garantir exatamente as quatro liberdades que o software livre defende. Uma das peculiaridades desta licença é que além das quatro liberdades que se tem sobre o software, ainda garante-se o direito autoral do(s) desenvolvedor(es). Mesmo que você tenha apenas iniciado o software com um pequeno pedaço de código ou se você colaborou com uma grande rotina no decorrer do desenvolvimento, o seu nome estará fixado nos direitos autorais do software.

Richard Stallman, o criador do projeto **GNU**, faz uma analogia entre os programas de computador e as receitas de comida, fazendo uma comparação entre o que é ser livre e o que é ser proprietário. Ele diz que um programa se parece muito com uma receita, ambos são uma lista de passos que devem ser seguidos com cuidado, com regras que determinam quando terminar ou quando retornar. No final, obtém-se um certo resultado.

Stallman, ainda em sua analogia diz que se você gosta de cozinhar, provavelmente troca as receitas com os amigos e as altera também! Se você mudou a receita e o resultado te agrada e seus amigos gostam de comer isso, você talvez dará a versão alterada da

receita a eles. Essa liberdade é equivalente ao que temos com o software livre, podemos compartilhar o conhecimento e aprimorá-lo, já com o software proprietário apenas existem fórmulas secretas que não podem ser reproduzidas ou aprimoradas.

## 2.6. Interfaces Gráficas

Neste livro mostramos programas feitos em diferentes bibliotecas gráficas, em geral aplicativos Qt são associados ao ambiente KDE e aplicativos GTK ao ambiente Gnome. Esses ambientes utilizam exclusivamente essas bibliotecas gráficas em seus aplicativos, porém isso não significa que um aplicativo feito para um ambiente não irá funcionar corretamente em outro. Com raras exceções todos os aplicativos funcionam corretamente em qualquer ambiente.

As diferenças mais notáveis para o usuário ao utilizar um programa feito em uma biblioteca gráfica diferente, são ligadas ao tema do programa que tende a ser diferente, bem como as janelas de abrir, salvar e imprimir, mas raramente são diferenças que comprometam de alguma forma o uso do programa.

O uso simultâneo de bibliotecas gráficas diferentes tende a aumentar o consumo de memória do computador, isso já foi motivo de preocupação, porém atualmente essa diferença é perceptível apenas em computadores muito obsoletos. A capacidade disponível em computadores atuais torna essa pequena diferença no uso de memória algo irrelevante.

### 2.6.1. KDE/Qt

O **KDE** é um ambiente de trabalho feito com a biblioteca **Qt**, que visa disponibilizar praticamente todos os aplicativos necessários para o uso de um computador doméstico ou de escritório, sua primeira versão foi lançada em julho de 1998. Atualmente está na versão 4.8 e a cada 6 meses é lançada uma nova versão trazendo mais funcionalidades e correções de erros.

Um grande destaque para o ambiente **KDE** é sua possibilidade de personalização, dispõe de uma série de opções fáceis para configurar o ambiente, trocando o posicionamento e aparência das ferramentas.

**Qt** é originalmente uma biblioteca gráfica, ou seja, é utilizada para criar interfaces de fácil compreensão para o usuário, com botões, listas de opções e todos os recursos que estamos habituados a utilizar. Atualmente se tornou um grande “framework”, ou seja, disponibiliza uma série de ferramentas de desenvolvimento que simplificam a criação de inúmeros programas, indo de facilidades para criação da interface gráfica até mesmo à parte de áudio e integração com sistemas de bancos de dados.

A biblioteca **Qt** passou por algumas formas de licenciamentos diferentes, que variavam de acordo com o sistema operacional utilizado e também a licença do programa criado com a biblioteca. A partir de 2005 passou a disponibilizar uma versão GPL para todos os sistemas operacionais compatíveis, e a partir de 2009 passou a disponibilizar sob a licença LGPL. O licenciamento LGPL facilita a criação de aplicativos comerciais e até mesmo aplicativos proprietários utilizando a biblioteca **Qt**.